

UNIVERSITAT POLITÈCNICA DE VALÈNCIA  
ESCOLA POLITÈCNICA SUPERIOR DE GANDIA  
I.T. TELECOMUNICACIÓ (SIST. ELECTRÒNICS)

---



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



ESCUELA POLITÉCNICA  
SUPERIOR DE GANDIA

**“Disseny i Implementació d'un Sistema  
Electrònic de Control d'Estabilitat i  
Posicionament d'Objectes”**

*TREBALL FINAL DE CARRERA*

Autor:  
**Josep Gil Mora**

Director:  
**José Pelegrí Sebastià**

*GANDIA, 2012*



*Per què és tot tan complicat?  
Les coses són tan complicades com un vol que siguen*

Fins que Elizabeth Yáñez no se'n va eixir de la meua vida, mai se m'havia passat pel cap agrair o dedicar el treball de final de carrera a ningú. Sí, un detall egoista per la meua part. I sí, resulta curios veure com sempre ens donem compte de moltes coses quan algú decideix eixir-se'n de la nostra vida...

Pot ser continues pensant que un detall d'aquest estil per part d'una persona és excessiu... i pot ser tingues raó, però no importa, perquè de vegades les coses són el que un vol que siguin i les dedicatòries no són una excepció, al fi i al cap, una dedicatòria pot ser tant una successió de paraules escrites sobre un paper, com també pot ser una simple forma de recordar-li a les persones que, una determinada persona, pot influir en la vida d'una altra fins el punt de fer-la tremolar en tots els aspectes

I sí algú que entra i ix de la vida d'una persona es mereix una dedicatòria, els que sempre estan en ella, no es mereixen menys. Després de tot, vosaltres sou els que em recolzeu sempre, inclús quan no me'n adone, i els que, a pesar de tot, continueu aguantant-me a mi... i a les meves coses. Sí, em referisc a tots els que esteu al meu voltant, especialment als que no esteu sempre físicament al meu costat... Alex, Pep, Salva, sé que açò soles són paraules, però en tot aquest temps heu estat en el meu cap més del que pareix... encara que no tant com m'agradaria dir i com vosaltres vos mereixieu ;)

També voldria agrair a Toni Bou, el tècnic de laboratori de les vesprades, tot el que ha fet per mi, tant des de el punt de vista tècnic com el personal. Sense la seva ajuda aquest projecte no hauria sigut possible ja que no soles m'ha ajudat considerablement en la part del disseny amb l'*Orcad Layout*, sinó que a més, amb els seus comentaris i idees m'ha fet caure en detalls que d'una altra forma hauria tardat en descobrir o, directament, no hauria pogut veure. Toni, Linux sempre serà millor que Windows!!!

Per últim, agrair a tots els que em deixen pel camí la part que els pertoca. Pot ser deuria citar-vos de la mateixa manera que ho he fet amb els altres, però de vegades, no és bona idea reconèixer la importància d'una persona amb paraules, sinó amb fets. No creus?



# Índex de continguts

|  |    |
|--|----|
| 1- Introducció i objectiu del projecte.....  | 10 |
| 2- Localització d'un objecte mitjançant satèl·lits.....  | 11 |
| 2.1- Introducció: La localització per satèl·lit i els sistemes de coordenades.....   | 11 |
| 2.2- Els sistemes de navegació NAVSTAR-GPS, GLONASS i GALILEU.....   | 12 |
| 2.3- Conceptes bàsics.....   | 14 |
| 2.4- GPS Diferencial i Sistemes d'Augment Basat en Satèl·lits (SBAS).....  | 16 |
| 2.5- Missatges de navegació GPS.....   | 17 |
| 2.6 Un protocol per a la transmissió de missatges GPS: NMEA-0183.....  | 18 |
| 2.6.1- Frases específiques de NMEA-0183 per a receptors GPS.....   | 19 |
| 3- Sensors de moviment: Una altra forma de localitzar/posicionar objectes.....   | 24 |
| 3.1- Introducció: Per què utilitzar altres formes de localitzar objectes?.....   | 24 |
| 3.2- Acceleròmetres.....   | 24 |
| 3.2.1- Introducció: Què és un acceleròmetre?.....  | 24 |
| 3.2.2- Tipus d'acceleròmetres.....   | 24 |
| 3.2.3- Aplicacions dels acceleròmetres.....  | 28 |
| 3.3- Giroscopis.....   | 29 |
| 3.3.1- Introducció: Què és un Giroscopi?.....  | 29 |
| 3.3.2- Giroscopis no mecànics: Giroscopis d'estructura ressonant i Giroscopis de Fibra Òptica (FOG).....                             | 31 |
| 3.3.3- Aplicacions.....  | 32 |
| 3.4- Inclínometres: Tipus i aplicacions.....   | 33 |
| 3.5- Magnetòmetres.....  | 35 |
| 3.5.1- Introducció: Què és un Magnetòmetre?.....   | 35 |
| 3.5.2- Tipus de magnetòmetres.....   | 36 |
| 3.5.3- Aplicacions dels magnetòmetres.....   | 44 |
| 3.6- Sensors MEMS: els “tot en un”.....  | 45 |
| 3.6.1 Problemàtica dels dispositius MEMS: efectes de la temperatura, <i>Random Walk i Bias Stability</i> .....                       | 49 |
| 4- Sistemes de Navegació Inercial, una aplicació específica dels sensors de moviment i dels sistemes de localització d'objectes..... | 53 |
| 4.1- Introducció: Sistemes de Navegació, Navegació Inercial i Sistemes Guiats.....   | 53 |
| 4.2- Història dels Sistemes de Navegació Inercial.....   | 55 |
| 4.3- Sistemes de Navegació Inercial bàsics.....  | 56 |
| 5- Una possible aplicació d'un sistema de control d'estabilitat: Sensor d'ones.....  | 59 |
| 5.1- Introducció: Per què mesurar ones?.....   | 59 |
| 5.2- Ones: un dels elements inseparables de l'oceà.....  | 60 |
| 5.3- Tipus d'ones.....   | 61 |
| 5.4- Matemàtiques dels oceans: Equacions de moviment de les ones.....  | 63 |
| 5.5- Teoria lineal d'ones de superfície de l'oceà.....   | 70 |
| 5.5.1- Relació de dispersió.....   | 71 |
| 5.5.2- Velocitat de fase.....  | 71 |
| 5.5.3- Velocitat de grup.....  | 72 |
| 5.5.4- Energia i altura significant d'ona.....   | 72 |
| 5.5.5- Trencament de l'ona.....  | 73 |
| 5.6- Teoria no lineal d'ones de superfície de l'oceà.....  | 74 |
| 5.6.1- Teoria de l'ona solitària o Tsunami.....  | 75 |
| 5.7- Mesurant ones: Espectre d'una ona.....  | 76 |
| 5.7.1- Mostreiant la superfície de l'oceà.....   | 77 |
| 5.7.2- Càlcul de l'espectre d'una ona.....   | 78 |

|   |     |
|---|-----|
| 6- Solucions possibles i opció triada.....                              | 80  |
| 6.1- Hardware.....  | 80  |
| 6.1.1- Components necessaris per al disseny.....                        | 80  |
| 6.1.2- Etapa de processament de dades i control.....                    | 80  |
| 6.1.3- Etapa de sensat.....   | 83  |
| 6.1.3.1- Elecció del receptor de GPS.....                               | 83  |
| 6.1.3.2- Elecció dels sensors d'inèrcia.....                            | 84  |
| 6.1.4- Etapa d'alimentació.....   | 86  |
| 6.1.5- Interfície Entrada/Eixida.....                                   | 86  |
| 6.2- Software.....  | 87  |
| 7- Descripció general dels components escollits.....                    | 88  |
| 7.1- El microcontrolador PIC24FJ256GB106.....                           | 88  |
| 7.1.1- Opcions passades i futures: PIC18F46J50 i DSPIC33EP256MU806..... | 89  |
| 7.2- El sensor d'inèrcia ADIS16400.....                                 | 91  |
| 7.2.1- Especificacions temporals del ADIS16400.....                     | 92  |
| 7.2.2- Format de la comunicació amb la IMU i estructura interna.....    | 93  |
| 7.2.3- Calibratge dels diferents sensors de l'ADIS 16400.....           | 96  |
| 7.2.3.1- Calibratge: automàtic o manual?.....                           | 97  |
| 7.2.3.2- Calibrant els giroscopis.....                                  | 97  |
| 7.2.3.3- Calibrant els acceleròmetres.....                              | 98  |
| 7.2.3.4- Calibrant els magnetòmetres.....                               | 100 |
| 7.2.4- Exemple del procés de lectura de l'ADIS16400.....                | 104 |
| 7.3- El receptor de GPS 610-F.....                                      | 107 |
| 7.4- Altres components: <i>Display</i> de 20 caràcters i 2 línies.....  | 108 |
| 8- Esquema elèctric i distribució dels components del sistema.....      | 110 |
| 8.1- Circuit d'Alimentació.....   | 111 |
| 8.2- Circuit de control.....  | 111 |
| 8.3- Interfície Entrades/Eixides.....                                   | 113 |
| 8.3.1- Teclat.....  | 113 |
| 8.3.2- <i>Display</i> LCD.....  | 114 |
| 8.3.3- Connexió amb l'ADIS16400.....                                    | 114 |
| 8.3.4- Connexió amb el receptor de GPS.....                             | 115 |
| 8.3.5- Lector de targetes <i>MicroSD</i> .....                          | 116 |
| 8.3.6- Port USB.....  | 116 |
| 8.4- Distribució dels components en la PCB.....                         | 117 |
| 8.5- Placa prototip: errades en el disseny, contratemps i millores..... | 120 |
| 9- Programació del <i>Firmware</i> del sistema.....                     | 123 |
| 9.1- Fitxer <i>main.c</i> .....   | 123 |
| 9.2- Fitxer <i>menus.c</i> .....  | 126 |
| 9.2.1- Organització bàsica del menú del sistema.....                    | 128 |
| 9.2.2- Menú <i>Datalogger</i> i Llegir IMU.....                         | 129 |
| 9.2.3- Menú Obtenir posició GPS i Calibrar IMU.....                     | 131 |
| 9.2.4- Restaurar calibratge de la IMU i Menú Extraure/Muntar uSD.....   | 132 |
| 9.2.5- Ajustar RTCC i Informació del Sistema.....                       | 133 |
| 9.3- Fitxer <i>lcd.c</i> .....  | 134 |
| 9.4- Fitxer <i>gps.c</i> .....  | 135 |
| 9.4.1- Funció <i>captura_gprmc()</i> .....                              | 135 |
| 9.4.2- Funció <i>mostrar_gprmc(char freq)</i> .....                     | 137 |
| 9.4.3- Funció <i>rellotge_gps()</i> .....                               | 139 |
| 9.4.4- Rutina d'atenció a la interrupció de la UART.....                | 140 |

|  |     |
|--|-----|
| 9.5- Fitxer <i>adis16x00.c</i> .....   | 141 |
| 9.5.1- Funció <i>Adis_Auto_Test()</i> .....  | 142 |
| 9.5.2- Funció <i>Adis_DfltCfg()</i> .....  | 143 |
| 9.5.3- Funcions de lectura.....  | 145 |
| 9.5.3.1- Funció <i>Read_Adis(unsigned char adress)</i> .....   | 145 |
| 9.5.3.2- Funció <i>Adis_Burst_Read()</i> .....   | 147 |
| 9.5.4- Funcions d'escriptura.....  | 149 |
| 9.5.4.1- Funció <i>Write_Adis(unsigned char adress, int data)</i> .....  | 149 |
| 9.5.4.2- Funció <i>Write_Adis_16bit (unsigned char adress, int data)</i> .....   | 150 |
| 9.5.5- Funció <i>Adis_Calibracio_22c()</i> .....   | 151 |
| 9.5.6- Funció <i>Adis_Manual_Calibration()</i> .....   | 152 |
| 9.5.7- Funció <i>Adis_Sensor_Calibration(unsigned char sensor)</i> .....   | 154 |
| 9.5.8- Funció <i>Read_Adis_RT(char mode, char freq)</i> .....  | 158 |
| 9.5.9- Funció <i>traduir_lectura_adis(unsigned char sensor, signed int lectura, unsigned char mode)</i> .....                          | 159 |
| 9.5.10- Funció <i>escriure_lectura_traduida_adis(char refresc)</i> .....   | 161 |
| 9.6- Fitxer <i>rtcc.c</i> .....  | 164 |
| 9.6.1- Funció <i>RtccInit()</i> .....  | 164 |
| 9.6.2- Funció <i>Mostrar_rtcc()</i> .....  | 165 |
| 9.6.3- Funcions <i>PIC24RTCCGetTime()</i> i <i>PIC24RTCCGetDate()</i> .....  | 169 |
| 9.6.4- Funcions <i>PIC24RTCCSetTime (int weekDay_hours, int minutes_seconds) i PIC24RTCCSetDate (int xx_year, int month_day)</i> ..... | 170 |
| 9.7- Fitxer <i>funcions_externes.c</i> .....   | 170 |
| 9.8- Llibreries d'aplicació de <i>Microchip</i> .....  | 171 |
| 9.8.1- Llibreria <i>Memory Disk Drive File System</i> .....  | 171 |
| 9.8.1.1- Edició de l'arxiu <i>FSConfig.h</i> .....   | 172 |
| 9.8.1.2- Edició de l'arxiu <i>SD-SPI.h</i> .....   | 173 |
| 9.8.1.3- Funcions utilitzades de la llibreria <i>MDD File System</i> .....   | 173 |
| 9.8.2- Llibreries USB.....   | 174 |
| 9.8.2.1- Edició de l'arxiu <i>usb_device.h</i> .....   | 175 |
| 9.8.2.2- Edició de l'arxiu <i>usb_device.c</i> .....   | 175 |
| 9.8.2.3- Funcions utilitzades de les llibreries USB.....   | 176 |
| 9.9- <i>Firmware</i> : coses a millorar i a implementar.....   | 177 |
| 10- Implementació física i pràctica del sistema.....   | 179 |
| 10.1- Implementació inicial amb un PIC18F46J50.....  | 179 |
| 10.2- Implementació final amb un PIC24FJ256GB106.....  | 183 |
| 11- Prova experimental del sistema.....  | 186 |
| 12.- Conclusions i últimes paraules.....   | 193 |
| 13- Bibliografia.....  | 199 |
| 13.1- Localització d'un objecte mitjançant satèl·lits.....   | 199 |
| 13.2- Sensors de moviment.....   | 200 |
| 13.3- Sistemes de Navegació Inercial.....  | 201 |
| 13.4- Exemple d'aplicació.....   | 202 |
| 13.5- Descripció general dels components escollits.....  | 202 |
| 13.6- Esquema elèctric i distribució dels components del sistema.....  | 203 |
| 13.7- <i>Firmware</i> .....  | 203 |
| 13.8- Implementació pràctica.....  | 204 |
| 13.9- Prova experimental del sistema.....  | 204 |



|  |     |
|--|-----|
| Apèndixs.....  | 205 |
| Apèndix 1: Matisacions i errades en la fulles de característiques.....   | 205 |
| ADIS 16400/16405 <i>Tri-Axis Inertial Sensor with Magnetometer</i> ..... | 205 |
| GPS-610F <i>GPS Receiver with Ext Antenna</i> .....                      | 206 |
| Apèndix 2: Conversacions amb personal d' <i>Analog Devices</i> .....     | 207 |
| 3 d'octubre del 2011.....  | 207 |
| 19 d'octubre del 2011.....   | 208 |
| 20 d'octubre del 2011.....   | 209 |
| 24 d'octubre del 2011.....   | 211 |
| 25 d'octubre del 2011.....   | 213 |
| 26 d'octubre del 2011.....   | 214 |
| 8 de Novembre del 2011.....  | 216 |
| 9 de Novembre del 2011.....  | 218 |
| 27 de Juny del 2012.....   | 219 |
| 14 d'agost del 2012.....   | 219 |
| 15 d'agost del 2012.....   | 221 |
| Apèndix 3: Esquemàtic de la implementació amb el PIC18F46J50.....        | 222 |

## 1- Introducció i objectiu del projecte

L'objectiu d'aquest Treball de Final de Carrera és l'estudi de viabilitat, el disseny i la implementació amb components comercials d'un sistema electrònic de control d'estabilitat i posicionament d'objectes, com per exemple, d'una plataforma flotant. Aquest disseny també intentarà deixar portes obertes a noves funcionalitats per a d'aquesta manera fer un ús més eficient dels recursos disponibles.

Per a aconseguir aquest objectiu, el nostre sistema utilitzarà sensors d'inèrcia i un receptor de GPS. Aquests dispositius estaran connectats a un microcontrolador el qual serà l'encarregat de recollir les dades proporcionades per aquests, tractar-les adequadament per al seu posterior emmagatzemament en una memòria no volàtil, com puga ser una targeta micro SD, o bé per a ser transmeses a un altre dispositiu utilitzant, per exemple, el bus USB.

Tot açò serà possible una vegada haguem estudiat alguns dels diferents mètodes de localització i posicionament d'objectes, coneguem quins dispositius electrònics són més útils per a implementar eixos mètodes i quin és el resultat experimental del conjunt una vegada s'haja creat la placa prototip del sistema, la qual no serà una altra cosa que la materialització física del disseny electrònic creat durant l'anàlisi del funcionament dels diferents components que formaran el sistema.

## 2- Localització d'un objecte mitjançant satèl·lits

### 2.1- Introducció: La localització per satèl·lit i els sistemes de coordenades

Els sistemes de localització per satèl·lit permeten definir la situació d'un objecte animat o inanimat sobre un sistema de referència. Per a poder determinar la localització d'un objecte és necessari detallar o bé les coordenades geogràfiques de latitud i longitud, o bé les coordenades X i Y del sistema UTM<sup>1</sup>. Tal i com veurem en la secció de Sentències NMEA, els receptors de GPS convencionals donen les dades de localització en el format de coordenades geogràfiques, raó per la que en aquesta secció sols donarem una descripció resumida del sistema de coordenades geogràfiques.

Així doncs, les coordenades geogràfiques permeten determinar la posició de cada punt a la terra utilitzant un sistema de coordenades esfèriques i dos coordenades anomenades longitud i latitud. Aquest sistema pren com a referència l'eix de rotació de la terra que uneix els dos pols i permet el posicionament en tres dimensions.

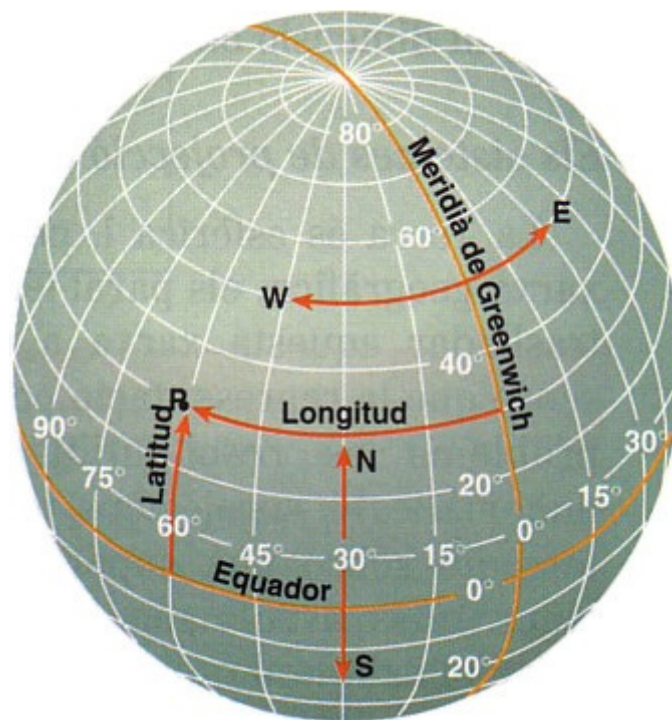


Figura 1: Representació de la longitud i la latitud.

Tal i com podem veure en la figura 1, la longitud pot anar en direcció Est o Oest i la latitud en direcció Nord o Sud. Per a facilitar el procés de visualització de la localització es creen unes línies imaginàries que van del pol Nord al pol sud (Meridians) i unes altres que van de l'Est a l'Oest (Paral·lels) pel que, al ser perpendiculars als meridians, tots els punts d'un mateix paral·lel tindran la mateixa latitud. Al paral·lel que s'agafa de referència, és a dir, al paral·lel 0, se li dona el nom d'Equador i divideix el globus en hemisferi Nord i hemisferi Sud. En canvi, al meridià 0 se li dona el nom de meridià de *Greenwich*.

Una vegada explicat açò, podem definir la latitud com la distància existent entre un punt qualsevol i l'equador. Aquesta distància variarà de 0° a 90° i vindrà especificada respecte al hemisferi sobre el que s'ha calculat (N o S). En canvi, definirem la longitud com la distància entre un punt i el meridià de *Greenwich*, variarà de 0° a 180° i, al igual que en la latitud, s'especificarà una orientació la qual

<sup>1</sup> Universal Transversa Marcator

serà Est sí el meridià està a l'orient del meridià de referència, o Oest sí està localitzat a l'occident del mateix. Cal destacar que, tant el pol Nord com el pol Sud no tenen longitud i que aquest sistema de coordenades no té en compte el fet que la terra no és totalment esfèrica, sinó que s'aproxima a la d'una el·lipsoide, pel que un grau de longitud o de latitud és diferent en diferents punts geogràfics, cosa que fa necessària la realització de correccions sobre les coordenades obtingudes sí es vol millorar la precisió.

## **2.2- Els sistemes de navegació NAVSTAR-GPS, GLONASS i GALILEU**

Com dèiem en la secció anterior, els sistemes de navegació donen la informació de localització respecte un sistema de coordenades. Esta informació és transmesa pels mateixos a receptors que estan dissenyats de tal manera que són capaços d'analitzar i interpretar aquestes dades.

Actualment hi han tres sistemes de localització però sols dos estan en funcionament: el NAVSTAR-GPS i el GLONASS. S'espera que el tercer sistema de localització, el GALILEU, entre en funcionament cap el 2014, però degut als constant retards per problemes tècnics, és possible que la seva posta en marxa siga passada eixa data. Com ara veurem, tots tres sistemes tenen característiques molt semblants:

- **El sistema NAVSTAR-GPS** (*Navigation Signal Timing and Ranging Positioning System*) més conegut com GPS<sup>2</sup>, és un sistema de localització creat pel departament de defensa americà i disposa de 27 satèl·lits, on 3 dels quals són de reserva. Aquests 27 satèl·lits estan situats a uns 20200 Km d'alçada, tarden poc menys de 12 hores en realitzar una volta a la terra i estan repartits en 6 plans orbitals els quals tenen una inclinació de 55° respecte l'equador. Tot açò fa que hi hagen com a mínim 4 satèl·lits per orbita i que com a mínim 4 d'aquests siguen visibles des de qualsevol punt de la terra.

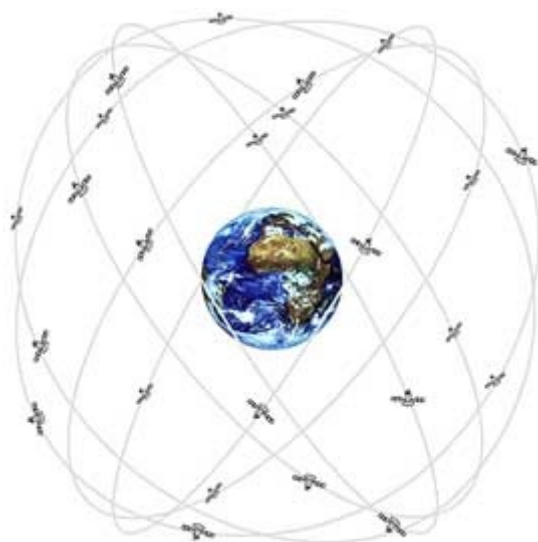


Figura 2: Òrbites dels satèl·lits que formen el sistema NAVSTAR-GPS

Tots els satèl·lits que formen el sistema porten cadascun 4 rellotges atòmics, 2 d'ells de reserva, per a poder donar el temps en format UTC<sup>3</sup> amb una precisió de 60ns a 5ns. Aquesta precisió tan elevada és la que fa possible que els satèl·lits es puguin utilitzar pel posicionament i la navegació ja que el càlcul de la posició es fa mesurant la diferència de temps entre un missatge i el següent.

<sup>2</sup> *Global Positioning System*

<sup>3</sup> *Universal Time Coordinated*

● **El sistema GLONASS** (*Global'naya Navigatsionnaya Sputnikovaya Sistema*) és un sistema de localització creat pel govern rus inicialment per motius militars però que posteriorment ha quedat obert a l'ús civil amb la mateixa precisió de la que disposava l'ús militar. A l'igual que el sistema GPS, en el GLONASS també hi han 27 satèl·lits i 3 d'ells són de reserva. Els altres 24 estan situats a una altura de 19100 Km i disposats en una inclinació de  $64.8^\circ$  respecte a l'equador en tres plans orbitals en el que arriben a hi haure 8 satèl·lits. Les principals diferències respecte al sistema Americà són dos: es necessiten uns 40 minuts menys per a completar una volta al planeta i la precisió a l'hora de localitzar un objecte situat prop del pol nord és major per la disposició orbital dels satèl·lits d'aquest sistema.



Figura 3: Òrbites dels satèl·lits que formen el sistema GLONASS

● **El sistema GALILEU** és el nom que se li va a donar al sistema europeu. Aquest sistema encara està en desenvolupament i va nàixer amb l'objectiu d'eliminar la dependència que té el continent amb els sistemes rus i americà.

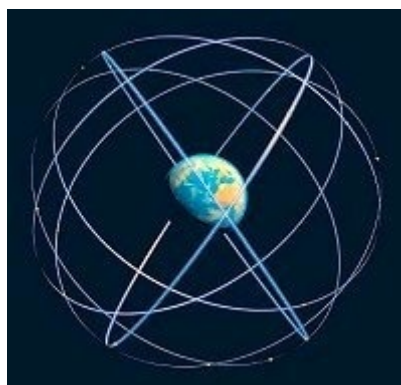


Figura 4: Òrbites dels satèl·lits que formen el sistema GALILEU

Tècnicament contarà amb 30 satèl·lits repartits en 8 òrbites i en cada òrbita es preveu que hi hauran un mínim de 3 satèl·lits, els quals estaran a 23616 Km de la terra, presentaran una inclinació de  $56^\circ$  respecte a l'equador i necessitaran poc més de 14 hores per a completar una volta al planeta. Tot açò permetrà millorar la precisió de la localització pel que, indirectament, es milloraran servicis com el SAR<sup>4</sup> permetent d'aquesta manera que els missatges d'auxili arriben quasi instantàniament quan actualment necessiten al voltant d'una hora.

4 Search And Rescue, Cerca i Rescat

### 2.3- Conceptes bàsics

Per a entendre el funcionament d'un sistema de localització s'han de tenir en compte conceptes com la triangulació o les possibles fonts d'error que poden afectar al sistema a l'hora de calcular correctament la localització de l'objecte.

Així doncs, per a que un sistema de posicionament done una localització correcta, és necessària una triangulació entre els diferents satèl·lits del sistema ja que la posició del objecte en qüestió es dedueix a partir de la posició dels satèl·lits que entren en joc durant la comunicació amb el receptor. Cal destacar que, tot i que el terme “triangulació” pot induir a pensar que entren en joc 3 satèl·lits, en realitat són necessàries les senyals de 4 satèl·lits per a que la triangulació es realitzi correctament. Durant aquest procés, els satèl·lits involucrats amb la comunicació amb el receptor, que fins que no entre en funcionament el sistema GALILEU no serà major de 12, envien a la terra en la freqüència d'ús civil 1575.42 MHz el temps indicat pels seus rellotges atòmics i els paràmetres per determinar la posició exacta. Com que amb el missatge s'envia l'hora i es coneix la velocitat de propagació de les ones de ràdio, el receptor pot calcular la distància entre ell i el satèl·lit i enviar-la de volta per a que el satèl·lit pugui, a partir del temps de viatge de la senyal de tornada, calcular en quin punt de l'esfera que té com a centre el mateix satèl·lit es troba el receptor, pues el radi d'aquesta esfera és igual a la distància entre el receptor i el mateix satèl·lit tal i com s'aprecia en la figura 5:

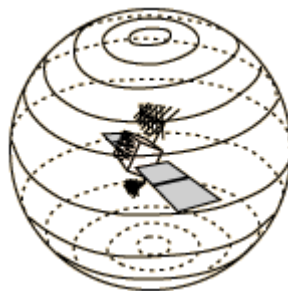


Figura 5: Satèl·lit com a el centre de l'esfera

Sí aquest procés el repeteixen els tres o quatre satèl·lits que estan a la vista del receptor, el resultat final acabarà sent la posició del receptor ja que a partir d'unir els punts comuns de les esferes de cada satèl·lit i, descartant els punts no comuns, apareix una circumferència al centre de la qual serà on probablement es trobe el receptor. Tot açò ho podem visualitzar millor sí observem la figura 6:

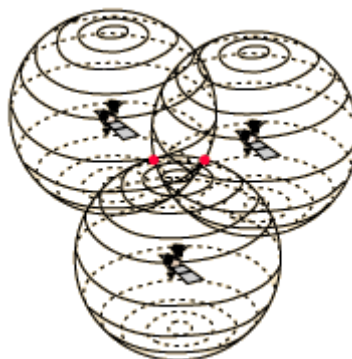


Figura 6: Tres esferes reduceixen la posició a dos punts diferents.

Com els receptors no utilitzen un rellotge amb tanta precisió com la que proporciona un rellotge atòmic cal fer una quarta mesura. Aquesta última mesura validarà o no la posició resultant d'unir les tres esferes amb l'última: sí esta nova esfera creua pel punt calculat anteriorment, la posició és correcta. De no ser així, el receptor és capaç de fer una correcció a partir de les dades donades per aquesta quarta esfera de manera que s'obté una posició més exacta. Cal dir que esta no és l'única

correcció que es realitza pues hi ha un nombre considerable de factors que poden introduir errors en el càlcul de la posició. Els més importants són els següents:

- **Desfasaments entre els rellotges dels satèl·lits i dels receptors**: en el cas dels rellotges dels satèl·lits és menor i es poden reduir amb la monitorització dels mateixos. En el cas del dels receptors, la desviació és major pel que solen corregir-se a partir de la solució d'un sistema de 4 equacions de 4 incògnites.
- **La ionosfera**: aquesta capa de l'atmosfera retarda les ones de ràdio un nombre de segons que depèn del nombre d'ions/m<sup>3</sup> i de l'activitat ionosfèrica del moment.
- **Fenòmens meteorològics**: de tots és sabut l'efecte dels mateixos sobre les comunicacions de ràdio pel que no entrarem en detalls, doncs bàsicament un sistema de localització per satèl·lit és un sistema de transmissió d'informació per ràdio.
- **Errors d'òrbita**: són els produïts pels efectes gravitatoris del sol o la lluna, pel que les estacions terrestres hauran de controlar la posició dels satèl·lits i ajustar-la.
- **Interferències elèctriques imprevistes**: aquestes interferències poden produir, entre altres efectes, càlculs erronis de posició que poden arribar a no ser detectats pel sistema sí la magnitud d'aquestes interferències és petita.
- **Errors "multi camí"**: són els errors produïts pel rebot de la senyal ja que sí l'ona de ràdio es troba amb un obstacle, pot rebotar en ell i fer que al receptor li arribi la senyal original i la rebotada. Per a minimitzar els efectes d'aquest error, els receptors solen utilitzar antenes dissenyades especialment per a evitar aquest efecte i, per norma general, solen incloure DSPs avançats per a eliminar aquest tipus d'interferència.
- **La geometria de les circumferències creades pels satèl·lits o DOP<sup>5</sup>**: les circumferències marcades pels satèl·lits no són una línia determinada i exacta, sinó que és una zona difusa i en lloc de creuar-se generant un punt concret, generen una àrea pel que, quan més prop estiguen aquests, més precisa serà l'àrea.
- **Receptor GPS**: les especificacions i paràmetres tècnics dels receptors també són una font d'error ja que, tant el nombre de canals que tenen, com la precisió del processador matemàtic que fa els càlculs, afecten al resultat final.
- **Disponibilitat Selectiva o SA<sup>6</sup>**: és un sistema de distorsió creat pels departaments de defensa de certs països que distorsiona les senyals de navegació amb l'objectiu d'empitjorar l'exactitud del posicionament quan el servei de GPS és utilitzat per un civil. Teòricament aquests sistemes van ser desactivats per EUA i Rússia fa anys per a que els civils pogueren gaudir totalment de l'exactitud d'aquests sistemes de localització però qui sap...

<sup>5</sup> *Dilution Of Precision*, Dissolució de la Precisió

<sup>6</sup> *Selective Availability*, Disponibilitat Selectiva

## 2.4- GPS Diferencial i Sistemes d'Augment Basat en Satèl·lits (SBAS)

El GPS Diferencial o DGPS és un sistema que proporciona correccions de les dades GPS que reben els receptors dels satèl·lits. Aquestes correccions provenen de receptors de referència i permeten una major exactitud en el càlcul de la posició arribant en alguns casos a ser de sols centímetres. Aquests receptors de referència estan localitzats en uns punts de coordenades coneguts i poden ser des d'una simple antena fins un satèl·lit geostacionari. Com aquests també reben les dades de GPS, es poden utilitzar per a calcular l'error entre la posició real i la calculada, eliminant d'aquesta manera molts errors, d'ací el nom d'aquest sistema.

Com hi han molts dispositius els quals poden ser utilitzats per a fer els càlculs de DGPS, hi han també moltes maneres d'obtenir aquestes dades, però els més habituals solen ser els servicis RDS de les emissores de ràdio comercials i les dades descarregades d'Internet.

La finalitat dels Sistemes d'Augment Basat en Satèl·lits (SBAS) és molt pareguda a la del DGPS, sols que, a més, permeten calcular la integritat de les dades GPS. La manera d'aconseguir-ho és molt pareguda al DGPS ja que també es parteix d'estacions de referència perfectament localitzades, però amb la diferència de poder enviar les dades corregides als usuaris utilitzant la mateixa freqüència utilitzada per un receptor GPS d'ús civil. Actualment els principals sistemes SBAS són:

- **WAAS o *Wide Area Augmentation System***: és un sistema d'ús exclusiu aeronàutic dels EUA. Permet que l'exactitud del GPS passe dels 100 metres a 7-8. Aquesta millora en l'exactitud s'aconsegueix utilitzant 25 estacions de referència les quals estan constantment calculant una llista de satèl·lits visibles en eixe instant, una correcció exacta del càlcul de la distància de cada satèl·lit i una referència de temps de cada correcció. Totes aquestes dades es transmeten a una central que les analitza i compara per a determinar la validesa i integritat de les mateixes. Una vegada fet açò, aquesta informació s'envia a un satèl·lit geostacionari per a que la propague a tots els receptors de l'àrea a la qual dona servici el satèl·lit en qüestió.

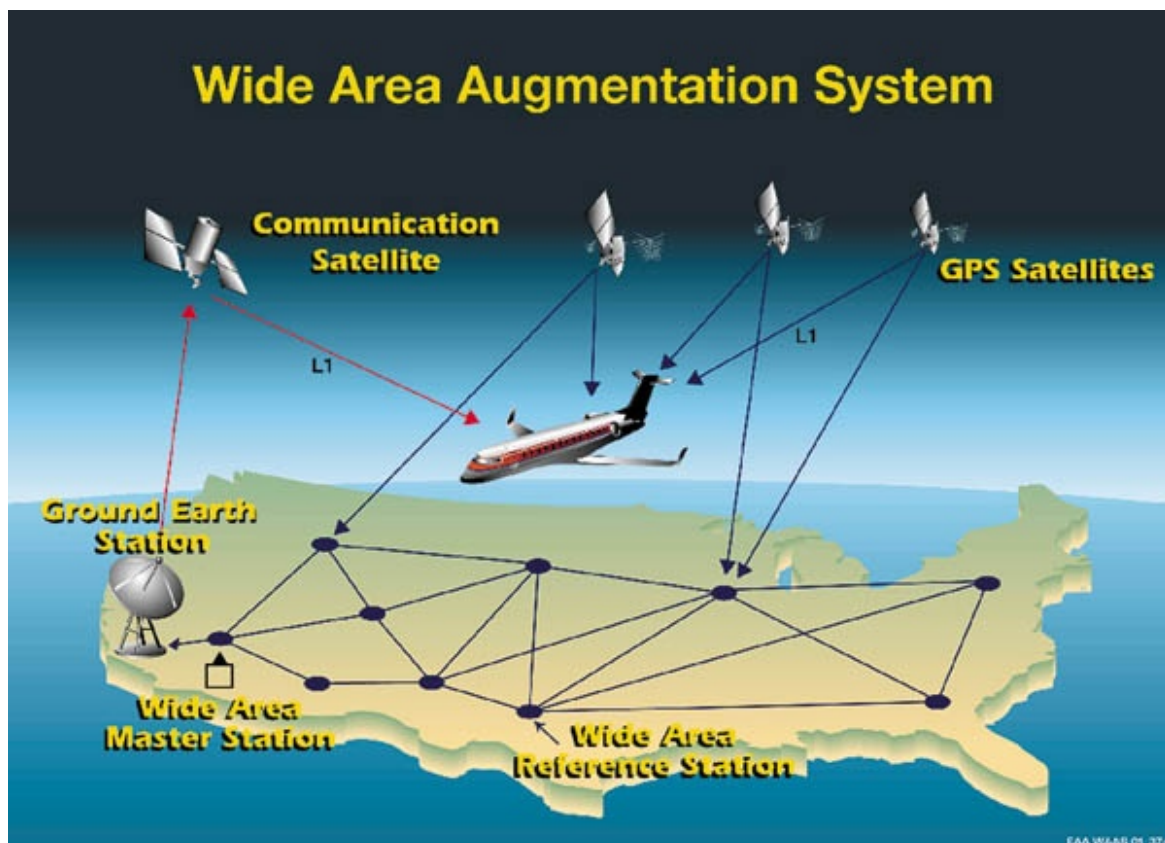


Figura 7: Esquema de funcionament del sistema WAAS



- **MSAS o Multifuncional Satellite Augmentation System:** Aquest sistema d'augment per satèl·lit és procedent de Japó i, a l'igual que l'anterior, és exclusivament d'ús aeronàutic. La diferència està en la precisió pues en aquest pot arribar a ser de 1.5 a 2 metres.
- **GAGAN o GPS Aided Geo Augmented Navigation:** És un sistema d'origen Indi el qual entrarà en funcionament en uns mesos. Una vegada actiu serà d'ús exclusiu aeronàutic però també permetrà ser utilitzat pels servicis d'emergència. Constarà de 18 estacions i permetrà als pilots no sols tenir una precisió d'uns tres metres, sinó que, entre altres coses, millorarà l'accés a aeroports i l'ús de l'espai aeri en qualsevol condició climàtica, augmentant d'aquesta manera la seguretat o l'eficiència i eficàcia dels paràmetres predefinitos que s'utilitzen, per exemple, a l'aterrar.
- **EGNOS o European Geostacionary Navigation Overlay Service:** Com el nom indica, és un sistema Europeu que es diferencia dels altres perquè permet que el pugui utilitzar qualsevol usuari sense importar si és un avió, un vaixell, un servei d'emergència, una persona, etc. El seu funcionament és semblant al del WAAS ja que es parteix d'estacions de transmissió que estan perfectament localitzades i que s'encarreguen de transmetre les dades de correcció a tres satèl·lits geostacionaris de comunicacions del INMARSAT. Gràcies a les dades transmises per aquestes estacions, com que mesuren l'error dels sistemes GPS i GLONAS, la precisió pot reduir-se fins als 2 metres.

## 2.5- Missatges de navegació GPS

Els missatges de navegació GPS consisteixen en una transmissió continua d'informació des de els satèl·lits a una velocitat de 50bps. Aquests missatges contenen informació sobre la seva posició orbital exacta, posició aproximada de tots els satèl·lits, estat i hora del sistema junt altres dades per a corregir-la. Com que amb 50 bits no hi ha prou per a codificar tota aquesta informació, el que es fa és agrupar-la en trames de manera que hi han fins 25 trames diferents de longitud 1500 bits. Com es transmeten a una velocitat de 50bps, el temps necessari per a enviar una trama és de 30 segons, com hi han 25 trames diferents, el temps de transmissió d'un missatge és d'uns 12 minuts i mig.

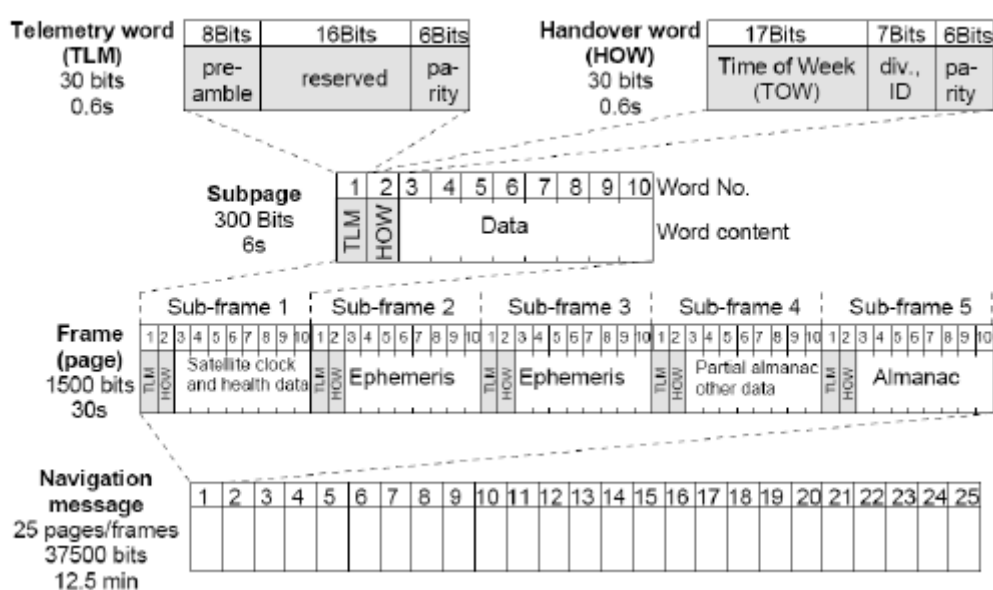


Figura 8: Estructura dels missatges de Navegació GPS.

Com podem veure en la figura 8, les trames es divideixen en 5 sub-trames de 300 bits i estes, al mateix temps, es divideixen en paraules de 30 bits. Les sub-trames 1, 2 i 3 contenen les efemèrides d'alta exactitud i les dades d'*off-set* del rellotge. En concret, la primera sub-trama conté els coeficients del polinomi de segon grau que guarda l'*off-set* del rellotge i les altres dos transporten els paràmetres orbitals. Les sub-trames 4 i 5 contenen informació del almanac i informació sobre l'estat dels sistemes del satèl·lit.

## **2.6 Un protocol per a la transmissió de missatges GPS: NMEA-0183**

Com sol ser habitual en les comunicacions, per a que les transmissions dels missatges siguin satisfactòries, s'ha de seguir un protocol. El cas del GPS no és una excepció i podem trobar tant protocols definits per organismes oficials com protocols definits pels propis fabricants de receptors en un intent d'augmentar el rendiment i la funcionalitat dels mateixos i, d'eixa manera, posicionar-se per davant dels seus competidors. Afortunadament per a nosaltres, tot i la gran varietat de protocols disponibles avui en dia, tots els receptors són compatibles amb el protocol NMEA-0183 pel que no serà necessari analitzar els altres protocols.

El NMEA-0183 és un protocol de compliment voluntari creat per la *National Marine Electronics Association*, una associació no lucrativa creada per fabricants, distribuïdors, comerciants, institucions educatives i altres interessats en "l'electrònica marina". Aquest estàndard defineix un *interface* elèctric i un protocol de dades per a la comunicació entre instrumentació d'equips d'ús marítim. La primera versió va ser alliberada en Març de 1983 i, des deleshores, a anat actualitzant-se regularment de manera que actualment la revisió més actual del protocol és la 4.0, la qual es caracteritza per elevar la velocitat de transmissió a 38400 *bps* (bits per segon) i afegir 59 noves frases a l'anterior revisió del protocol.

Des de el punt de vista elèctric, els dispositius compatibles amb el protocol poden configurar-se com a transmissor (*talker*), receptor (*listener*) o una combinació dels dos. La comunicació entre ells es realitza mitjançant l'ús d'uns missatges formats per entre 20 i 80 caràcters ASCII els quals s'envien als receptors utilitzant una transmissió serie asíncrona configurada a una velocitat de 4800 bits per segon, 8 bit de dades amb  $d7=0$ , 1 bit de parada, sense paritat ni *handshake* pel que no hi ha control d'errors, i, per tant, no es pot garantir la correcta arribada del missatge. Destacar que, en un mateix circuit, pot hi haure més d'un *listener* i que el protocol no obliga al ús de cable blindat ni d'un connector específic. Per a millorar la immunitat del sistema al soroll l'estàndard recomana que el *talker* siga conforme al estàndard RS-422 ja que aquest, al ser un sistema diferencial, presenta major immunitat al soroll que els estàndards que utilitzen en la transmissió nivells de tensió TTL o CMOS.

Respecte al format de dades del protocol, destaca el fet que totes les dades transmeses tenen forma de frases, o siga que el que en realitat s'està transmetent són exclusivament caràcters ASCII imprimibles, pues a part d'estos, sols es permeten els dos caràcters que s'utilitzen per a indicar el fi de la frase: el de canvi de línia (*Carriage Return*) i el de línia en blanc (*Line Feed*). Així doncs, el protocol distingeix tres tipus de frases o sentències:

- **Tipus *Talker***: les envia el *talker* i tenen el format  $\$ttsss,d1,d2,\dots\langle CR\rangle\langle LF\rangle$ , on *tt* identifica al *talker* i *sss* és l'indicador de sentència. Per norma general, tot el que hi ha entre l'identificador de frase i els caràcters de fi de frase són camps de dades, encara que de forma opcional es pot afegir un camp de *checksum* el qual és el resultat de fer una OR exclusiva a tots els caràcters entre el caràcter \$ i aquest camp. Tots aquests camps estan separats entre ells per una coma i la grandària d'aquestes frases pot arribar a ser de 80 caràcters sense incloure ni el \$ ni la combinació CR/LF. Els camps de data queden definits segons el tipus de frase.

•**Tipus Propietari**; al ser un estàndard en el que hi han implicats diversos fabricants, es permet l'ús de frases específiques creades per un fabricant pels seus equips. Aquestes frases comencen amb una P a la que segueixen 3 lletres que especifiquen al fabricant. Tot el que va després d'estes lletres ja depèn del fabricant pel que NMEA no dona més especificacions.

•**Tipus Petició (Query)**; són les que utilitza un *listener* per a demanar a un *talker* una frase en particular. El seu format és \$tllQ,sss[CR][LF], on els dos primers caràcters són l'adreça assignada al *listener*, els dos següents identifiquen al *talker* i els tres finals contenen el identificador de la frase que ha demanat el *listener*.

Dins d'estes dos classes de sentències l'estàndard defineix una gran varietat de frases. Aquesta varietat permet obtenir tant informació bàsica com complexa, doncs inclou informació tan "simple" com conèixer l'hora i data actual, passant per coses més "normals" com conèixer la distància d'un punt a un altre fins a arribar a informació que pot ser més "crítica", com és l'estat del sistema. Com sols ens interessen les frases relacionades amb el GPS, sols ens centrarem en relacionades amb aquest dispositiu.

### 2.6.1- Frases específiques de NMEA-0183 per a receptors GPS

•**GSV - GPS Satellite in View**; Mostra el nombre de satèl·lits a la vista, nombre PRN, els angles d'elevació i d'azimut i la relació S/N<sub>0</sub>. Es transmeten detalls de 4 satèl·lits per missatge, pel que la informació dels restants es transmet en els successius missatges de tipus GSV. Conté 8 camps de dades l'organització dels quals és la següent:

```
$GPGSV,x,x,xx,xx,xx,xxx,xx,...,xx,xx,xxx,xx *hh<CR><LF>
```

1 2 3 4 5 6 7 4 5 6 7 8

Exemple:

```
$GPGSV,3,1,09,28,81,225,41,24,66,323,44,20,48,066,43,17,45,336,41*78<CR><LF>
```

```
$GPGSV,3,2,09,07,36,321,45,04,36,257,39,11,20,050,41,08,18,208,43*77<CR><LF>
```

Figura 9: exemple de frase GPGSV

| Camp | Nom                   | Exemple | Descripció   |
|------|-----------------------|---------|--|
| 1    | Nº de missatges       | 3       | Nombre de missatges GSV que es transmetran (1 a 3)                         |
| 2    | Missatge nº           | 1       | Indica l'ordre del missatge GSV rebut                                      |
| 3    | Satèl·lits a la vista | 9       | Total de satèl·lits a la vista (0 a 12)                                    |
| 4    | ID del satèl·lit      | 28      | IGPS 01 a 32, SBAS 33 a 54   |
| 5    | Elevació              | 81      | Elevació del satèl·lit en graus (000 a 359)                                |
| 6    | Azimut                | 225     | Angle azimut del satèl·lit en graus (000 a 360)                            |
| 7    | SNR                   | 41      | S/N <sub>0</sub> en dB (00 a 99) 00 quan no s'està en mode de localització |
| 8    | Checksum              | 78      |  |

Taula 1: Descripció camps frase GPGSV

● *VTG - Course Over Ground and Ground Speed*; Com l'acrònim indica, rumb i velocitat actual respecte a la superfície. Aquesta sentència té 6 camps de dades i la següent estructura:

GPVTG,x.x,T,x.x,M,x.x,N,x.x,K,a\*hh<CR><LF>

1 2 3 4 5 6

**Exemple:**

\$GPVTG,89.68,T,,M,0.00,N,0.0,K,A\*5F<CR><LF>

Figura 10: exemple de frase GPVTG

| Camp | Nom       | Exemple | Descripció  |
|------|-----------|---------|---|
| 1    | Rumb      | 89.68   | En graus (000.0 a 359.9)  |
| 2    | Rumb      |         | Rumb magnètic en graus (000.0 a 359.9)  |
| 3    | Velocitat | 0.00    | Velocitat sobre la superfície en nus (000.0 a 999.9)  |
| 4    | Velocitat | 0.00    | Velocitat sobre la superfície en km/h (0000.0 a 1800.0)   |
| 5    | Mode      | A       | N: Dades no vàlides<br>A: Mode Autònom<br>D: Mode Diferencial<br>E: Mode Estimació<br>M: Mode Manual<br>S: Mode Simulació |
| 6    | Checksum  | 5F      |   |

Taula 2: Descripció camps frase GPVTG

● *ZDA - Time and Data*; s'explica per ella mateixa. Conté 7 camps de dades els quals ens proporcionen informació de tipus horària i de calendari. La seva estructura és la següent:

\$GPZDA, hhmmss.sss,dd,mm.yyyy, , ,xxx<CR><LF>

1 2 3 4 5 6 7

**Exemple:**

\$GPZDA,104548.04,25,03,2004,,,\*6C<CR><LF>

Figura 11: exemple de frase GPZDA

| Camp | Nom            | Exemple   | Descripció   |
|------|----------------|-----------|--|
| 1    | Temps UTC      | 104548.04 | Hora UTC (format 24h)                                    |
| 2    | Temps UTC: Dia | 25        | Dia (01 a 31)  |
| 3    | Temps UTC: Mes | 03        | Mes (01 a 12)  |
| 4    | Temps UTC: Any | 2004      | Any (4 dígits)   |
| 5    |                |           | Hora de la zona local; 0 sí no hi ha eixida del receptor |
| 6    |                |           | Com l'anterior, però amb els minuts                      |
| 7    | Checksum       | 60        |  |

Taula 3: Descripció camps frase GPZDA

● *CGA - Global Positioning System Fix Data*; Temps, posició i dades relacionades amb la fixació de posició d'un receptor GPS. La frase conté 13 camps de dades i presenta la següent estructura:

```
$GPGGA,hhmmss.sss,ddmm.mmmm,a,dddmm.mmmm,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh<CR><LF>
```

1 2 3 4 5 6 7 8 9 10 11 12 13

**Exemple:**

```
$GPGGA,060932.448,2447.0959,N,12100.5204,E,1,08,1.1,108.7,M,,,0000*0E<CR><LF>
```

Figura 12: exemple de frase GPGGA

| Camp | Nom                       | Exemple       | Descripció   |
|------|---------------------------|---------------|--|
| 1    | Temps UTC                 | 060932448.448 | Hora UTC de la posició (format 24h)  |
| 2    | Latitud                   | 2447.0959     | Latitud en format <i>ddmm.mmmm</i>   |
| 3    | Indicador N/S             | N             | Indicador latitud en l'hemisferi   |
| 4    | Longitud                  | 12100.5204    | Longitud en format <i>dddmm.mmmm</i>   |
| 5    | Indicador E/W             | E             | Indicador de longitud en l'hemisferi   |
| 6    | Indicador de qualitat GPS | 1             | 0: no disponible<br>1: posició fixada, mode SPS<br>2: posició fixada, mode DGPS<br>3: posició fixada, mode PPS<br>4: Mode RTKi<br>5: Mode RTKF<br>6: Mode d'aproximació<br>7: Mode manual<br>8: Mode simulació |
| 7    | Nº satèl·lits utilitzat   | 08            | Nº de satèl·lits en ús (0-12)  |
| 8    | HDOP                      | 1.1           | <i>Dilution Of Precision</i> horitzontal (00.0 – 99.9)   |
| 9    | Altitud                   | 108.7         | Altitud sobre el nivell del mar  |
| 10   | <i>Geoid separation</i>   |               | En metres i d'acord a l'el·lipsoide WGS-84   |
| 11   | “Edat” DGPS               |               | Data de l'última actualització de les dades DGPS   |
| 12   | Id. de la estació DGPS    | 00000         | De 0000 a 1023. Nul sí no s'utilitza el DGPS.  |
| 13   | <i>Checksum</i>           | 0E            |  |

Taula 4: Descripció camps frase GPGGA

●RMC - Recommended Minimum Specific GPS/Transit Data: Dades de temps, data, posició, rumb i velocitat proporcionades pel receptor de navegació. La frase conté 13 camps de dades i presenta la següent estructura:

```
$GPRMC,hhmmss.sss,A,dddmm.mmmm,a,dddmm.mmmm,a,x.x,x.x,ddmmyy,x.x,a,a*hh<CR><LF>
```

1      2   3            4      5    6 7   8    9   10 11 12   13

**Exemple:**

```
$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,,A*25<CR><LF>
```

Figura 13: exemple de frase GPRMC

| Camp | Nom                           | Exemple    | Descripció  |
|------|-------------------------------|------------|---|
| 1    | Temps UTC                     | 092204.999 | Hora UTC de la posició (format 24h)   |
| 2    | Status                        | A          | V: Avis d'atenció del receptor GPS<br>A: Dades vàlides  |
| 3    | Latitud                       | 4250.5589  | Latitud en format <i>ddmm.mmmm</i>  |
| 4    | Indicador N/S                 | S          | Indicador de latitud en l'hemisferi   |
| 5    | Longitud                      | 14718.5084 | Longitud en format <i>ddmm.mmmm</i>   |
| 6    | Indicador E/W                 | E          | Indicador de longitud en l'hemisferi  |
| 7    | Velocitat sobre la superfície | 000.0      | Velocitat en nucs (000.0 a 999.9)   |
| 8    | Direcció sobre la superfície  | 000.0      | Direcció en graus (000.0 a 180.0)   |
| 9    | Data UTC                      | 211200     | Hora UTC de la fixació de posició (format 24h)  |
| 10   | Variació magnètica            |            | En graus (000.0 a 180.0)  |
| 11   | Variació magnètica            |            | Direcció de la variació: E o W.   |
| 12   | Indicador de mode             | A          | N: Dades no vàlides<br>A: Mode Autònom<br>D: Mode Diferencial<br>E: Mode Estimació<br>M: Mode Manual<br>S: Mode Simulació |
| 13   | Checksum                      | 25         |   |

Taula 5: Descripció camps frase GPRMC

●GSA - GPS DOP and Active Satellites; especifica el mode de funcionament del receptor de GPS, el nombre de satèl·lits utilitzats indicats per la frase GCA i/o GNS i els valors de l'error DOP. Conté 7 camps de dades i la següent estructura:

```
$GPGSA,A,x,xx,xx,xx,xx,xx,xx,xx,xx,xx,xx,xx,xx,x.x,x.x,x.x*hh<CR><LF>
```

1 2 3 3 3 3 3 3 3 3 3 3 3 4 5 6 7

**Exemple:**

```
$GPGSA,A,3,01,20,19,13,,,,,,,,,40.4,24.4,32.2*0A<CR><LF>
```

Figura 14: exemple de frase GPGSA

| Camp | Nom                        | Exemple         | Descripció   |
|------|----------------------------|-----------------|--|
| 1    | Mode                       | A               | M: Mode manual; forçat a treballar en 2D o 3D<br>A: Mode automàtic, es permet el canvi automàtic entre 2D i 3D |
| 2    | Mode (Tipus de fixació)    | 3               | 1: Fixació no disponibles<br>2: 2D<br>3: 3D  |
| 3    | Satèl·lit utilitzat (1-12) | 01,20,19,13.... | ID del satèl·lit. Del 01 al 32. 12 màxim   |
| 4    | PDOP                       | 40.4            | Posició de la DOP (00.0 a 99.9)  |
| 5    | HDOP                       | 24.4            | DOP Horitzontal (00.0 a 99.9)  |
| 6    | VDOP                       | 32.2            | DOP Vertical(00.0 a 99.9)  |
| 7    | Checksum                   | 0A              |  |

Taula 6: Descripció camps frase GPGSA

● GLL - Latitude and Longitude with time of position Fix and Status; Latitud i longitud de la posició actual, temps i estatus. Conté 8 camps de dades i la següent estructura:

```
$GPGLL,ddmm.mmmm,a,dddmm.mmmm,a,hhmmss.sss,A,a*hh<CR><LF>
```

1 2 3 4 5 6 7 8

**Exemple:**

```
$GPGLL,4250.5589,S,14718.5084,E,092204.999,A,A*2D<CR><LF>
```

Figura 15: exemple de frase GPGLL

| Camp | Nom               | Exemple    | Descripció  |
|------|-------------------|------------|---|
| 1    | Latitud           | 4250.5589  | Latitud en format <i>ddmm.mmmm</i>  |
| 2    | Indicador N/S     | S          | Indicador latitud en l'hemisferi  |
| 3    | Longitud          | 14718.5084 | Longitud en format <i>ddmm.mmmm</i>   |
| 4    | Indicador E/W     | E          | Indicador de longitud en l'hemisferi  |
| 5    | Temps UTC         | 092204.999 | Hora UTC de la posició (format 24h)   |
| 6    | Estat             | A          | A: Dades vàlides<br>V: Dades no vàlides   |
| 7    | Indicador de mode | A          | N: Dades no vàlides<br>A: Mode Autònom<br>D: Mode Diferencial<br>E: Mode Estimació<br>M: Mode Manual<br>S: Mode Simulació |
| 8    | Checksum          | 2D         |   |

Taula 7: Descripció camps frase GPGLL

### 3- Sensors de moviment: Una altra forma de localitzar/posicionar objectes

#### 3.1- Introducció: Per què utilitzar altres formes de localitzar objectes?

Tot i que en la majoria de casos un sistema de localització per satèl·lit va a ser suficient per a localitzar un objecte, no sempre podrem utilitzar-lo ja que, per exemple, pot donar-se el cas que, degut a la inestabilitat meteorològica de la zona en la que es troba el receptor GPS, la comunicació amb els satèl·lits no es pugui realitzar impedit, d'eixa manera, la recuperació de la posició actual pel que hauríem d'obtenir la posició d'una altra forma. Com? Fàcil: utilitzant sensors capaços de mesurar directa o indirectament moviment, doncs sí coneixem la nostra posició inicial, sols hem de mesurar la distància recorreguda en un espai de temps i, una vegada fet açò, comprovar en quina direcció s'ha produït el desplaçament per a que, després de resoldre les corresponents equacions matemàtiques, obtenir la nostra posició actual.

#### 3.2- Acceleròmetres

##### 3.2.1- Introducció: Què és un acceleròmetre?

Els acceleròmetres, com el nom indica, són dispositius capaços de mesurar acceleració, encara que cal dir que aquests dispositius no sempre mesuren necessàriament el canvi de velocitat d'un objecte en l'espai, és a dir, la coordenada de l'acceleració. De fet, aquests dispositius no solen mesurar aquest canvi de velocitat o acceleració, sinó que, per norma general, mesuren l'acceleració que experimenta l'objecte quan es troba en caiguda lliure, cosa que implica que l'acceleròmetre en eixe cas no proporciona una eixida de 1G sinó de 0. Com la mesura proporcionada per aquests aparells és relativa a la força de la gravetat, quan estan en repòs sobre l'eix Z donen una eixida de 1G pel que aquest *off-set* deu de tenir-se en compte a l'hora d'interpretar la mesura i ser compensat sí és necessari. La diferenciació entre dos tipus d'acceleració i l'aparició d'aquest *off-set*, es deu al principi d'equivalència d'Einstein: els efectes de la gravetat sobre un objecte són indistingibles de l'acceleració.

##### 3.2.2- Tipus d'acceleròmetres

La forma de treballar d'un acceleròmetre electrònic és semblant a la d'un pes enganxat en un moll: quan el sistema experimenta acceleració, el pes es desplaça fins que la força que exerceix el moll el frena. El punt on el pes quedarà frenat un instant és proporcional a l'acceleració experimentada, pel que mesurant el desplaçament podem obtenir l'acceleració. Per a visualitzar aquesta idea, observem el sistema massa-moll representat en la figura 16:

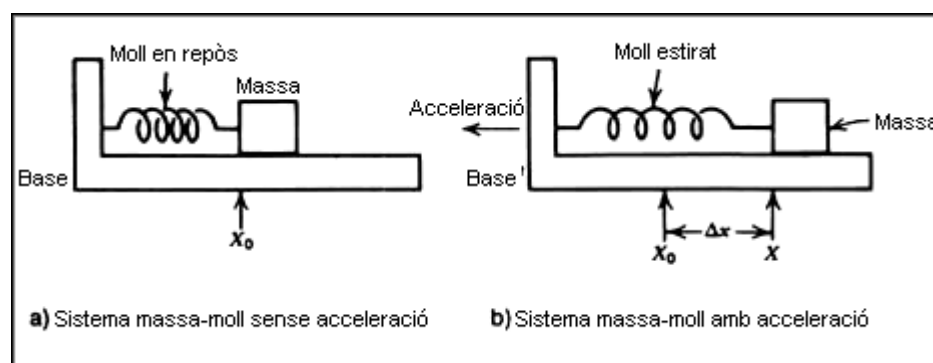


Figura 16: Principi bàsic de funcionament d'un acceleròmetre



Com podem veure en aquesta figura, el pes està unit a la base mitjançant un moll i pot desplaçar-se lliurement per la mateixa. Mentre el sistema estiga en repòs, el moll no exercirà cap força sobre la massa (figura 16a), però quan experimente una acceleració cap a l'esquerra (figura 16b), el moll s'allargarà de forma proporcional a la força tal i com descriuen les lleis de *Newton* i *Hook*:

$$ma = kDx$$

on  $k$  és la constant del moll,  $Dx$  és l'allargament del moll,  $m$  el pes de la *massa sísmica* i  $a$  és l'acceleració. Gràcies a la fórmula anterior, la mesura de l'acceleració es redueix a una mesura de l'extensió, o desplaçament lineal, del moll. Òbviament aquest argument continua complint-se si l'acceleració experimentada fos en sentit contrari, sols que el moll no s'allargaria, sinó que es comprimiria.

Depenent de quin element s'utilitza per a convertir el moviment mecànic en una senyal elèctrica, els acceleròmetres es classifiquen en:

- **Acceleròmetres piezoelèctrics:** aquests sensors es basen en la propietat que tenen certs tipus de cristalls de generar electricitat quan es veuen sotmesos a una pressió. Aquesta propietat és també la base de sensors tan "familiars" com puga ser l'agulla d'un tocadiscs o determinats tipus de micròfons.

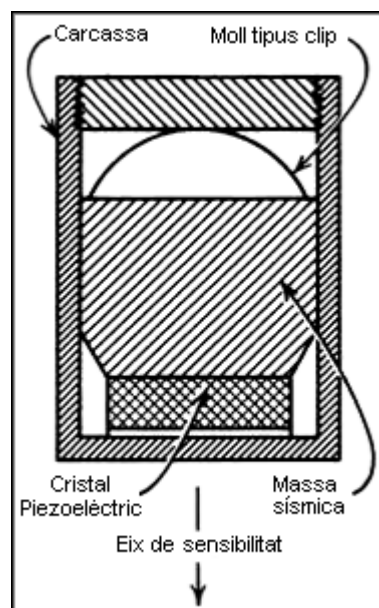


Figura 17: Diagrama intern bàsic d'un acceleròmetre piezoelèctric.

Tal i com podem veure en la figura 17, aquest tipus d'acceleròmetres es poden implementar col·locant la *massa sísmica* sobre el cristall, de manera que, al exercir-se sobre aquesta una pressió, apareixerà un voltatge que serà proporcional a l'acceleració, ja que  $F = ma$ , pel que mesurant aquesta diferència de potencial podrem obtenir el valor de l'acceleració. Com el cristall és una font d'alta impedància, serà necessària la utilització d'un detector de baix soroll i alta impedància d'entrada. També serà necessari amplificar l'eixida ja que els valors típics d'eixida d'aquests sensors solen estar dins del rang de mili-volts. Cal destacar que la freqüència natural d'aquests dispositius sol ser major de 5kHz pel que són ideals per a aplicacions d'alta freqüència com pot ser la mesura de vibracions, colps, impactes, etc.

Dins d'aquest tipus d'acceleròmetres podem trobar tres tipus més: els de tipus *shear*, els de tipus *flexural beam* i els de tipus *compression*, encara que aquest últim estan en desús. Els primers són els més utilitzats ja que disposen d'unes dimensions reduïdes i, al estar els cristalls aïllats, són més immunes a variacions de temperatura o deformacions de la base, raó per la qual el seu rendiment és òptim per a la gran majoria d'aplicacions. En canvi els segons sols són ideals per a aplicacions de baix cost o que requerisquen mesurar acceleracions baixes.



Figura 18: Acceleròmetre de tipus *Shear* amb encapsulat ceràmic

- **Acceleròmetres piezoresistius:** aquest tipus d'acceleròmetres utilitzen com a moll un cristall semiconductor al que se li enganxa la *massa sísmica*. La deformació dels cristalls es mesura instal·lant sobre els mateixos galgues extensomètriques que, al mateix temps, són connectades a un pont de *wheatstone*, de manera que al aplicar-se una acceleració al sistema, les galgues acaben patint una deformació i, per tant, produint una eixida elèctrica proporcional a l'acceleració experimentada, doncs el canvi produït en la seva resistència es veurà també reflectit en el voltatge que caurà sobre elles.

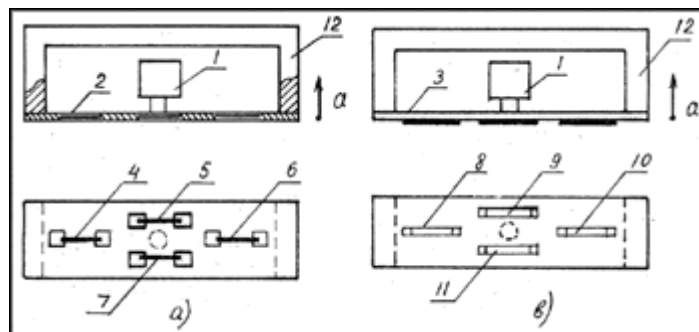


Figura 19: Dos tipus d'acceleròmetres piezoresistius: amb galgues “difuses” (a) i amb galgues “cimentades” (b)

En la figura 19 podem observar l'esquema bàsic i la disposició dels dos principals tipus d'acceleròmetres piezoresistius. L'objecte marcat com (1) és la *massa sísmica*, el (2) és la capa de semiconductor, el (3) és la capa ceràmica i 4,5,6,7 o 8,9,10 i 11 són, respectivament, les posicions de les galgues difuses i cimentades.

Els acceleròmetres piezoresistius es caracteritzen per tenir una gran sensibilitat i una excel·lent resposta senyal-soroll cosa que, sumada a la seva resposta en DC, els fa ideals per a mesurar polses de llarga duració com els que s'experimenten en els impactes a gran velocitat.

● **Acceleròmetre capacitiu/inductiu/resistiu:** es tracta d'acceleròmetres que connecten en l'extrem de la *massa sísmica* que no està enganxada al moll un d'estos tres components passius, de manera que quan la massa sísmica es mou per un canvi en l'acceleració, es produeix un canvi proporcional en les propietats elèctriques del component en qüestió (Capacitat, Inductància o Resistència), cosa que al mateix temps produeix un canvi en la tensió d'eixida del sensor.

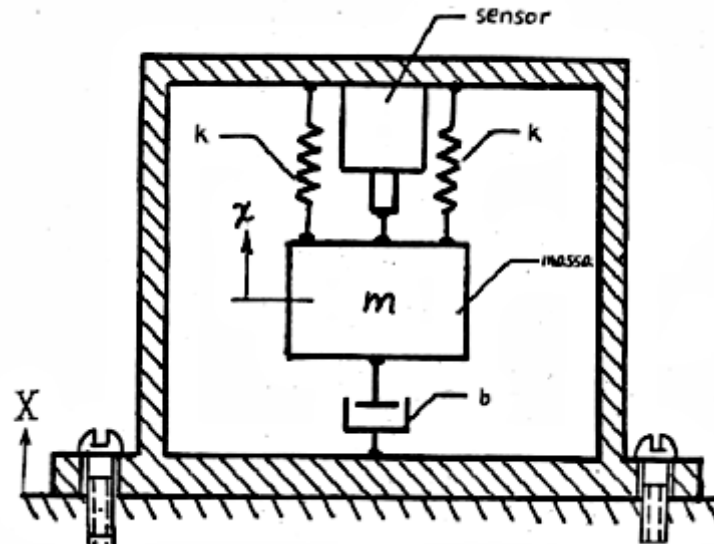


Figura 20: Diagrama intern bàsic d'un acceleròmetre capacitiu

En la figura 20 podem veure un exemple de l'estructura interna bàsica d'un acceleròmetre capacitiu. Com podem observar, la massa sísmica està penjada de dos molls de constant elàstica idèntica  $k$  de manera que, quan es produeix una acceleració, la massa sísmica es mou variant la separació entre les plaques del condensador. Aquest canvi en la separació provocarà un canvi en la capacitat del mateix, i per tant, un canvi en la tensió mesurada en les seves plaques. El funcionament d'un acceleròmetre resistiu o d'un acceleròmetre inductiu és similar a aquest, sols que el paràmetre que canvia al desplaçar-se la massa sísmica no és la capacitat sinó la resistència o la inductància.



Figura 21: Acceleròmetre capacitiu

Els sensors d'aquest tipus es caracteritzen per tenir una resposta en freqüència bona i una eixida estable dins del rang de temperatures compreses entre els  $-20$  i els  $120^{\circ}\text{C}$ . Tot açò sumat a la seva resposta en DC els fa ideals per a mesurar acceleracions petites o de baixa freqüència tals com les experimentades per les suspensions dels cotxes i inclús poden utilitzar-se per a mesurar l'efectivitat dels frens o monitoritzar trajectòries.

### **3.2.3- Aplicacions dels acceleròmetres**

En un primer moment es podria pensar que els acceleròmetres no tenen molta utilitat pràctica, però la realitat és que aquests sensors poden utilitzar-se en una gran quantitat d'aplicacions:

- En **Enginyeria** a part d'utilitzar-se per a estudiar el rendiment de motors i dels frens dels vehicles, també s'utilitzen per a estudiar vibracions d'estructures i inclús terratrèmols.
- En **Biologia** se solen utilitzar per a controlar el gasto energètic d'un animal, sobre tot quan no es pot tenir un accés fàcil com passa amb les espècies aquàtiques.
- En **Aplicacions industrials**, per a controlar l'estat (salut) de motors o màquines que tinguen alguna classe de moviment rotatori com bombes, torres de refrigeració, etc. En aquest entorn també es poden utilitzar per a controlar que les vibracions estiguen dins d'un marge, ja que de no estar-ho, l'aparell supervisat pot avariar-se i tenir una reparació inviable.
- **Aplicacions mèdiques**: l'exemple més conegut és el de les sabates esportives que mesuren la distància recorreguda i calculen les possibles calories consumides durant eixe desplaçament, però n'hi han d'altres de més rellevància doncs aquests sensors ens poden ajudar a detectar moviments en pacients mentre se li realitza alguna prova en la qual és important que estiga immòbil, com ara mateixa quan se li està fent un electrocardiograma, o bé per a detectar la profunditat de les compressions del massatge respiratori durant un procés de des-fibril·lació.
- **Navegació**: els acceleròmetres són útils en els sistemes de navegació inercial ja que d'aquesta manera no necessiten una referència externa per a conèixer els canvis de velocitat, a part que poden disposar d'aquesta informació instantàniament. A més, com la gravetat varia amb l'altura, també poden ser utilitzats per a mesurar canvis d'altura.
- **Transport**: a part d'utilitzar-se per a estudiar vibracions dins del cotxe i millorar el confort del conductor i/o passatgers, s'utilitzen principalment pel control d'estabilitat electrònic d'un vehicle i per a determinar com de "fort" ha sigut un impacte i, a partir del resultat obtingut, que el sistema de control del cotxe active el *airbag* o no.
- **Dispositiu d'entrada**: un acceleròmetre pot ser utilitzat també com un mitjà d'entrada de dades tipus teclat o ratolí d'ordinador. El millor exemple d'açò són els mòbils actuals de gamma mitja els quals, per a canviar l'orientació de la pantalla, fan ús d'un acceleròmetre. A part d'aquest ús, hi han mòbils que disposen d'aplicacions que utilitzen aquest sensor com a comptador de passes.
- **Estabilització d'imatge**: principalment s'utilitzen per a evitar imatges borroses o desenfocades ja que al detectar moviment no es captura el contingut dels sensors CCD de la càmera. Una altra aplicació que se'ls sol donar en aquest camp és la de facilitar l'orientació de la imatge: paisatge o retrat.
- **Gravímetre**: com el nom indica, un acceleròmetre també pot ser utilitzat per a mesurar camps de gravetat locals, encara que per a poder ser utilitzat com a tal, se'l deu ser aïllar d'alguna manera ja que un acceleròmetre convencional és sensible també al soroll. A part d'açò i encara que el principi de funcionament és el mateix, el seu disseny està fet de manera que siguin més sensibles a canvis en la mateixa per a, d'esta manera, detectar petits canvis en la gravetat.

- **Control d'integritat de dispositius**: els acceleròmetres són ideals per a detectar colps ja que un colp no és una altra cosa que un canvi brusca en la velocitat (acceleració) d'un objecte, pel que si detectem un canvi brusca en la mateixa, podem actuar instantàniament de manera que els desperfectes provocats al dispositiu a protegir siguin mínims. Com a exemple d'açò tenim els discs durs d'ordinador portàtil, que quan detecten un canvi brusca en l'acceleració, actuen "aparcant" els capçals de lectura/escriptura del disc dur per tal d'evitar pèrdua de dades.

### 3.3- Giroscopis

#### 3.3.1- Introducció: Què és un Giroscopi?

Un giroscopi és un dispositiu el qual basa el seu funcionament en el principi de conservació del moment angular, cosa que el fa útil per a mesurar o mantenir l'orientació. Des de el punt de vista mecànic, el giroscopi és un cos amb simetria de rotació el qual gira al voltant del seu propi eix de simetria. Quan el giroscopi és sotmès a un moment de força que tendeix a canviar l'orientació del eix de rotació, l'efecte que experimenta és diferent al que experimentaria un cos que no gira, pues en el cas del giroscopi, l'efecte produït és un canvi d'orientació en una direcció perpendicular a la direcció intuïtiva de gir mentre que, si es tractés d'un objecte que no està rotant, l'efecte produït seria un canvi en la seva direcció.

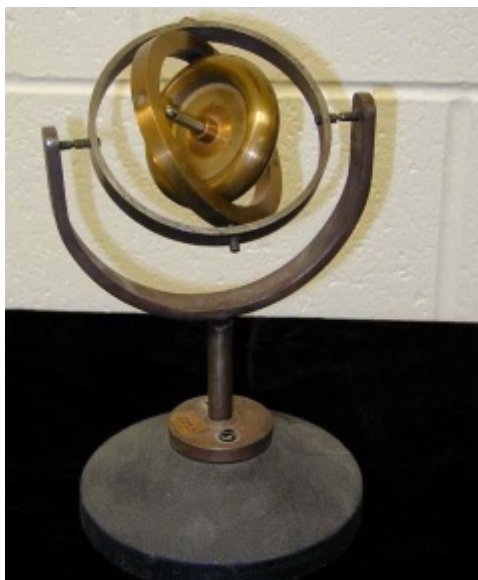


Figura 22: Giroscopi mecànic de 3 eixos

Aquest dispositiu va ser inventat en l'any 1852 per la persona que va demostrar la rotació de la terra utilitzant un simple pendol: *Léon Foucault*. De fet, *Foucault* va crear aquest aparell per a simplificar el procés al mateix temps que intentava trobar la raó per la qual la velocitat de rotació del pendol era un factor  $\sin(\lambda)$  més lenta que la velocitat de rotació de la terra. Així doncs, amb l'ajuda del seu col·laborador *Froment*, va presentar un aparell capaç de conservar una rotació suficientment ràpida, 150-200 voltes per minut, durant un temps suficient, uns 10 minuts, per tal que la rotació poguera ser mesurada. Més tard *Foucault* se'n va adonar que, si bloquejava certs moviments del suport de l'aparell, aquest es posava en línia amb el Nord, descobrint d'aquesta manera el girocompàs.

Per tal d'explicar el funcionament d'un giroscopi mecànic suposarem que aquest és idèntic al de la figura 23, és a dir, que es tracta d'un disc muntat sobre un eix horitzontal al voltant del qual gira el disc lliurement i a gran velocitat. També suposarem que un observador sosté l'eix del fons amb la

mà esquerra mentre que amb la mà dreta subjecta l'altre extrem. Doncs bé, si aquest observador tracta de fer girar l'eix cap a la dreta, o siga baixa la mà dreta i puja la mà esquerra, observarà com, al contrari que es podria pensar en un primer moment, el giroscopi empeny la seva mà dreta i tira de la seva mà esquerra. Aquest efecte, conegut també com *efecte giroscopi*, es deu a que el disc gira lliurement sobre l'eix, ja que, en l'instant en el que desequilibrem l'eix cap a un costat, és com si estiguérem aplicant una successió d'impulsos al disc. Aquests impulsos afegiran una petita velocitat perpendicular a les velocitats individuals de les masses que formen el disc produint al cap d'uns instants un canvi en la direcció, però no en el mòdul, de la velocitat. Com a exemple d'aquest efecte o d'objectes que es basen en el mateix tenim dos joguines clàssiques: la baldufa i el jo-jo.

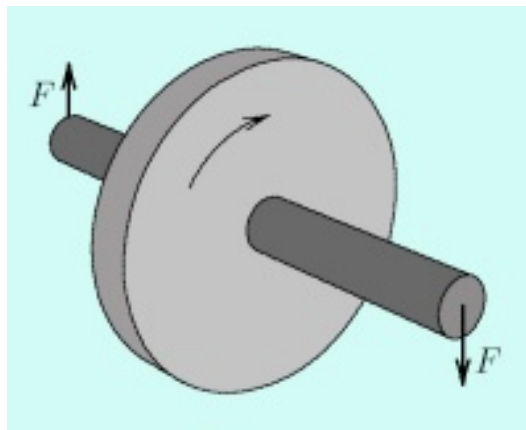


Figura 23: Quan s'intenta baixar el costat dret aquest no baixa, sinó que es mou cap a l'observador.

Cal dir que, en un giroscopi mecànic, a part del moviment rotatori al voltant de l'eix de simetria, apareixen dos moviments principals: El de precessió i el de nutació. El primer és el que fa útil el giroscopi com a element de referència de direcció, pues quan s'aplica un moment perpendicular a l'eix de rotació, es produeix una rotació sobre un eix perpendicular al moment i a la velocitat angular.

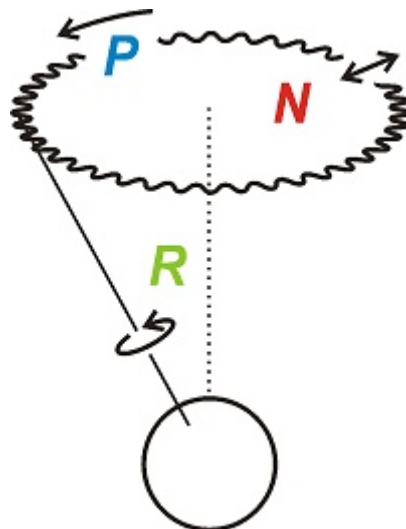


Figura 24: Moviments d'un giroscopi: en verd el de rotació, en roig el de nutació i en blau el de precessió

El segon moviment apareix quan el moment que causa el moviment de precessió canvia de valor, o siga, quan hi ha un canvi de velocitat de precessió. Com aquest canvi no és instantani hi ha un període de transició en el qual el giroscopi "cedeix" un poc al moment en la mateixa direcció com passaria en un objecte que no gira. Passat eixe temps transitori, el giroscopi recupera el que ha perdut i oscil·la en la direcció del moment al voltant de la trajectòria de precessió final tal i com podem veure en la figura 24.

### **3.3.2- Giroscopis no mecànics: Giroscopis d'estructura ressonant i Giroscopis de Fibra Òptica (FOG).**

Actualment el concepte de giroscopi no s'implementa exclusivament partint del model mecànic creat per *Foucault*, sinó que s'utilitzen altres mètodes per a materialitzar físicament la idea representada per aquest aparell sent les més importants el giroscopis làser, els giroscopis de fibra òptica (FOG) i els giroscopis basats en estructures ressonadores.

Un giroscopi de fibra òptica o FOG, es basa en la contrapart electromagnètica de la mecànica de rotació: l'*efecte Sagnac*. Segons l'experiment realitzat en 1913 per aquest científic francès, dos rajos de llum, inicialment en fase, els quals viatgen en sentits oposats per una circumferència de, per exemple, fibra òptica arribaran a un punt determinat de la circumferència en diferent moment, és a dir, desfasats, sí el circuit pel qual viatja la llum ha experimentat una rotació mentre els dos rajos encara estaven en moviment, pues al produir-se aquest desplaçament, el punt d'entrada/eixida canvia, fent d'aquesta manera que el raig que es desplaça en el mateix sentit en el que s'ha rotat el circuit arribi abans al punt final ja que la distància que recorre és menor. Com la distància recorreguda és proporcional a la velocitat angular de la plataforma rotatòria, sols caldrà mesurar el desfasament dels rajos per a conèixer a quina velocitat està girant el aparell. I això és el que fa un *FOG*: calcula la velocitat angular en cada instant i una vegada fet açò, calcula el gir multiplicant la velocitat angular per temps.



Figura 25: Giroscopi de Fibra Òptica (FOG)

Degut a que aquest tipus de giroscopis no contenen elements sensibles a la vibració ni components mòbils i van equipats amb rellotges d'alta precisió per a poder mesurar la llum, són extremadament fiables i sensibles ja que poden arribar a detectar sense problemes velocitats de gir de tan sols 0,00001 graus/segon, raó per la que poc a poc han anat guanyat importància front als giroscopis làser.

En canvi, un giroscopi d'estructura ressonant es basa en el principi físic pel qual, un objecte que vibra, tendix a continuar vibrant en el mateix pla en el que gira el seu punt de recolzament. En la literatura d'enginyeria a aquest tipus de giroscopi se li dona el nom de giroscopi vibratori de *Coriolis*, doncs a mesura que va rotant el pla d'oscil·lació, la resposta detectada pel transductor equival al terme de la força *Coriolis* de l'equació de moviment. Esta força, en determinats tipus d'aplicació, produeix un patró de vibració de precessió sobre l'eix de rotació. Al mateix temps, aquest patró indueix, sobre una ona en repòs, un lent moviment de precessió de velocitat angular diferent a la de la vibració original produint d'aquesta manera el que es coneix com l'efecte "inèrcia d'ona", el qual no sols és la base de funcionament de, per exemple, els giroscopis piezoelèctrics o

els giroscopis multiaxials electrònics, sinó que a més, és el que permet que els giroscopis basats en aquest efecte siguin més barats, simples i menuts que un giroscopi mecànic de similar precisió.

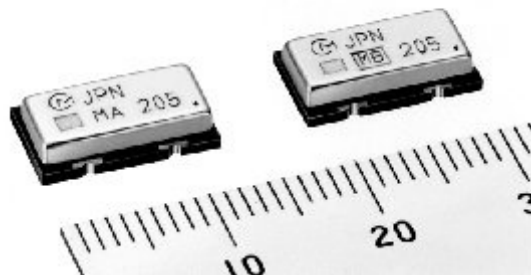


Figura 26: Un parell de giroscopis piezoelèctrics

### 3.3.3- Aplicacions

Fins l'aparició dels giroscopis no mecànics, els giroscopis s'havien estat utilitzant quasi exclusivament com a brúixoles, però hui en dia, gràcies a que la tecnologia actual no sols permet reduir considerablement la grandària d'un dispositiu electrònic, sinó a que més, també permet treure partit al efecte *Sagnac*, els podem trobar en una gran varietat d'aplicacions: des de aplicacions tan bàsiques com pot ser el saber en quin sentit ha canviat l'orientació de la pantalla d'un telèfon mòbil fins a aplicacions tan complexes com són els sistemes de navegació inercial o el guiat d'aeronaus.

Cal dir que, sí actualment els giroscopis tenen tantes aplicacions, és per les reduïdes dimensions dels giroscopis d'estructura ressonant pues, gràcies a aquesta característica, poden ser integrats fàcilment en dispositius electrònics tan quotidians i comuns hui en dia com pot ser un telèfon mòbil o el *joystick* d'una vídeo consola com puga ser la *Wii*. Però com dèiem abans, no sols aquestes dos indústries s'han beneficiat de la major precisió que ofereixen aquests sensors a l'hora de detectar moviment en l'espai, altres com el de l'aviació també ho han fet ja que, entre altres funcionalitats, aquests giroscopis permeten posicionar una nau, controlar el rotor de cola d'un helicòpter o estabilitzar una plataforma en més d'un eix, doncs aquest tipus de giroscopis són capaços d'induir i controlar una oscil·lació creada per ells mateixa.



Figura 27: Giroscopi com a sistema d'estabilització en un aeronau.

Una altra indústria que treu profit als avantatges proporcionats pels giroscopis és la del automòbil. En aquesta indústria, els giroscopis s'utilitzen en conjunció amb un sensor de gir col·locat en el volant per a implementar els sistemes electrònics de control d'estabilitat. Açò es deu a que els giroscopis també poden mesurar la velocitat de gir, és a dir, com de ràpid gira un objecte sobre un eix en concret, pues la seva tensió d'eixida és proporcional a la velocitat de gir, pel que sí instal·lem un giroscopi que mesure l'orientació en l'eix Z (*Yaw*), quan el *micro* del cotxe detecte que s'està girant a més velocitat de la normal, actuarà sobre els frens del cotxe per tal d'intentar estabilitzar-lo. Sí la grandària i el preu del sensor no és una limitació important en l'aplicació i necessitem més



precisió que la que ens ofereix un giroscopi vibratori, aleshores deuríem de considerar l'ús d'un giroscopi de fibra òptica ja que les seues prestacions són superiors en tots els aspectes a les que ens pot oferir un giroscopi convencional.



Figura 28: Sensor d'inèrcia LN-200 de 6 eixos, 3 dels quals són giroscopis de tipus FOG.

Gràcies a l'ús de giroscopis d'aquests tipus podem controlar amb total precisió sistemes on el control d'estabilitat i orientació és crític, raó per la qual és habitual trobar en avions o transbordadors espacials sistemes de navegació inercial que utilitzen aquest tipus de giroscopis per a calcular i obtenir l'orientació del vehicle en cada instant.

### **3.4- Inclínòmetres: Tipus i aplicacions**

Els inclinòmetres, als que normalment se'ls anomena mesuradors de nivell, són instruments que s'utilitzen principalment per a mesurar angles de pendents, elevació o depressió d'un objecte respecte a la gravetat. Tal i com podem observar en la figura 29, aquests sensors poden arribar a ser molt simples ja que amb un fil, un clip i un mesurador d'angles podem construir-ne un fàcilment. Destacar que l'angle mesurat per aquest dispositius és referent a un horitzó artificial creat per ells mateixa.

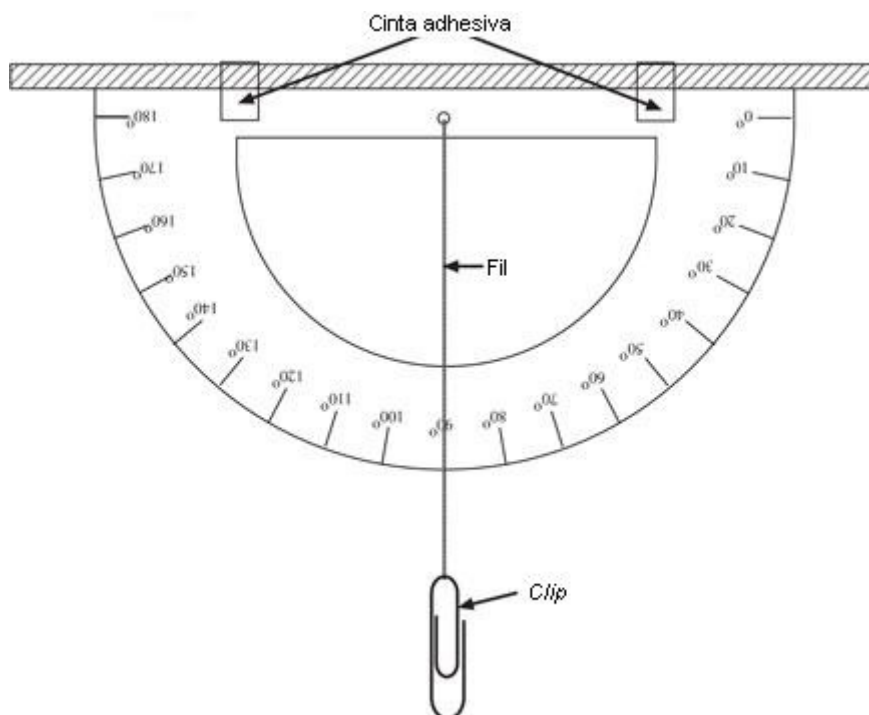


Figura 29: Exemple de com de simple pot arribar a ser un inclinòmetre.

Òbviament la precisió i complexitat d'aquests sensors depèn de la tecnologia utilitzada en la seva fabricació, pel que ens podem trobar amb els següents tipus d'inclinòmetres:

- Inclinòmetre Electrolític: presenten una mesura molt precisa i són capaços de crear una eixida elèctrica que mostra exactament la inclinació a la qual està sotmesa una estructura respecte la gravetat. Destaquen per la seva extraordinària robustesa.



Figura 30: Inclinòmetre de tipus electrolític

- Inclinòmetre de mercuri o de gas: en aquest tipus d'inclinòmetre l'angle el marca, depenent del tipus, una boleta de mercuri o de gas que es troba dins d'un recipient metàl·lic o de cristall lleugerament doblat. Quan es produïska una inclinació, la bola es mourà indicant d'aquesta manera l'angle.



Figura 31: Exemple d'inclinòmetre de mercuri

- Inclinòmetre de tipus pèndul: sol utilitzar-se per a determinar l'estabilitat d'un vehicle marí. Consisteix en un cilindre metàl·lic ple de mercuri sobre el qual se suspèn un pèndul. Quan es produeix moviment (inclinació) el pèndul es mou i mostra l'angle d'inclinació.



Figura 32: Inclinòmetre de tipus pèndul.

- Inclinòmetre digital: La seva principal característica és la seva precisió. Com el nom indica, són totalment digitals i, com és d'esperar, es beneficien de tots els avantatges de la tecnologia doncs actualment permeten que un sistema micro-processat tinga la informació d'angle en un espai realment reduït. Destacar que alguns d'aquests inclinòmetres són en realitat acceleròmetres, ja que quan es produeix un moviment d'inclinació, també es produeix un canvi proporcional en les coordenades de l'acceleració.



Figura 33: Inclinòmetre digital de dos eixos.

Respecte a les aplicacions, tot i que es podria arribar a pensar en un moment que aquests dispositius sols són útils per a mesurar angles, en realitat la seva utilitat va un poc més lluny ja que gràcies a ells i a la seua precisió es pot mesurar la latitud, l'altitud, les desviacions o deformacions d'un objecte respecte a un eix, detectar moviments abruptes o inclús utilitzar-se com a sistema d'entrada de dades com és el cas del *wii-mote*.

### 3.5- Magnetòmetres

#### 3.5.1- Introducció: Què és un Magnetòmetre?

Un magnetòmetre és un aparell de mesura que serveix per a determinar la força i/o direcció d'un camp magnètic. El mètode més conegut per a mesurar aquest fenomen físic és la brúixola però actualment hi han més mètodes i de més precisió que aquest com, per exemple, els magnetòmetres de nucli saturat<sup>7</sup>, els quals solen utilitzar-se en les naus espacials per a mesurar les tres components del camp magnètic, o els magnetòmetres de precessió protònica els quals, al contrari que els anteriors, s'utilitzen per a mesurar les variacions de la magnitud del camp magnètic de, per exemple, la terra.

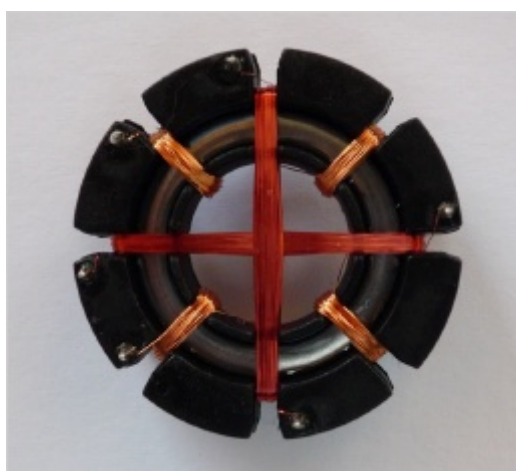


Figura 34: Magnetòmetre de saturació de dos eixos.

<sup>7</sup> Aquests magnetòmetres també reben el nom de *Fluxgate Magnetometer*.

Tal i com veurem en les seccions posteriors, aquests dispositius tenen gran quantitat d'aplicacions ja que el seu ús no es limita exclusivament a la navegació o a la localització d'objectes, sinó que també permeten als científics estudiar planetes i als enginyers estudiar i analitzar la propagació d'ones de ràdio. Això sí, tot i que es podria pensar que els magnetòmetres també poden ser utilitzats com a detectors de metall, la realitat és que, al no mesurar la conductivitat del material com fan aquests últims, sols poden ser utilitzats per a detectar materials fèrrics. L'únic avantatge que tenen sobre els detectors de metalls és la distància a la que són capaços de detectar els objectes ja que, mentre que en un detector de metall esta no sol passar dels dos metres, en un magnetòmetre s'arriba fàcilment a les desenes de metres.

### 3.5.2- Tipus de magnetòmetres

Els magnetòmetres solen classificar-se segons la informació que ens proporcionen del camp magnètic. Així doncs, sí la mesura és el valor total del camp magnètic, estarem davant d'un magnetòmetre escalar, mentre que sí aquest és capaç de mesurar el valor del mateix en una direcció en concret i, per tant, ens proporciona el valor d'una component, estarem front a un magnetòmetre del tipus vectorial. Tal i com podem veure en la figura 35, dins d'aquesta classificació també podem subdividir-los d'acord amb el camp magnètic màxim que poden mesurar: sí el camp magnètic és inferior a 1mT estarem parlant d'un magnetòmetre, però sí és major, aleshores estarem parlant d'un *Gaussimetre*.

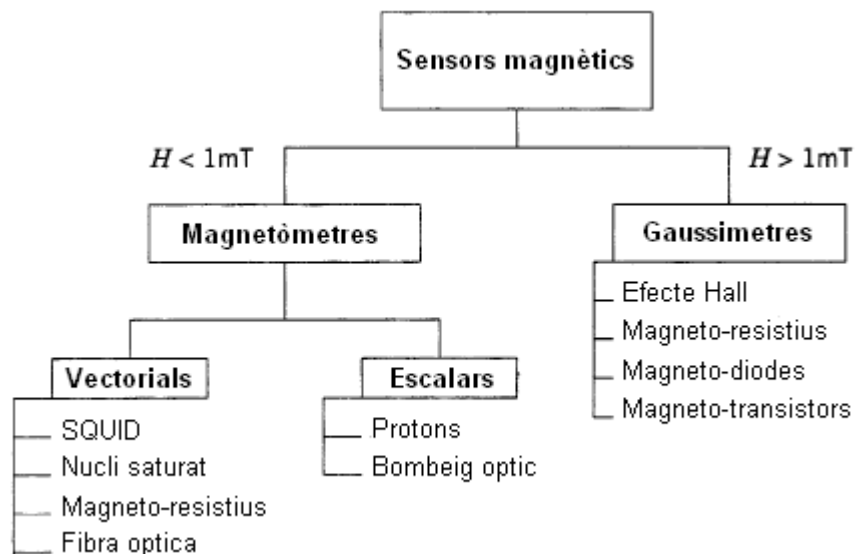


Figura 35: Classificació dels sensors magnètics.

El principals inconvenients i avantatges dels magnetòmetres escalars són conseqüència d'aprofitar les propietats atòmiques i nuclears de la matèria per a mesurar el camp magnètic, doncs per un costat els fa relativament insensibles a l'orientació i els permet tenir molta resolució, però per l'altre els limita el rang de mesura a 20uT-100uT i obliga a que el camp magnètic a mesurar siga totalment uniforme al llarg del element de sensat.

En la taula 8 podem trobar un petit llistat dels principals magnetòmetres existents actualment junt a les especificacions més importants dels mateixos. Com podem observar, les resolucions venen especificades en nano Tesles. Això es deu a que el Tesla és una unitat molt gran ja que, per exemple, el valor del camp magnètic en l'equador 0° és d'uns 31 *micro Tesles* mentre que un imant de nevera pot arribar a tenir 10 *mili Tesles*. Aquesta diferència tan gran fa que en enginyeria no s'utilitze el *Tesla* com a unitat magnètica, sinó que siga el *Gauss*, el qual equival a 100,000 *nano Tesles*.

| Instrument                | Rang (mT)            | Resolució (nT) | Ample de Banda (Hz) | Comentari                                      |
|---------------------------|----------------------|----------------|---------------------|--|
| Nucli saturat             | $10^{-10}$ a $10^6$  | Variable       | $10^{-1}$ a $10^6$  | Magnetòmetre vectorial d'ús genèric            |
| SQUID                     | $10^{-9}$ a 0.1      | 0.1            | DC a 5              | És el magnetòmetre més sensible                |
| Sensor efecte <i>Hall</i> | 0.1 a $3 \cdot 10^4$ | 100            | DC a $10^8$         | Ideal per a camps superiors a 1T.              |
| Magneto-resistiu          | $10^{-3}$ a 5        | 10             | DC a $10^7$         | Ideal per a aplicacions d'exigència intermèdia |
| Protons                   | 0.02 a 0.1           | 0.05           | DC a 2              | Magnetòmetre escalar d'ús genèric              |
| Bombament òptic           | 0.01 a 0.1           | 0.005          | DC a 5              | És el magnetòmetre escalar de major resolució  |

Taula 8: Característiques dels principals sensors de camp magnètic.

Un altra forma de classificar-los seria segons l'ús que se'ls dona, és a dir, sí s'utilitzen per a mesurar camps magnètics de variació ràpida seran del tipus AC, però en canvi, sí s'utilitzen per a mesurar variacions lentes com puga ser un procés de mineralització, aleshores el magnetòmetre serà de tipus DC. Nosaltres, en canvi, els classificarem segons el procediment utilitzat per a mesurar el camp magnètic. Segons aquesta classificació, els tipus de magnetòmetres més importants són:

- **Magneto-díode:** Com podem veure en la figura 36, un magneto-díode és en essència un díode convencional al que se li ha afegit, entre les regions P i N, una regió semi-conductora intrínseca<sup>8</sup>. La presència d'aquesta regió intermèdia no és casual ja que aquest tipus de sensors magnètics basen el seu funcionament en el que es coneix com efecte de magneto-concentració: la presència d'un camp magnètic B ortogonal al camp elèctric causa un agrupament dels portadors de la carrega en la part superior o inferior de la placa segons el sentit del camp magnètic.

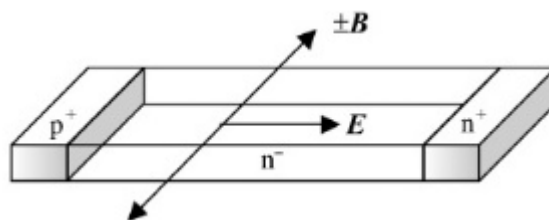


Figura 36: Estructura d'un magneto-díode genèric

Com una cara de la placa ha sigut polida i l'altra desbastada, el ritme de recombinació dels portadors es veurà modificat i, en presència d'un camp magnètic, modificarà les característiques I-V del díode. Desgraciadament aquesta relació és sensible a les variacions de temperatura i, a més, és no lineal, pel que sí també tenim en compte que és molt complicat reproduir còpies idèntiques d'aquests sensors, no sorprèn que aquests tinguin un escàs interès comercial.

<sup>8</sup> Aquesta estructura també rep el nom de díode PIN.

- **Magneto-transistor:** els magneto-transistors són transistors comuns que han sigut dissenyats de forma que la corrent de col·lector siga sensible a la presència d'un camp magnètic. Açò es pot aconseguir desviant la corrent dels portadors mitjançant la força de *Lorentz* o bé mitjançant la magneto-concentració, però independentment del mètode utilitzat, aquesta classe de sensors no sols presenta els mateixos inconvenients que un magneto-díode, sinó que a més presenten problemes de soroll i d'*off-set*, pel que, sumat al fet que desviar electrons mitjançant la força de *Lorentz* ve a ser com afegir un sensor d'efecte *Hall* en l'interior del magneto-transistor, fa que siga més que comprensible que aquest tipus de sensor tinga encara menys interès comercial que un magneto-díode.
- **Sensor magnètic d'Efecte Hall:** els sensors d'aquest tipus detecten el camp magnètic i la seva polaritat gràcies a una de les conseqüències de la força de *Lorentz*: l'efecte *Hall*. Bàsicament, aquests sensors consisteixen en una fina peça rectangular de material semiconductor de tipus P per la que circula una corrent elèctrica continua. Com podem observar en la figura 37, quan el dispositiu se situa dins d'un camp magnètic perpendicular a la corrent elèctrica, les línies del flux magnètic exerciran una força en el semiconductor que desplaçarà als electrons i els buits, pel que, mentre el semiconductor estiga sotmès a aquest camp, el moviment de portadors de càrrega continuarà i crearà, gràcies a l'efecte *Hall*, una diferència de potencial entre els dos extrems del semiconductor proporcional a la intensitat del camp magnètic, pel que mesurant aquesta tensió, podrem calcular la intensitat del camp magnètic.

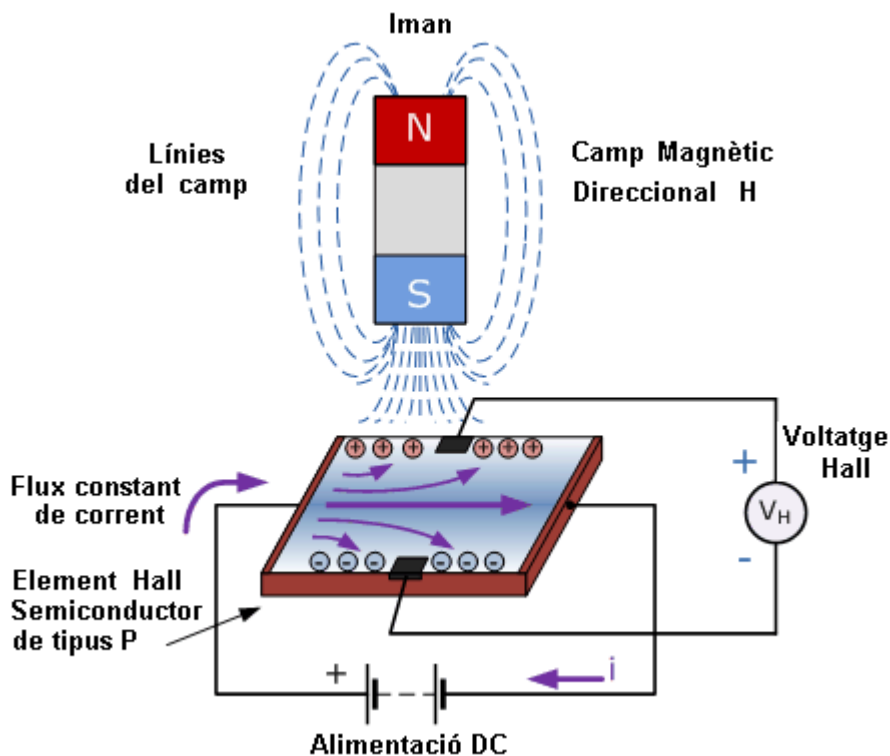


Figura 37: Esquema de funcionament d'un sensor Hall

Les propietats físiques i magnètiques d'aquest tipus de sensors els fan ideals per a mesurar camps magnètics molt intensos, pel que, per norma general, s'utilitzen per a implementar sensors de moviment.

●**Magnetòmetre resistiu o AMR (*Anisotropic MagnetoResistance Sensor*):** aquests magnetòmetres basen el seu funcionament en dos fets: la resistència elèctrica d'un material magnètic es veu afectada per la presència d'un camp magnètic i les seues propietats magnètiques depenen de la direcció del mateix. Això vol dir que, jugant amb la forma i les condicions de fabricació d'un objecte, podrem aconseguir que aquest presente unes característiques magnètiques concretes en un eix. Un exemple d'aquest fet el podem veure en la figura 39:

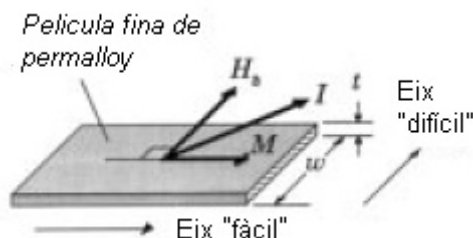


Figura 38: Esquema de fabricació d'un sensor AMR.

Durant la fabricació de l'element AMR de la figura, s'aplica un camp magnètic al llarg de la pel·lícula de *permalloy*<sup>9</sup> per tal de magnetitzar-lo i, d'aquesta manera, definir el seu eix "fàcil", és a dir, l'eix més susceptible als efectes del magnetisme. Si fem circular una corrent  $I$  de manera que forme un angle de  $45^\circ$  respecte a aquest eix, al aplicar un camp  $H_a$  perpendicularment amb el vector  $M$ , aquest rotarà i es produirà un canvi en la magnetoresistència del material.

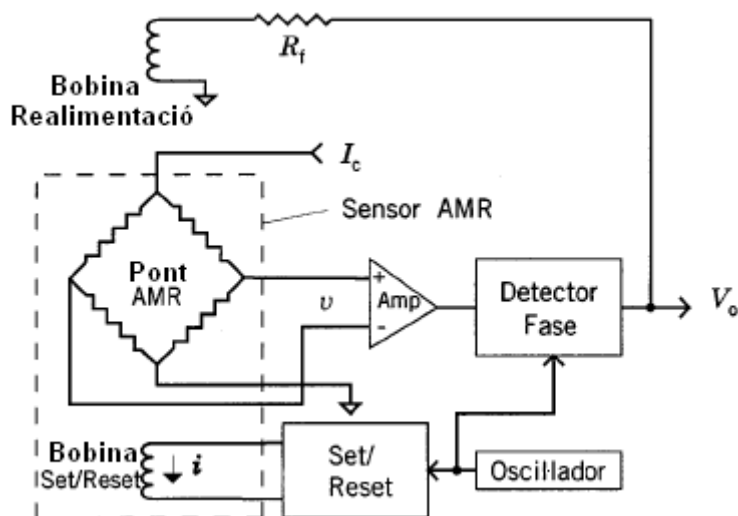


Figura 39: Diagrama bàsic d'un magnetòmetre AMR

Normalment, per a millorar la resposta d'eixida d'aquests sensors, se'ls sol disposar en un pont *Wheatstone* com el mostrat en la figura 39 ja que d'aquesta manera, segons el valor de les resistències, els podrem utilitzar-los com a magnetòmetres o com a *Gaussimetres*.

●**Magnetòmetre de nucli saturat o Magnetòmetre *Fluxgate*:** Generalment, aquest tipus de magnetòmetres està format per dos bobines, una d'excitació i una altra de lectura, les quals estan acoblades magnèticament per un nucli d'un material fàcilment magnetitzable. Aquesta disposició és la base del seu funcionament ja que, com el comportament del nucli front la magnetització no és lineal, si apliquem en el bobinat d'excitació una corrent que siga capaç de saturar el nucli en els dos sentits de manera simètrica i periòdica, quan s'aplique el camp magnètic a mesurar al bobinat de lectura, el camp extern es sumarà al camp magnètic existent i modificarà la saturació del nucli de manera que esta ja no serà igual en els dos sentits, pel

<sup>9</sup> El *Permalloy* és una aliatge magnètic composta generalment de níquel i ferro amb un alt efecte magneto-resistiu.

que sí mesurem aquesta diferència, obtindrem el valor de la component del camp magnètic que ha travessat la bobina de lectura.

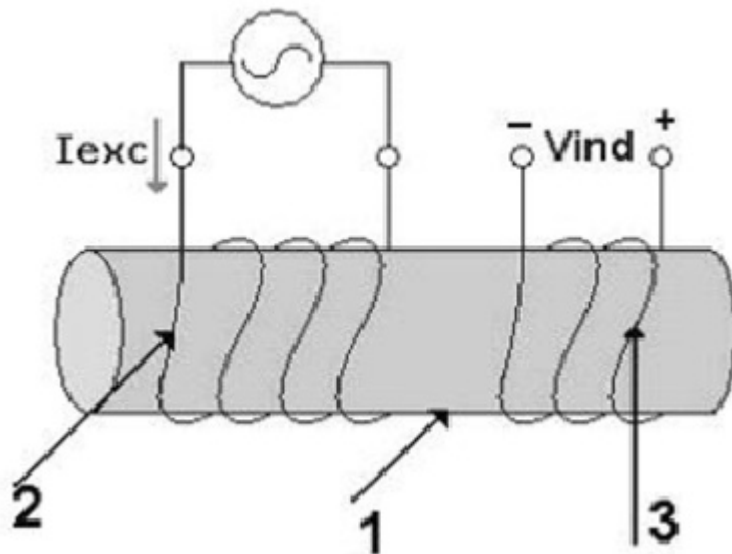


Figura 40: Esquema d'un magnetòmetre de nucli saturat: nucli (1), bobina d'excitació (2) i bobina de lectura (3)

Aquests magnetòmetres són menuts, robustos, fiables i consumeixen poc. Estes característiques, junt al fet que permeten mesurar les components vectorials d'un camp magnètic d'un valor comprés entre 1nT i 1mT amb una freqüència de fins varis mils d'hertz, els ha convertit en dispositius molt versàtils raó per la qual se'ls pot trobar en qualsevol tipus d'aplicació: des de la localització de mines fins la mesura i control de posició d'una nau espacial.

• **Magnetòmetre SQUID o magnetòmetre basat en un Dispositiu Superconductor d'Interferència Quàntica:** els magnetòmetres d'aquest tipus basen el seu funcionament en els efectes que produeix la presència d'un camp magnètic sobre la magnitud de la corrent crítica que flueix entre dos superconductors que estan separats per una fina capa d'aïllant.

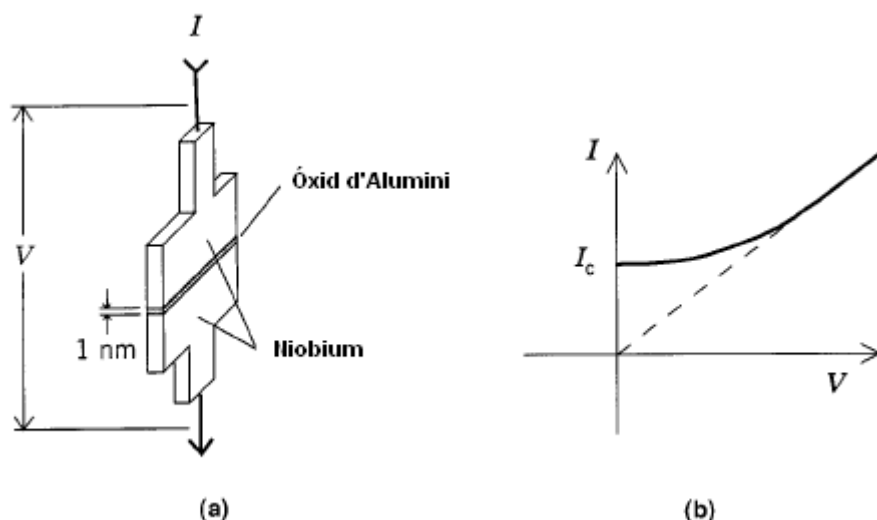


Figura 41 esquema d'un element SQUID (a) i la seva relació I-V (b)

Com podem veure en la figura, dos superconductors idèntics se separen per una capa d'aïllant molt fina. Quan la temperatura de la unió es reduïska a menys de  $-269^{\circ}\text{C}$ , la resistència elèctrica dels superconductors es veurà reduïda considerablement i una corrent de superconductor, anomenada corrent crítica  $I_c$ , fluirà en la unió i la seva magnitud serà una



funció periòdica del flux magnètic present en la mateixa unió. El valor màxim d' $I_c$  es donarà quan el valor del flux siga  $n\Phi_0$ , amb  $\Phi_0 = 2fW$ , mentre que el seu valor mínim tindrà lloc per a valors de flux igual a  $(n + 1/2)\Phi_0$ . Aquest fenomen rep el nom d'efecte *Josephson DC* i és un dels molts efectes que es dona quan, entre dos superconductors, es situa una fina capa de material aïllant i se'ls sotmet a una temperatura propera al zero absolut.

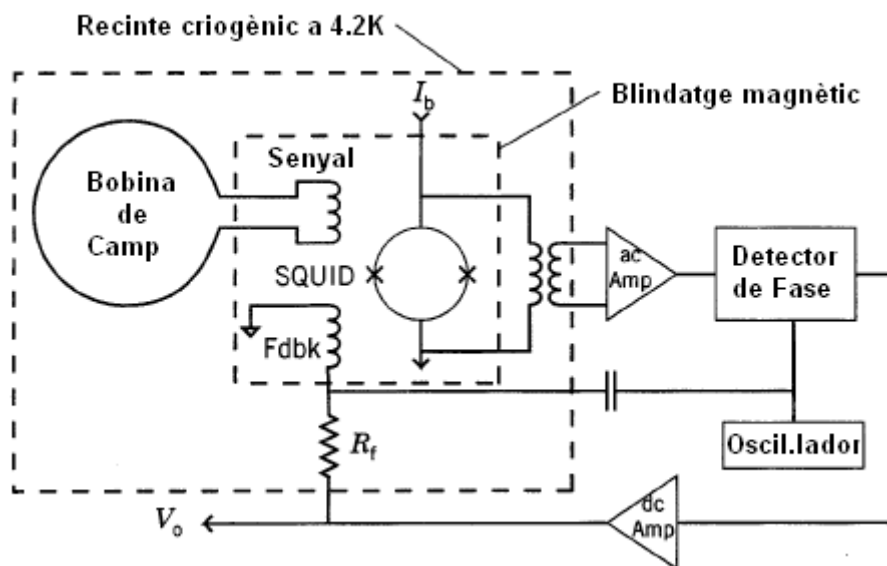


Figura 42: Magnetòmetre SQUID d'ampli rang dinàmic DC

Com es pot intuir a partir de la figura 42, per a que els materials superconductors estiguen a una temperatura propera al zero absolut el que es fa és submergir-los en heli líquid a l'interior d'un recipient criogènic. En aquestes condicions de temperatura, el llaç que detecta el camp i el bobinat de senyal formaran una bobina d'inducció DC i, quan s'aplique un flux extern a la bobina, es generarà una corrent en el llaç que mantindrà constant el flux net en la bobina, inclús per a camps magnètics DC.

Per a optimitzar la sensibilitat del SQUID, se'l polaritza magnèticament amb un altre camp al qual, posteriorment, se li superposa un petit camp magnètic AC de freqüència compresa entre 100kHz i 500kHz. Com l'eixida del SQUID és una senyal portadora d'amplitud modulada suprimida, un canvi en l'amplitud indicarà un canvi en el camp magnètic respecte al punt de polarització inicial, mentre que el canvi en la fase indicarà la polaritat del canvi. Una vegada la senyal d'eixida s'ha amplificat, es desmodula sincrònicament a banda base i, la senyal DC resultant, s'amplifica i és realimentada a través de la resistència  $R_f$  a la bobina acoblada al SQUID. La corrent que passa per la bobina de realimentació genera un camp magnètic en el SQUID que s'oposa al camp aplicat. Açò fa que el punt de treball del SQUID es mantinga molt prop del seu punt de polarització i que el factor escala del magnetòmetre depenga tant de  $R_f$  com de la constant de realimentació del bobinat de la bobina.

Tant el llaç de lectura, com el bobinat de senyal, el SQUID, la bobina i la resistència de realimentació  $R_f$  es troben dins de la cambra criogènica i, a excepció del llaç de lectura, tots estan blindats magnèticament. La resta del circuit funciona a temperatura ambient.

Actualment, els magnetòmetres basats en un dispositiu superconductor d'interferència quàntica són els més sensibles a l'hora de mesurar la força del camp magnètic, mesura que, per cert, és el resultat de mesurar un canvi en el camp magnètic i no una mesura directa del valor absolut del mateix.

●**Magnetòmetre de “Bombeig Òptic”**: Els magnetòmetres de bombeig òptic basen el seu funcionament en l'efecte Zeeman. Aquest físic holandès va descobrir que, a l'aplicar un camp magnètic a àtoms que estan emetent o absorbint llum, es produeix una divisió en les línies espectrals dels àtoms de manera que aquestes queden molt més juntes que les línies normals i el seu interval de freqüència passa a ser proporcional a la magnitud del camp aplicat. Com els àtoms sols poden estar en uns nivells d'energia concrets, aquest tipus de magnetòmetre el que fa és estimular-los òpticament de manera que es súper pobla un estat d'energia d'una de les línies espectrals per a després, mitjançant un camp magnètic de ràdio freqüència RF, despoblar-lo. La freqüència RF necessària per a realitzar la despoblació és igual a la diferència espectral de línies produïdes per un camp magnètic i, per tant, és proporcional a la força del camp magnètic.

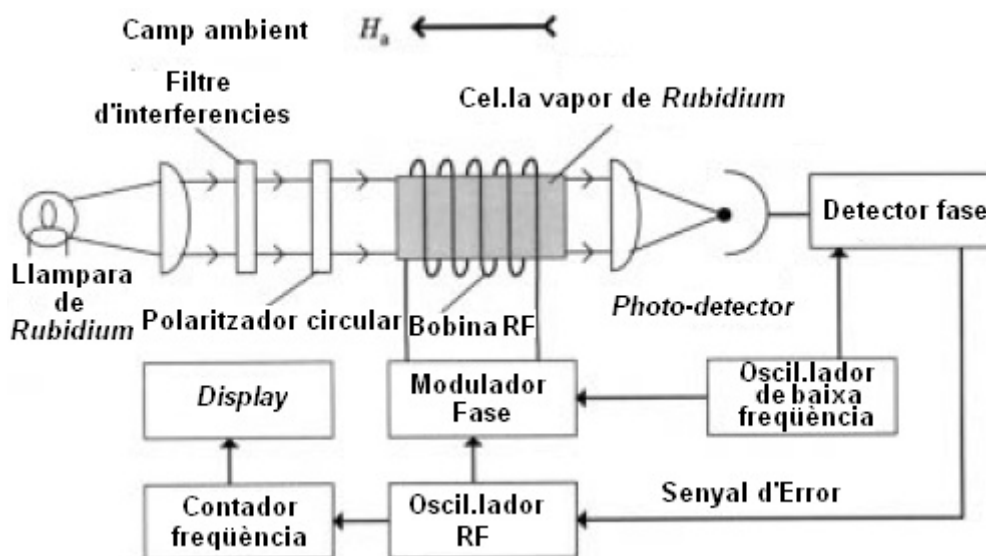


Figura 43: Esquema bàsic d'un magnetòmetre Servoed

Tal i com podem deduir del diagrama de blocs de la figura 43 i del que hem dit fins ara d'aquests sensors, aquests magnetòmetres reben el seu nom per utilitzar una bombeta per a excitar als electrons i produir canvis en els seus nivells d'energia. En el cas de la figura que ens ocupa, s'utilitza una bombeta de vapor de Rubidium per a excitar els electrons, però podria utilitzar-se qualsevol altra sempre que continguera vapor d'un element químic àlcali, com ara el Potassi o el Liti, ja que aquests mostren efectes més pronunciats a l'efecte Zeeman. Independentment de la bombeta utilitzada, mentre aquesta no estiga alimentada, els electrons estaran en repòs, però una vegada es connecte la bombeta i s'excite als electrons amb una llum polaritzada de longitud d'ona de 794.8nm, els electrons començaran a absorbir fotons i a canviar d'estat energètic d'acord amb les regles de transició. Açò vol dir que els electrons que estiguen en els nivells més baixos pujaran als més alts per a després caure a un nivell inferior, a excepció dels que estiguen en el nivell més alt, pues al no hi haure un nivell superior, no podran canviar de nivell. D'esta manera, el nivell més alt va omplint-se mentre els nivells inferiors van buidant-se fins que no continguem més electrons, moment en el que s'atura el bombeig. Per a que aquest no pare, el magnetòmetre de bombeig òptic es re-alimenta positivament amb la bobina de RF de manera que oscil·le a la freqüència de Larmor<sup>10</sup> doncs, al aplicar en angle recte al camp magnètic desconegut un petit camp de RF magnètic en la freqüència Larmor, aconseguirem que els electrons del nivell més alt baixen als nivells inferiors i, d'eixa manera, tornen a estar disponibles per ser bombejats a nivells superiors. En situacions normals, per a conèixer el valor del camp, sols hauríem de mesurar la freqüència RF, però en cas d'estar utilitzant com a oscil·lador RF un oscil·lador lineal controlat per

10 Freqüència de precessió d'una partícula carregada quan el seu moviment queda baix la influència d'un camp magnètic i una força central.

voltatge, hi hauria prou amb mesurar la seva tensió de control ja que esta seria una mesura indirecta de la freqüència de *Larmor*.

Sí per algun motiu el oscil·lador de RF no estigues centrat en la freqüència de *Larmor*, l'eixida del fotodetector canviaria de fase produint d'eixa manera una senyal d'error que s'utilitzaria per a corregir la freqüència d'oscil·lació del oscil·lador.

Els magnetòmetres d'aquest tipus poden utilitzar-se per a mesurar el camp magnètic a una velocitat de mostreig molt més alta que en un magnetòmetre de protons i solen aconseguir una resolució més alta. Tant la velocitat de mostreig com la resolució, són paràmetres interdependents.

● **Magnetòmetre de protons o de precessió protònica:** aquests sensors obtenen el valor del camp magnètic a partir de la freqüència a la que ressonen els protons en l'interior del camp magnètic a mesurar. Açò és possible perquè els protons experimenten un moviment de precessió quan es troben en l'interior d'un camp magnètic pel que, com la freqüència de precessió depèn sols de constants atòmiques i de la força del camp magnètic ambiental, la resolució a la que pot arribar un magnetòmetre d'aquest tipus és de tan sols una part per milió, cosa que els fa ideals per a mesurar camps magnètics de molt poca intensitat.

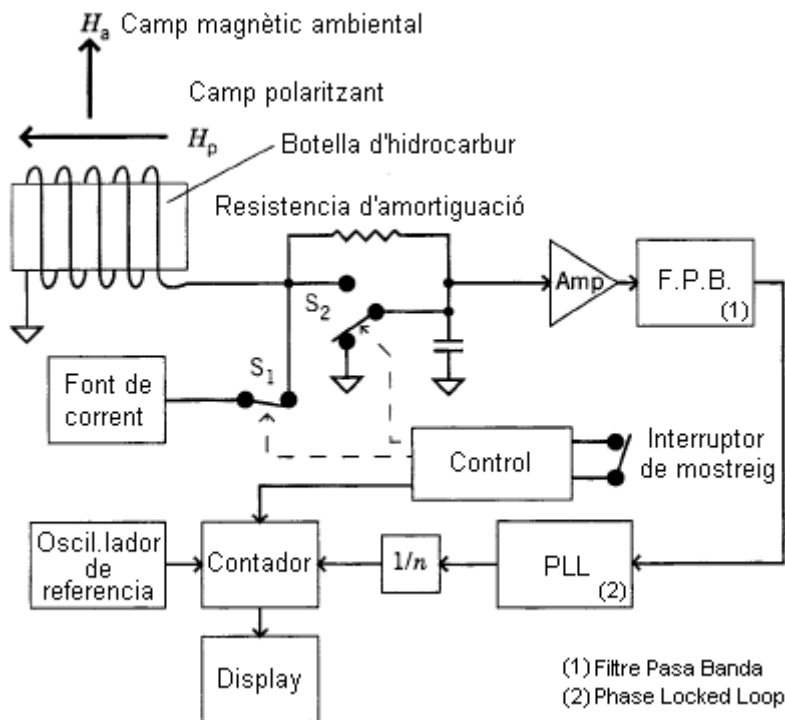


Figura 44: Diagrama de blocs d'un magnetòmetre de protons típic

Com podem observar en la figura 44, un magnetòmetre de protons consta d'una botella plena d'un hidrocarbur sobre la qual s'enrotlla un solenoide. Aquest solenoide servirà tant per a polaritzar el nucli dels protons del hidrocarbur com per a detectar el moviment de precessió dels mateixos. Abans de tancar S1 i d'aplicar el camp de polarització, el moment magnètic dels protons està alineat aleatòriament i la xarxa de magnetització és zero, però en el moment que s'aplique aquest camp, els protons experimentaran un moviment de precessió el qual induirà als els protons a orientant-se en la direcció del camp de polarització. Una vegada passe açò, sí interrompem la corrent obrint S1, obligarem als protons a re-alinear-se d'acord amb el camp magnètic ambiental, fet que produirà en aquests un moviment de precessió de freqüència proporcional al camp magnètic ambiental. Esta freqüència serà mostrejada i mesurada pel sistema de control per a posteriorment ser mostrada a l'usuari.

Les característiques d'aquest magnetòmetre el fan ideal per a usos generals tot i que, fins l'aparició d'altres tipus de sensors, era ideal per a l'exploració de minerals ja que el seu ample de banda és de tan sols un parell de Hertz, característica que no li permet mesurar variacions ràpides del camp magnètic. Per aquests motius normalment s'utilitza per a calibrar dispositius que generen camp magnètic o altres sensors magnètics.

### **3.5.3- Aplicacions dels magnetòmetres**

Els magnetòmetres tenen una gamma molt diversa d'aplicacions degut a la gran quantitat de camps en la que el magnetisme té alguna relació. Això vol dir que actualment ens els podem trobar en indústries tan dispars com la militar i la biològica:

● **Indústria automobilística:** els magnetòmetres d'efecte *Hall* dominen aquesta indústria ja que, per norma general, tots els sistemes de frenat ABS, és a dir, d'antiblocatge de rodes, utilitzen un sensor d'aquest tipus per a detectar quan la roda ha deixat de girar.

● **Arqueologia:** els magnetòmetres de nucli saturat solen utilitzar-se en aquest camp per a ajudar a localitzar vaixells afonats o objectes soterrats baix la terra.

● **Explotacions de recursos naturals:** en aquest sector s'utilitzen per a detectar perills a l'hora de perforar la superfície de la terra com poden ser les intrusions basàltiques.

● **Indústria militar:** els magnetòmetres en part existeixen per aquesta indústria ja que els magnetòmetres de nucli saturat van ser millorats durant la segona guerra mundial per tal que els avions pogueren detectar submarins. A part d'aquest ús, també se'ls utilitza per a guiar míssils o per a detectar artefactes explosius soterrats.

● **Entreteniment:** la gran majoria de *smartphones* actuals porten en el seu interior magnetòmetres els quals són utilitzats a mode de brúixoles per moltes aplicacions. A més, recentment, investigadors de *Deutsche Telekom* han afirmat haver utilitzat els magnetòmetres dels mòbils per a interactuar amb ell sense tocar-lo. Segons els mateixos això és possible si l'usuari subjecta algun objecte metàl·lic en la mà i s'utilitza el magnetòmetre del mòbil per a detectar canvis magnètics al voltant del mateix.

● **Navegació i guiat d'objectes:** els enginyers de satèl·lits solen utilitzar magnetòmetres *Fluxgate* per a controlar i determinar l'altitud a la que es troba el mateix o un objecte mòbil.

● **Geologia:** molts minerals tenen propietats magnètiques pel que utilitzant aquests sensors es poden detectar formacions dels mateixos i utilitzar-les per a estudiar el moviment de les plaques tectòniques i, d'eixa manera, detectar terratrèmols. Com les variacions del camp magnètic en aquests camps són petites i lentes, els magnetòmetres *Fluxgate* i els de precisió protònica són ideals en aquest tipus d'aplicació.

● **Biologia:** les senyals elèctriques produïdes pels sers vius també produeixen camps magnètics, pel que podríem utilitzar un magnetòmetre per a captar-los. Com aquests camps tenen tan poca intensitat s'han d'utilitzar els magnetòmetres més sensibles que existeixen: els SQUID. Aquest tipus de magnetòmetres s'utilitzen principalment en els equips de ressonància magnètica que podem trobar en qualsevol hospital, però també s'utilitzen en els detectors d'activitat neuronal, o magneto-encefalogrames, i en els magneto-cardiogrames, uns monitors d'activitat cardíaca basats en la tècnica *Magnetic Field Imaging*, la qual és una tècnica no invasiva per a detectar i registrar els batecs del cor sense efectes secundaris.

### 3.6- Sensors MEMS: els “tot en un”

Sí analitzem tranquil·lament els diferents tipus de sensors dels que hem estat parlant en les seccions anteriors, caurem en el detall que, a excepció dels giroscopis d'estructura vibratòria, tots tenen una grandària considerable i açò, depenent de l'aplicació, pot ser un problema ja que no sempre podem disposar del espai suficient per a instal·lar-los o inclús, en el cas dels acceleròmetres piezoelèctrics o dels giroscopis de tipus FOG, se'n isquen del nostre pressupost i facen totalment inviable el projecte. Sí es donara aquest cas, interessa saber que, a part de la modelació mecànica tradicional, existeixen altres modelacions amb una precisió similar i, a més, de menor cost i grandària: els sensors de tecnologia MEMS (*Micro ElectroMechanical Systems*).

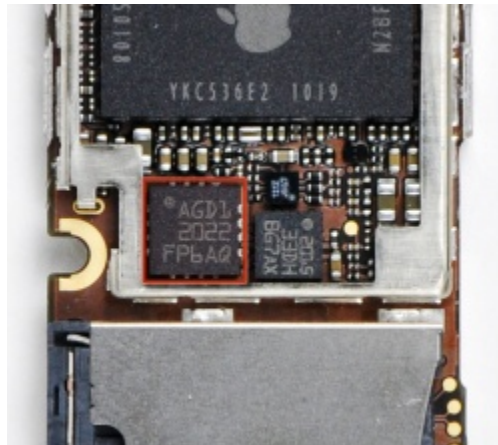


Figura 45: En el requadre roig, el giroscopi MEMS del Iphone 4

Els sensors MEMS estan formats per components de una grandària que està entre 1 i 100 micròmetres i, en conjunt, solen ocupar de 20 micròmetres a 1 mili metre de l'interior d'un substrat de silicònia o polímer. En este espai tan reduït podem trobar integrats tots els elements mecànics que formen el sensor junt a l'electrònica de control i d'adequació de les senyals dels elements mecànics.

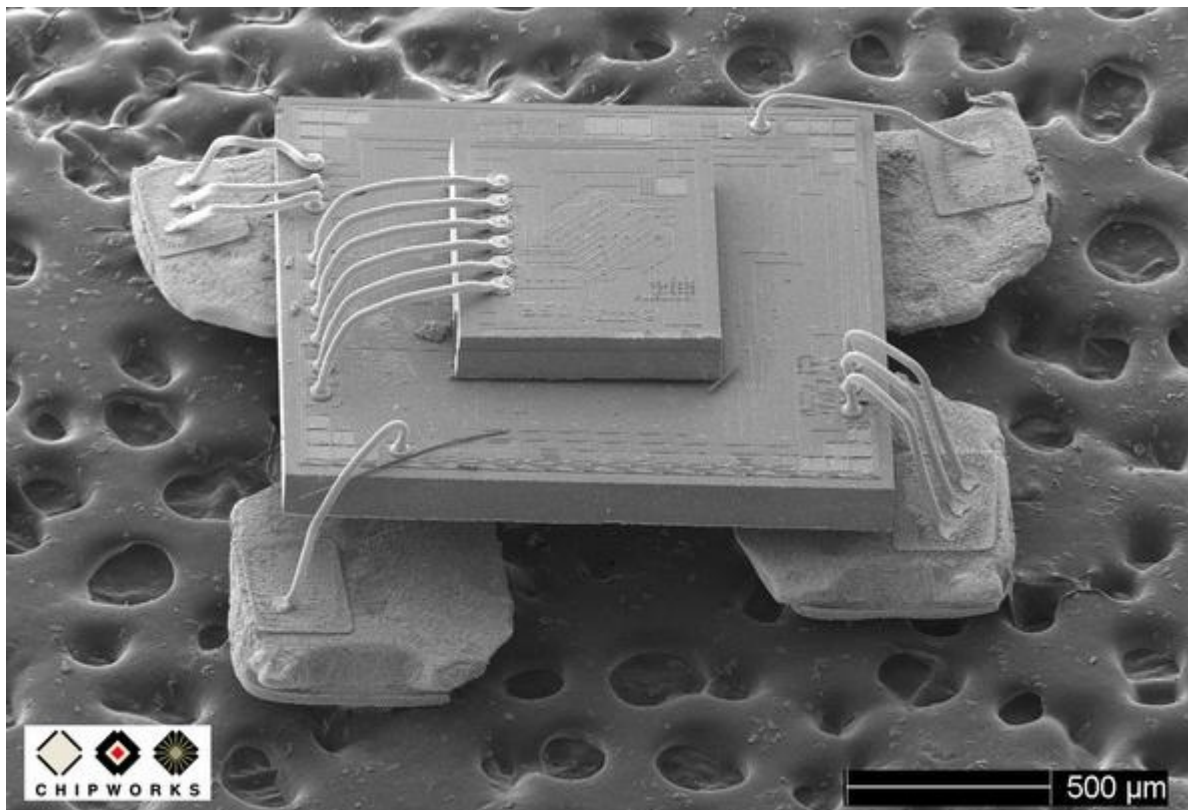


Figura 46: A sobre del ressonador d'un giroscopi MEMS, el circuit ASIC encarregat de processar les senyals d'eixida del mateix.

La fabricació d'aquests sensors, a excepció de la part de control, és molt complexa ja que les regles de “física tradicional” no sempre funcionen en grandàries tan diminutes, d'ací que no puguem ser fabricats utilitzant tècniques de producció comuns i es tinga que recórrer a tècniques més exòtiques com la *bulk micromachining* o la *surface micromachining*. Aquestes dos tècniques el que fan és menjar-se les zones del substrat que estan buides en el disseny original i afegir, normalment mitjançant tècniques de “dibuix”, les zones que falten. La diferència està en que una fa el dibuix “a la força” submergint el substrat de silicó en acid mentre que en l'altra es fa mitjançant l'afegiment de successives capes de la litografia, com sí de l'animació de una pel·lícula de dibuixos animats es tractara.

Tot i la complexitat del disseny i producció d'aquest tipus de sensor, en l'actualitat no és difícil trobar-se dins d'un mateix encapsulat, junt a la part de control, un giroscopi, un acceleròmetre i un magnetòmetre per cada eix del sistema, és a dir, fins 9 sensors en un cub d'uns 3 cm<sup>3</sup> de superfície, cosa que en la gerga d'aquests dispositius rep el nom de *9 degrees of freedom...* o el que és el mateix: 9 graus de llibertat. Cal destacar que, tot i que 9 graus de llibertat poden semblar molts, actualment també es comercialitzen dispositius amb més graus de llibertat ja que alguns inclús afegeixen sensors de pressió per a augmentar la precisió a l'hora d'obtenir la localització del sistema. Això sí, tanta llibertat té un preu: 480 euros sí demanem directament al fabricant almenys 100 unitats. De totes maneres, al ser un sistema que sol necessitaria una alimentació i una CPU externa per a funcionar, se'l podria considerar un cas excepcional ja que ni els giroscopis ni els acceleròmetres de tres eixos solen passar dels 10 euros de preu final al client.



Figura 47: ADIS 16407: sensor MEMS de 10 graus de llibertat,

Independentment del preu i, sense cap dubte, el millor d'aquesta tecnologia és que simplifica el disseny electrònic de un sistema de navegació, doncs el sistema no necessita fer un tractament previ de les senyals enviades per aquests sensors pues aquests ja incorporen en el mateix encapsulat l'electrònica necessària per a adaptar i digitalitzar els nivells de tensió proporcionats pels diferents sensors i, posteriorment, enviar-los al exterior a través d'un *interface* serie com el I2C o el SPI per a que, el processador del sistema, interprete aquesta informació i actue sobre els restants dispositius que controla per tal d'aconseguir la resposta desitjada del sistema.

Des del punt de vista del funcionament, els giroscopis, els acceleròmetres i els magnetòmetres de tipus MEMS es basen en els mateixos principis de funcionament que els seus “germans grans”, és a dir, que els podem trobar basats en els efectes piezoresistius, piezoelèctrics, capacitius, òptics o ressonants, sempre clar, que eixe sensor es puga fabricar basant-se en el principi corresponent. Els únics que no segueixen aquesta norma són els inclinòmetres ja que aquests utilitzen acceleròmetres per a obtenir l'angle d'inclinació de l'objecte respecte a la força de gravetat.

Tal i com podem veure en la figura 48, un acceleròmetre MEMS capacitiu està format pràcticament pels mateixos elements que un acceleròmetre capacitiu estàndard, sols que d'una grandària invisible a l'ull humà. Així doncs, en un acceleròmetre MEMS també tenim un moll en el que s'instal·len galgues extensomètriques per a augmentar la precisió de la mesura i sobre el qual es penja la massa sísmica. Al voltant del mateix se situen les plaques dels condensadors de manera que, quan es produïska una acceleració, la massa sortirà del repòs i la distància entre plaques es veurà modificada, produint d'aquesta manera una diferència de potencial proporcional a l'acceleració.

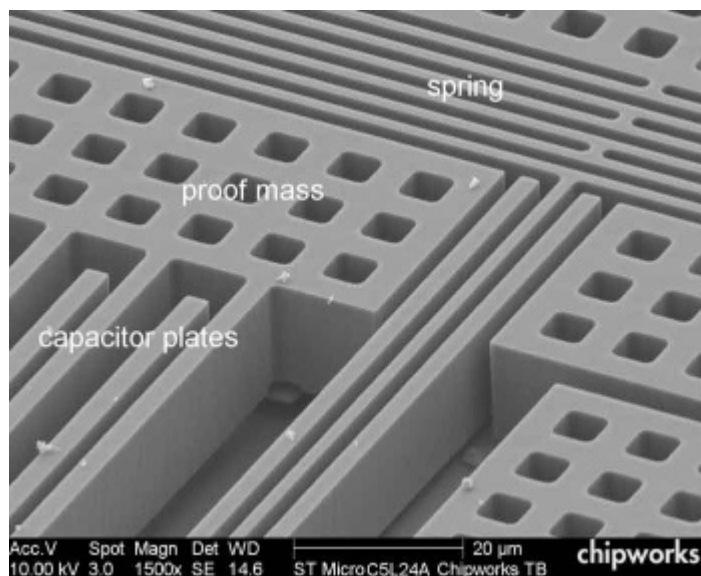


Figura 48: Estructura parcial micromètrica d'un acceleròmetre MEMS.

El cas d'un giroscopi MEMS és similar a l'anterior sols que, per raons obvies, el model seguit és el d'un giroscopi d'estructura ressonant, pel que la base del seu funcionament és la dels giroscopis d'aquest tipus. Com podem veure en la figura 49, aquest detall fa que el disseny intern d'un giroscopi MEMS siga més complex i espectacular que el d'un acceleròmetre per la dificultat existent a l'hora de crear una estructura ressonant en tan poc lloc. Esta complexitat també es veu reflectida en el preu, que sense ser cap exageració, arriba a doblar al d'un acceleròmetre.

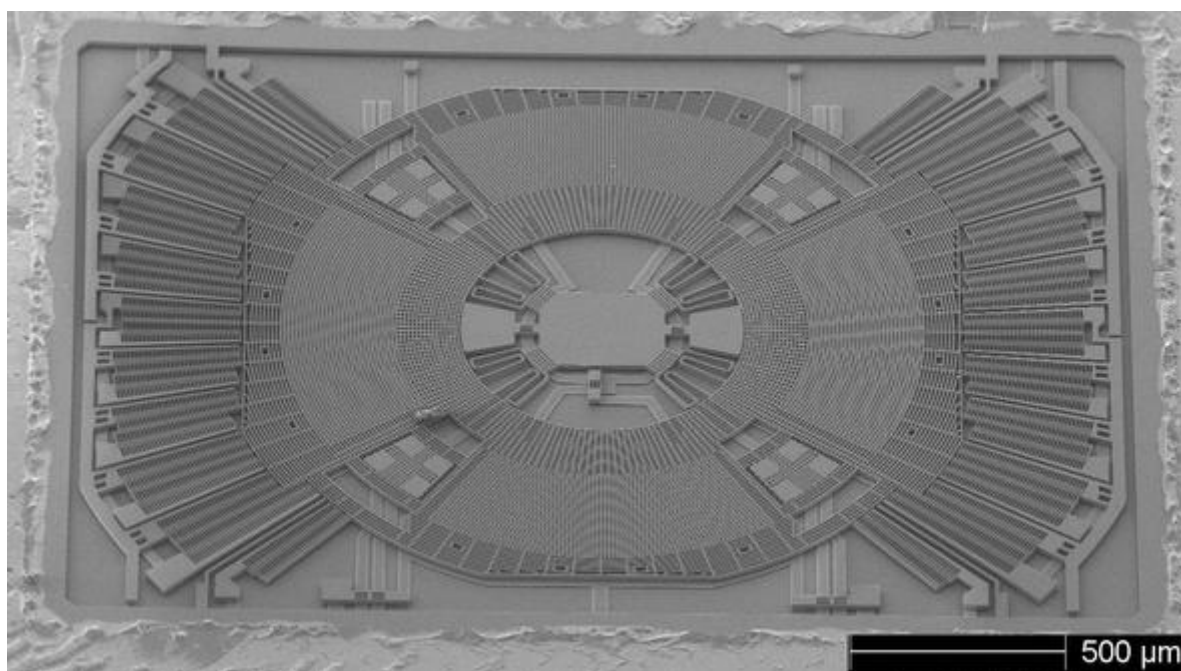


Figura 49: Estructura interna d'un giroscopi de 2 eixos.

Com estàvem dient, els giroscopi MEMS solen utilitzar la força *Coriolis* per a mesurar la velocitat de gir experimentada pel sistema. Per a entendre com es fa açò anem a suposar que estem en el centre d'una plataforma rotatòria, com la de la figura 50, caminant a certa velocitat en sentit contrari al sentit de rotació per tal de mantenir una posició relativa al sòl. Sí ara ens desplaçarem a la part externa d'aquesta, observarem que serà necessari caminar més ràpid si volem continuar mantenint la posició inicial relativa. Aquest augment de la velocitat és l'acceleració *Coriolis* i és perpendicular al nostre moviment radial.

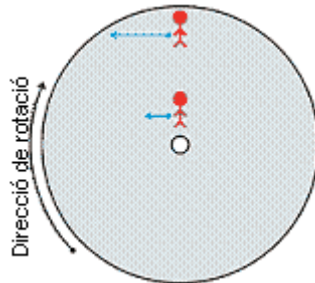


Figura 50: Representació de l'acceleració de *Coriolis*

Els giroscopis MEMS prenen partit d'aquest efecte situant sobre la plataforma giratòria una massa ressonant la qual està micro-mecanitzada en un substrat de silicó i unida a un marc del material anterior de manera que sols pugui ressonar en una única direcció. Tal i com s'observa en la figura 51, quan la massa es desplaça cap a la part externa de la plataforma experimentarà una força cap a l'esquerra mentre que el sentit de la força serà cap a la dreta sí el desplaçament és cap al centre de rotació.

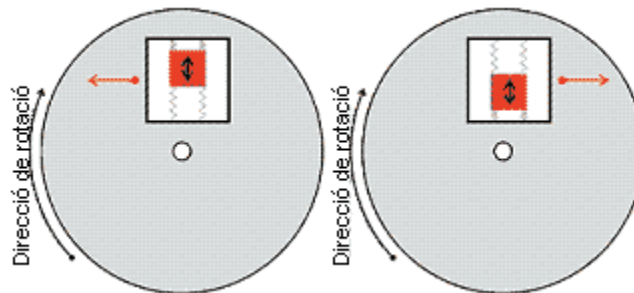


Figura 51: Efecte *Coriolis* sobre una massa ressonant suspesa en l'interior d'un marc

Per tal de mesurar l'acceleració *Coriolis* s'utilitza un muntatge similar al de la figura 52. En esta figura podem veure com la massa ressonant s'uneix al substrat intern del marc mitjançant molls de manera que esta forma un angle de  $90^\circ$  amb el moviment de ressonància. Com aquest muntatge està situat sobre una plataforma rotatòria, quan la massa ressonant experimente un moviment, el conjunt experimentarà l'acceleració de *Coriolis* la qual serà detectada pels "dits" de sensat, doncs al desplaçar-se el marc intern, la separació dels "dits" de sensat canvia i, com aquests venen a ser com les plaques d'un condensador, mesurant la nova capacitancia obtindrem el valor d'aquesta acceleració la qual, com sabem, és proporcional a la velocitat angular de rotació.

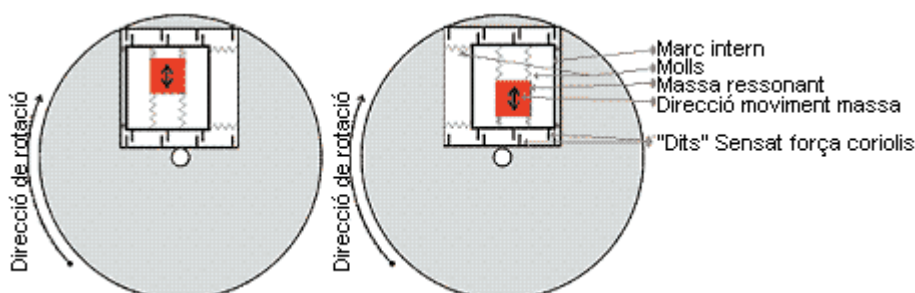


Figura 52: Desplaçament lateral del marc i la massa degut a l'efecte *Coriolis*



Al contrari que passa en els giroscopis o els acceleròmetres, els magnetòmetres MEMS no segueixen al peu de la lletra la base de funcionament dels seus germans majors, sinó que es basen en algun dels principis elementals de funcionament d'aquests. Així doncs, molts dels magnetòmetres MEMS disponibles actualment utilitzen o bé l'efecte magneto resistiu descobert per *Kelvin*, o bé la força de *Lorentz*, és a dir, la força que origina l'efecte *Hall*, per a mesurar el camp magnètic.

Aquests últims basen el seu funcionament en el moviment mecànic de l'estructura MEMS produït per la força de *Lorentz*, la qual actua sobre el portador de corrent en un camp magnètic. El moviment mecànic de la microestructura és detectat o bé elèctricament mitjançant transductors piezoresistius, o bé òpticament utilitzant LEDs. Generalment l'estructura mecànica es fa treballar a la seva freqüència de ressonància per a, d'eixa manera, obtenir la màxima senyal d'eixida possible.

| “Grau” del Sensor | Bias Accl. (mg) | Bias Accl. (ms <sup>-2</sup> )        | Bias Giro. (° hr <sup>-1</sup> ) | Bias Giro. (Rad s <sup>-1</sup> )       |
|-------------------|-----------------|---------------------------------------|----------------------------------|---|
| Marina            | 0.01            | 10 <sup>-4</sup>                      | 0.001                            | 5x10 <sup>-9</sup>                      |
| Aviació           | 0.03 - 0.1      | 3x10 <sup>-3</sup> - 10 <sup>-3</sup> | 0.01                             | 5x10 <sup>-8</sup>                      |
| Mig               | 0.1 - 1         | 10 <sup>-3</sup> - 10 <sup>-2</sup>   | 0.1                              | 5x10 <sup>-7</sup>                      |
| Tàctic            | 01/10/12        | 10 <sup>-2</sup> - 10 <sup>-1</sup>   | 1 - 100                          | 5x10 <sup>-6</sup> - 5x10 <sup>-4</sup> |
| Automoció         | >10             | >10 <sup>-1</sup>                     | >100                             | >5x10 <sup>-4</sup>                     |

Taula 9: Classificació dels sensors MEMS segons el seu error a l'eixida.

Per últim, cal dir que els dispositius MEMS, a pesar d'utilitzar alguns dels principis de funcionament dels sensors tradicionals, tenen precisions diferents i, al llarg del temps, la seva eixida es veu més fàcilment influenciada pels efectes de la temperatura i el soroll. Per això, com podem veure en la taula 9, als sensors MEMS també se'ls pot classificar en “graus”<sup>11</sup> segons la sensibilitat a aquests paràmetres.

### 3.6.1 Problemàtica dels dispositius MEMS: efectes de la temperatura, *Random Walk* i *Bias Stability*

Com hem vist al final de la secció anterior, els sensors MEMS, també es poden diferenciar per la seva resposta al soroll al llarg del temps. Aquesta classificació no és casual ja que, depenent de l'aplicació, un sensor fabricat amb aquesta tecnologia pot deixar-nos de ser útil pues no és el mateix detectar un canvi de velocitat brusc, que tenir que detectar un petit canvi en la mateixa. Per això, a l'hora d'examinar la fulla de característiques d'un sensor MEMS, és important fixar-se en un parell de paràmetres: el *Random Walk* i la *Bias Stability* al llarg del temps.

|                            | GG1320AN (Giroscopi laser) | GG5300 (Giroscopi triaxial MEMS) |
|----------------------------|----------------------------|----------------------------------|
| Dimensions                 | 88x88x45 mm                | 50x50x30 mm                      |
| Pes                        | 454 gr.                    | 136 gr.                          |
| Temps d'arranc             | <4seg.                     | <1seg.                           |
| Potència                   | 1.6W @ 15V, 0.375W @ 5V    | <800 mA @ 5V                     |
| T <sup>a</sup> de treball  | -54° a 75°C                | -45° a 85°C                      |
| <i>Angular Random Walk</i> | 0.0035°/√                  | 0.2°/√                           |
| <i>Bias Stability</i>      | 0.0035°/h                  | <70°/h                           |

Taula 10: Comparació entre un giroscopi òptic i un giroscopi MEMS

<sup>11</sup> Traducció literal de “Grade”

La taula 10 mostra dos exemples ben diferenciats: el d'un giroscopi làser i el d'un giroscopi MEMS. Com podem veure en la mateixa, les diferències que hem comentat anteriorment queden totalment reflectides en la taula: grandària a favor dels sensors MEMS i precisió/fiabilitat a favor dels sensors basats en una altra tecnologia. Òbviament, esta afirmació és el resultat de comparar dos valors d'aquesta taula dels que fins ara no hem parlat: *Angular Random Walk* i *Bias Stability*. Per a entendre com afecten aquests paràmetres en la resposta i saber quin sensor pot ajustar-se millor a les nostres necessitats hem de saber que consisteix cadascun i com afecta al sensor dependent de si aquest és un giroscopi o un acceleròmetre.

Així doncs, el **Random Walk** és el soroll que apareix a l'eixida del sensor degut al soroll termo-mecànic produït pel constant moviment dels elements que formen el sensor. Aquest soroll fa que les mostres obtingudes del sensor estiguen pertorbades per una seqüència de soroll blanc el qual és simplement una seqüència de variables aleatòries no correlacionades de mitja zero. En aquest cas, cada variable aleatòria està distribuïda d'igual forma i té una variança finita igual a  $\sigma^2$ .

Per a veure quin efecte produeix aquest soroll en la senyal integrada, podem fer un senzill anàlisi si resollem les integrals mitjançant el mètode del rectangle<sup>12</sup>. Aleshores, si suposem que  $N_i$  és l'enèsima variable aleatòria de la seqüència del soroll blanc, que cada  $N_i$  està distribuït en la seqüència de manera idèntica segons  $E(N_i)=E(N)=0$  i amb variança finita  $Var(N_i)=Var(N)=\sigma^2$ , per la definició de soroll blanc tenim  $Conv(N_i,N_j)=0$  per a tot  $i \neq j$ , pel que el resultat d'integrar, o doble integrar en el cas del acceleròmetre, la senyal del soroll blanc  $\epsilon(t)$  en un interval de temps  $t=n \cdot \delta t$  és:

$$\int_0^t \epsilon(\tau) d\tau = \delta t \sum_{i=1}^n N_i$$

per al cas d'un giroscopi, mentre que en el d'un acceleròmetre val:

$$\int_0^t \int_0^t \epsilon(\tau) d\tau d\tau = \delta t \sum_{i=1}^n \delta t \sum_{j=1}^i N_j = \delta t^2 \sum_{i=1}^n (n-i+1) N_i$$

on  $n$  és el nombre de mostres llegides del dispositiu durant el període temps  $t$  i  $\delta t$  és el temps entre les successives mostres. Utilitzant la formula  $E(aX+bY)=aE(x)+bE(y)$  i  $Var(aX+bY)=a^2Var(X)+b^2Var(Y)+2abConv(X,Y)$ , on  $a$  i  $b$  són constants i  $X$  i  $Y$  són variables aleatòries, es dedueix l'error i la variança :

$$E\left(\int_0^t \epsilon(\tau) d\tau\right) = \delta t \cdot n \cdot E(N) = 0$$

$$Var\left(\int_0^t \epsilon(\tau) d\tau\right) = \delta t^2 \cdot n \cdot Var(N) = \delta t \cdot t \cdot \sigma^2.$$

per al cas del giroscopi i, per al cas de l'acceleròmetre, són iguals a:

$$E\left(\int_0^t \int_0^t \epsilon(\tau) d\tau d\tau\right) = \delta t^2 \sum_{i=1}^n (n-i+1)E(N_i) = 0$$

<sup>12</sup> Aquest mètode d'integració consisteix en fer passar a través del punt  $(a, f(a))$  a un polinomi d'ordre zero que actuarà com a funció interpoladora.

$$\text{Var} \left( \int_0^t \int_0^t \epsilon(\tau) d\tau d\tau \right) = \delta t^4 \sum_{i=1}^n (n-i+1)^2 \text{Var}(N_i) = \frac{\delta t^4 n(n+1)(2n+1)}{6} \text{Var}(N) \approx \frac{1}{3} \cdot \delta t \cdot t^3 \cdot \sigma^2$$

i, per tant, com podem veure en els dos casos, el soroll introdueix un error *random walk* de mitja zero en la senyal integrada de la desviació estàndard. L'altra conclusió que podem extreure d'aquests resultats és que, en el cas dels giroscopis, la desviació estàndard creix de manera proporcional amb l'arrel quadrat del temps tal com mostra la seva equació:

$$\sigma_\theta(t) = \sigma \cdot \sqrt{\delta t \cdot t}$$

En el cas d'un acceleròmetre, sempre que el temps de mostreig siga elevat, és a dir, sempre que  $\delta t$  siga menut, podem aproximar de la següent forma:

$$\sigma_s(t) \approx \sigma \cdot t^{3/2} \cdot \sqrt{\frac{\delta t}{3}}$$

Com a l'usuari final li interessa saber com afecta el soroll a la senyal integrada, els fabricants solen especificar el soroll utilitzant una mesura del *random walk*. En el cas dels acceleròmetres, com hi ha una doble integració per a calcular el seu valor, rep el nom de *Velocity Random Walk*, mentre que en els giroscopis rep el nom de *Angular Random Walk*.

Tot açò vol dir que sí estem utilitzant el giroscopi triaxial MEMS amb un *Angular Random Walk* de  $0.2^\circ/\sqrt{h}$ , al cap d'una hora tindrem una desviació de  $0,2^\circ$  mentre que esta passarà a ser de  $0,28^\circ$  ( $0.2 \times \sqrt{2}$ ) passades les dos hores, de  $0,35^\circ$  ( $0.2 \times \sqrt{3}$ ) en la tercera hora, etc. És per aquest motiu que, sí ens interessa mantenir l'orientació, haurem de tenir en compte estes desviacions ja que passades unes hores sense moure'ns, el dispositiu pot pensar que l'orientació ha canviat uns quants graus des que es va iniciar el sistema quan no té perquè haver sigut d'eixa manera.

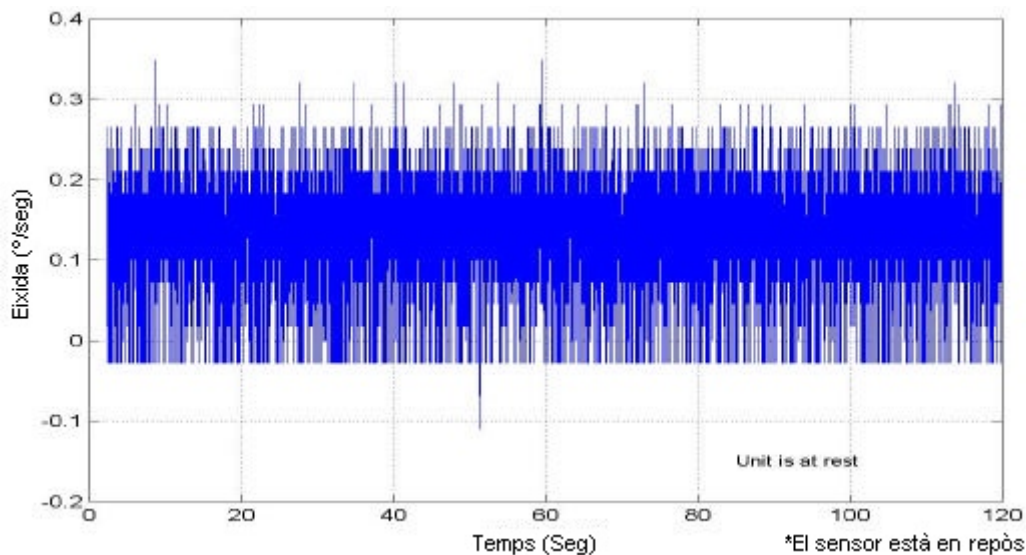


Figura 53: exemple de l'eixida d'un giroscopi VG-700A de la casa *Crosbow* mentre està en repòs

Un altre paràmetre a tenir en compte és el ***Bias Stability*** del sensor. Açò es deu a que l'eixida d'un sensor MEMS no és estable i, tal i com podem veure en la figura 53, sol oscil·lar al llarg del temps degut al soroll de pàrpelleig de l'electrònica del mateix. Aquest soroll té un espectre  $1/f$  i els seus efectes s'aprecien en els components electrònics quan aquests estan treballant a baixes freqüències, pues a altes freqüències aquest soroll sol ser emmascarat pel soroll blanc. Per aquest motiu, a les fluctuacions que apareixen en l'eixida per aquest tipus de soroll se les sol tractar com *random walk*

i, per això, aquesta mesura descriu com pot canviar l'eixida del dispositiu en condicions constants al llarg d'un període de temps d'uns 100 segons de duració.

Per a entendre millor aquest paràmetre anem a suposar que  $B_t$  és una eixida coneguda per al temps  $t$ , aleshores, si estem utilitzant un giroscopi amb un *bias stability* de  $0.01^\circ/\text{h}$  als 100 segons, tindrem que l'eixida en l'instant  $(t+100)$  serà una variable aleatòria amb un valor esperat de  $B_t$  i una desviació estàndard de  $0.01^\circ/\text{h}$ . Com a mesura que passa el temps l'eixida va canviant, és pot dir que aquest paràmetre produeix un efecte idèntic al del *random walk*. La diferència està en que si fos el mateix efecte, l'eixida del sensor augmentaria sense limit a mesura que anés passant el temps, però no és d'aquesta manera, pel que l'efecte de *random walk* de segon i tercer<sup>13</sup> ordre al qual s'aproxima quan es parla d'aquest paràmetre en un giroscopi i un acceleròmetre, sols és vàlid en períodes de temps curts.

L'últim efecte físic del que ens queda parlar i que pot produir una desviació en l'eixida del sensor MEMS és la temperatura. A l'igual que en els casos anteriors, l'eixida d'un sensor d'aquest tipus canvia en el temps segons aquest paràmetre de forma lineal, pel que a mesura que el sistema vaja calfant-se, la seva eixida anirà augmentant progressivament al ritme indicat pel paràmetre *Bias Temperature Coefficient*. Degut a aquest efecte, els sensors d'inèrcia incorporen internament un termòmetre que permet a l'usuari anar compensant les desviacions en l'eixida segons la temperatura. En cas que ens interesse mantenir l'orientació serà recomanable llegir-lo periòdicament per a d'esta manera compensar els efectes de la temperatura en l'eixida del sensor.

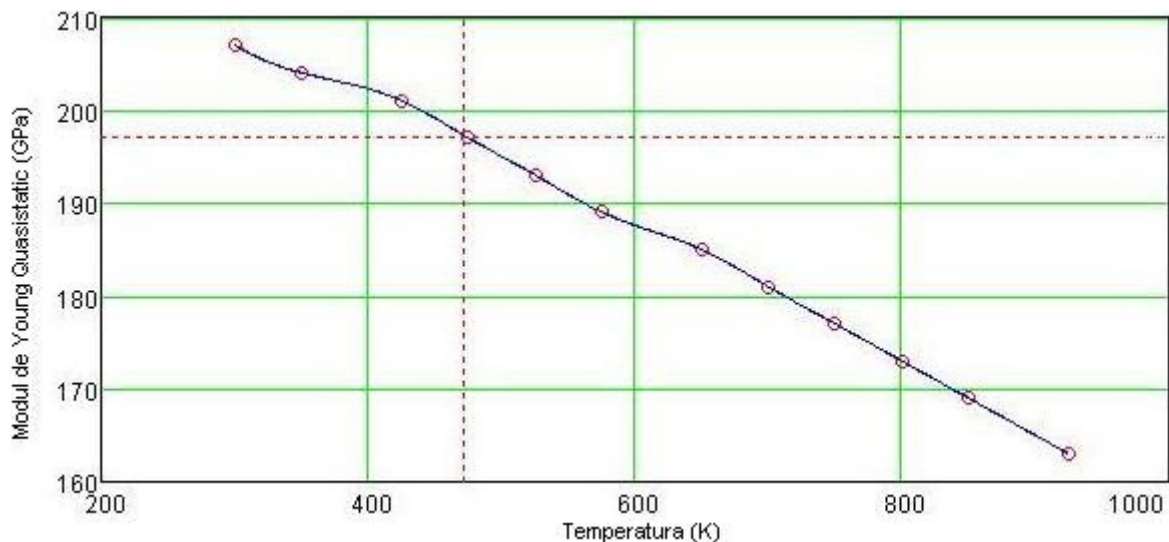


Figura 54: efectes de la temperatura sobre el mòdul Young del Niquel

En el que respecta als magnetòmetres MEMS, com mostra la figura 54, quan la temperatura augmenta, el mòdul de *Young* del material utilitzat per fabricar la microestructura disminueix produint una pèrdua de rigidesa en la mateixa. Mentrestant, al mateix temps que augmenta la temperatura, també augmenten l'expansió tèrmica i la conductivitat tèrmica pel que apareixerà una pressió interna sobre l'estructura mòbil la qual desplaçarà la freqüència de ressonància de l'estructura, cosa que, a efectes pràctics, es tradueix com a soroll sobre la freqüència de ressonància i en la tensió d'eixida del sensor. A més, a part del soroll provocat per la temperatura en la microestructura mecànica del sensor, també haurem de considerar el soroll que produirà la temperatura en el circuit electrònic encarregat de captar la senyal d'eixida de l'estructura mecànica, pel que s'haurà d'afegir un circuit extern que s'encarregue de compensar i corregir el soroll de fluctuació mecànica si estem captant la senyal amb un sensor òptic, o el soroll de *Johnson*<sup>14</sup> en cas de capturar-la mitjançant un sensor piezoresistiu.

<sup>13</sup> L'orde d'un *random walk* és en realitat el nombre d'integrals que s'han efectuat sobre el *random walk* original

<sup>14</sup> Soroll creat per l'excitació tèrmica dels electrons

## 4- Sistemes de Navegació Inercial, una aplicació específica dels sensors de moviment i dels sistemes de localització d'objectes

### 4.1- Introducció: Sistemes de Navegació, Navegació Inercial i Sistemes Guiats

Fins ara hem estat parlant de com funcionen els sistemes de posicionament/localització d'objectes mitjançant satèl·lits o de com funcionen els sensors que ens faciliten informació sobre l'orientació, la inclinació o l'acceleració sobre un o més plans, però no hem arribat a parlar dels resultats d'unir aquests dispositius en un mateix dispositiu. Siga com siga, per a poder entendre millor les diferències existents entre aquests sistemes, el millor és definir-los individualment:

• **Sistema de Navegació:** tal i com em vist en l'apartat sobre els sistemes de localització per satèl·lit, un sistema de Navegació és aquell el qual permet localitzar un objecte sobre un sistema de referència, és a dir, un sistema que generalment es limitarà a poc més que facilitar la localització d'un objecte ja siga utilitzant un satèl·lit, un sistema de coordenades, etc.



Figura 55: Sistema de Navegació basat en GPS

• **Sistema de Navegació Inercial:** a l'igual que un sistema de navegació, aquest també ens proporciona una localització, sols que sense utilitzar referències externes com puga ser un satèl·lit, sinó que ho fa a partir de la informació proporcionada per un acceleròmetre i/o un giroscopi. Presenten l'inconvenient de necessitar algú que els facilite la seva posició inicial i per això s'utilitzen conjuntament amb als sistemes de navegació doncs proporcionen més precisió i informació a l'operador del sistema.



Figura 56: Rada R-100F, Sistema de navegació inercial basat en giroscopi de fibra òptica.

• **Sistema Guiat:** aquests sistemes són una millora dels Sistemes de Navegació Inercial ja que incorpora totes les funcions d'aquests i, a més, permeten prescindir de la persona que controla el sistema, cosa que els converteix en sistemes totalment autònoms a l'hora de desplaçar-se.

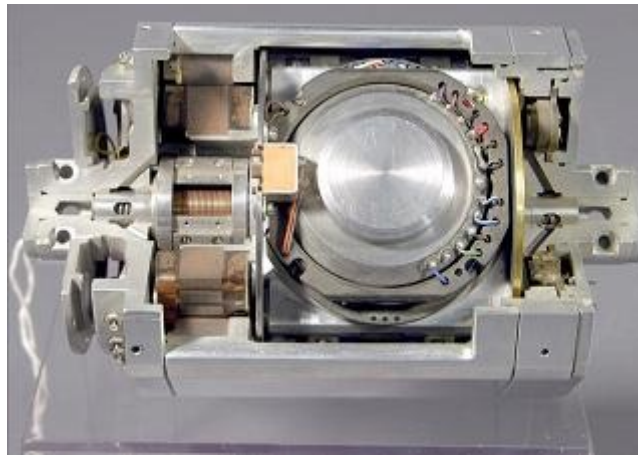


Figura 57: Sistema de guiat d'un míssil

Com ja haurem deduït d'aquestes definicions, tant un Sistema de Navegació Inercial com un Sistema guiat poden detectar tant un canvi en la seva posició geogràfica, com un canvi en la seva velocitat i/o un canvi en la seva orientació. Açò ho aconsegueixen mesurant l'acceleració lineal i angular que s'aplica al sistema i, com no necessiten una referència externa després de la inicialització, són immunes a les interferències.

Per a poder fer totes aquestes deduccions i càlculs, aquests dos sistemes utilitzen giroscopis i Acceleròmetres en combinació amb una computadora. Els primers s'utilitzen per a calcular la velocitat angular del sistema en el marc de referència inercial, mentre que els segons s'utilitzen per a mesurar l'acceleració lineal del sistema en el marc de referència inercial, però sols en direccions relatives al moviment del sistema. Una vegada obtingudes eixes dades, sí la computadora les integra individualment aplicant com a condició inicial el marc de referència inercial, podem saber en cada moment l'orientació i velocitat actual del sistema i, sí tornem a integrar la velocitat, obtindrem la posició inercial. Tot i que seria possible obtenir aquestes dades sense una computadora, el procés de navegació inercial es complicaria considerablement pel que l'ús d'una computadora és quasi una obligació.

Actualment, els sistemes de Guiat Inercial solen utilitzar-se en combinació amb un sistema de navegació basat en satèl·lit i amb un sistema de filtratge digital. El sistema inercial facilita dades actuals mentre que el sistema per satèl·lit s'utilitza per a corregir els errors acumulats pel sistema inercial.

Com el nostre projecte no té com a objectiu dissenyar un sistema guiat, en aquesta memòria no entrarem en detalls sobre els mateixos i simplement ens limitarem a parlar molt per damunt dels sistemes de navegació inercial aprofitant que el nostre sistema té els components necessaris per a poder ser utilitzat com a tal. Tot i això, tampoc profunditzarem molt ja que els sistemes de navegació inercial tenen una carrega teòrica important i, per tant, una problemàtica addicional que no està directament relacionada amb els objectius bàsics d'aquest projecte.

## 4.2- Història dels Sistemes de Navegació Inercial

Els sistemes de Navegació Inercial van ser creats inicialment per l'americà *Robert Goddard* amb la idea de ser utilitzats en coets de propulsió. Els experiments d'aquest científic amb sistemes giroscòpics rudimentaris van despertar l'interès d'importants científics alemanys de l'època i feu que, en la segon guerra mundial, els sistemes guiats dels alemanys combinaren dos giroscopis, un acceleròmetre lateral i una computadora analògica per a ajustar i controlar l'azimut dels míssils.

Més endavant, quan Alemanya va perdre la segon guerra mundial, els científics alemanys encarregats del disseny dels sistemes de guiat dels míssils alemanys van ser transferits a EUA on van continuar treballant pel programa Americà d'investigació militar. Allí, en 1950, van dissenyar el sistema de guiat de reserva dels míssils balístics intercontinentals *Atlas*. Aquest sistema de guiat anava a ser la combinació d'un sistema totalment autònom amb un sistema de localització de superfície i un sistema de comandament. La finalitat perseguida pel projecte va ser el principi d'una controvèrsia filosòfica en la que, en alguns camps, roman sense resoldre. Finalment, aquest sistema "auto contingut", per raons obvies, va prevaldre en les aplicacions de míssils balístics mentre que en el camp de l'exploració espacial es va arribar a una solució de compromís.



Figura 58: míssil intercontinental *Atlas* SM-65

En l'estiu de 1952, els Drs. *Richard Battin* i *Halcome "Hal" Laning Jr.* van començar a investigar solucions al guiat basades en l'ús de computadores. El Dr. *Laning*, amb l'ajuda de *Phil Hankins* i *Charlie Werner*, començaren a treballar en un llenguatge de programació algebraic per a l'IBM 650 al qual van anomenar MAC. Aquest llenguatge de programació és fàcilment llegible i es caracteritza per utilitzar un format de tres línies, disposar de notacions vector-matriu, *mnemonics* i sub-guions indexats. Degut a açò i a la potència del mateix, aquest sistema és el que ha servit de base al llenguatge utilitzat actualment en els transbordadors espacials : el HAL/S.

Al mateix temps, el Dr. *Battin* i el Dr. *Laning* van reprendre el treball d'anàlítica del sistema de guiat dels *Atlas* junt a altres científics que havien treballat en el mateix, pues el sistema inicial de guiat *delta* avaluava la diferència en la posició a partir d'una trajectòria de referència i, a partir d'aquesta, es calculava la velocitat VGO que devia aconseguir-se per tal de corregir la trajectòria actual i portar a zero aquesta velocitat VGO. Els fonaments matemàtics d'aquesta idea eren vàlids, però van tenir que ser descartats perquè les computadores analògiques no tenien suficient potència de càlcul per a resoldre els desafiaments inherents d'un guiat inercial precís. Aquesta problemàtica va ser resolta poc després amb com el conegut sistema de guiat Q, el qual disposava en una mateixa matriu Q les equacions de moviment, és a dir, les derivades parcials de la velocitat respecte l'eix, de manera que el producte vectorial creuat ( $v, xdv, /dt$ ), pogués ser utilitzat com les senyals bàsiques necessàries per al correcte funcionament del pilot automàtic. Tot i que aquesta tecnologia data dels anys 60, a dia de hui, continua considerant-se informació confidencial ja que els sistemes de guiat de míssils actuals basen el seu funcionament en variacions d'aquest sistema.

Anys després d'açò, la NASA va contractar a una divisió de *General Motors, Delco Electronics*, i al MIT per a que dissenyaren i, posteriorment, produïren el sistema de guiat i navegació dels mòduls lunar i de comandament *Apolo*. Per a la llançadora espacial es va utilitzar un sistema de guiat de llaç obert per a guiar-la des de el llançament fins que aquesta es separe dels coets de propulsió. Després de la separació, el sistema de guiat principal passa a ser un sistema de *guiat explícit* o *PEG*<sup>15</sup>, el funcionament del qual és una combinació del Sistema Q i dels atributs de correcció-predicció del sistema *Delta* original. Tot i que s'han fet moltes actualitzacions al sistema durant els últims 30 anys, com per exemple afegir un GPS al sistema, el nucli del sistema de guiat quasi no ha canviat.



Figura 59: Detall d'una de les parts de control del sistema de navegació inercial *Delco Carousel*

Òbviament, tots aquests conceptes també es van posar en pràctica en el camp de l'aviació comercial sent el millor exemple el sistema de navegació inercial *Delco Carousel*, el qual va permetre una automatització parcial de la navegació ja que, fins l'arribada dels sistemes de control de vol, permetien als pilots introduir en la computadora una serie de punts de pas, de manera que el sistema de navegació inercial, a partir de la posició i velocitat calculada, guiava l'avió fins el següent punt de pas. La comoditat d'aquest sistema va ser tal que *Boeing* el va utilitzar en els seus avions per a controlar el vol, encara que, per raons de fiabilitat, en configuracions triples.

### 4.3- Sistemes de Navegació Inercial bàsics

Com sabem, un sistema de navegació inercial és un sistema integrat que pot obtenir automàticament la velocitat, altitud i posició d'un objecte amb el fi de dirigir el seu rumb futur. Per a aconseguir-ho, el sistema utilitza una computadora i varies *unitats de mesurament inercial*, que no són altra cosa que els sensors dels que hem parlat en la secció sobre *els Sensors de moviment*.

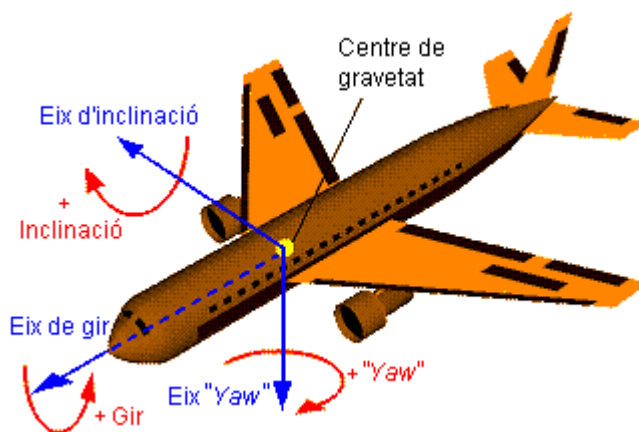


Figura 60: Representació de les diferents “velocitats angulars” segons l'eix



Per a determinar quina rotació està experimentant el vehicle en cada eix de l'espai s'utilitza un acceleròmetre angular. En canvi, per a mesurar les acceleracions no gravitacionals, s'utilitzarà un acceleròmetre lineal per cada eix que interesse controlar, pel que sí el nostre vehicle pot desplaçar-se en qualsevol de les tres dimensions, com per exemple l'avió de la figura 60, necessitarem tres sensors de cada tipus per a poder obtenir les dades necessàries per a determinar tots els moviments experimentats per l'objecte o vehicle.

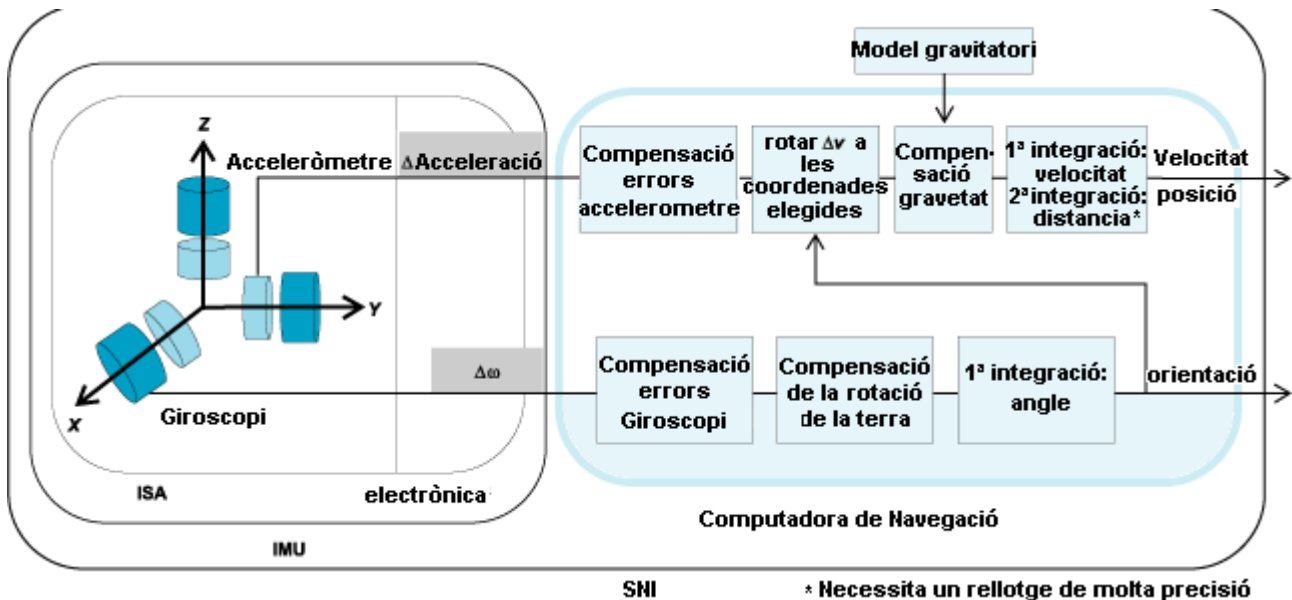


Figura 61: Diagrama de blocs bàsic d'un sistema de navegació inercial.

Una vegada obtingudes les dades d'acceleració i orientació, el processador calcula la posició actual del vehicle mitjançant diferents processos d'integració seqüencial: primer s'obté la velocitat actual integrant respecte al temps l'acceleració obtinguda en cada eix junt a una aproximació de la gravetat i, una vegada obtinguda aquesta, s'integra el resultat per a calcular la posició actual. Hem de tenir en compte que, tant el soroll en l'eixida dels sensors involucrats en el procés com la posterior integració d'aquesta van a provocar un error en el sistema que anirà acumulant-se a mesura que passe el temps pel que és prou habitual trobar-se en aquests sistemes una etapa de filtrat *Kalman*<sup>16</sup> i un receptor GPS per a corregir aquests errors i, de pas, millorar considerablement la precisió del sistema. A més, sí es tracta d'un sistema inercial que treballa prop de la superfície de la terra, aleshores també s'haurà de compensar l'error produït per la curvatura de la terra, pues sí no es té en compte aquesta, el centre de la plataforma del vehicle deixarà d'apuntar al centre de la terra a mesura que augmenti la distància recorreguda en el mateix sentit i direcció.

A l'hora de classificar els Sistemes de Navegació Inercial, degut a que generalment estan formats per giroscopis i acceleròmetres, el que es sol tenir en compte és el tipus de muntatge que s'utilitza per muntar els diferents sensors. Aleshores, segons el muntatge podem diferenciar entre:

- **Sistema Inercial Gimbalat:** tal i com s'observa en la figura 62, en aquests sistemes els sensors es col·loquen en una plataforma central articulada mecànicament mitjançant *gimbals*; un conjunt de tres anells cadascun dels quals disposa d'un parell de rodaments que, en repòs, formen un angle de 90° i permeten a la plataforma girar-se sobre qualsevol eix. Gràcies a açò, la plataforma manté l'orientació mentre el vehicle està rotant pel que podem mesurar directament qualsevol dels tres angles d'orientació sí mesurem la rotació experimentada pels rodaments i sí, a més, afegim un poc més d'electrònica, podem obtenir també els valors de l'acceleració lineal pues la direcció d'esta es manté constant.

<sup>16</sup> El filtrat *Kalman* és un algoritme que, a partir de les mesures obtingudes, produeix aproximacions de variables desconegudes amb major precisió que la que s'obtidria amb una única mesura.

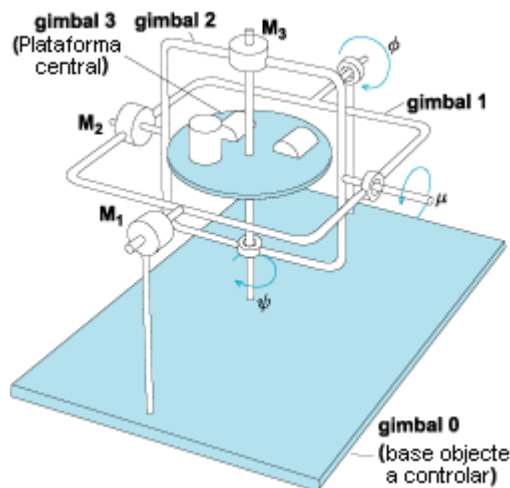


Figura 62: Esquema plataforma de 3 gimbals.

Degut a que aquests sistemes necessiten que la base de la plataforma estiga apuntant sempre en la mateixa direcció i, com el moviment de precessió d'un giroscopi tendix a desalinejar la plataforma, per a cancel·lar aquest efecte es sol afegir un segon giroscopi amb la mateixa inèrcia de rotació i velocitat de gir que el ja existent.

El principal problema d'aquest muntatge és la part mecànica del mateix pues el sistema pot deixar de funcionar sí un moviment bruscat arriba a bloquejar algun element mecànic del sistema, pel que per a evitar-ho, o bé es substitueixen aquests elements per cambres de fluids, o bé les maniobres es realitzen amb molt de compte.

- **Sistemes Strapdown:** Aquesta disposició rep aquest nom perquè els sensors estan directament “enganxats” al vehicle. Al no utilitzar parts mecàniques de precisió com els sistemes anteriors, el cost es veu reduït al mateix temps que augmenta la fiabilitat i el rang dinàmic del sistema. A canvi, no sols hem d'utilitzar més potència de càlcul per a conèixer la velocitat i posició, sinó que la precisió és menor.

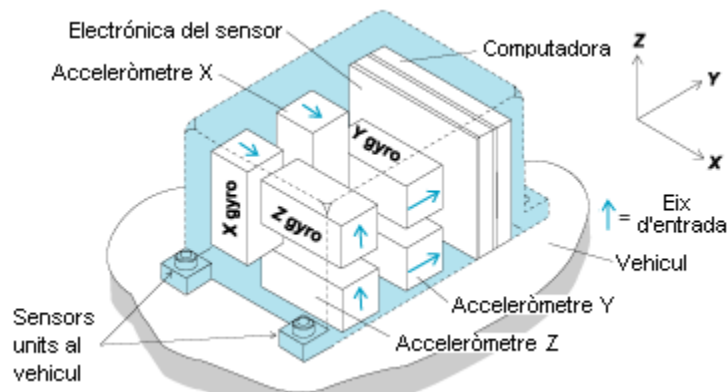


Figura 63: Disposició dels sensors en un SNI de tipus Strapdown

Independentment de com es munten els sensors en la plataforma, la precisió d'aquests sistemes depèn bàsicament de dos paràmetres: l'exactitud del rellotge utilitzat per a realitzar els processos d'integració i, sobretot, de la qualitat dels giroscopis utilitzats. De fet, la diferència de precisions entre els sistemes Strapdown i els Gimbaled es deu a que no utilitzen el mateix tipus de giroscopis doncs, la naturalesa dels sistemes Gimbaled s'adapta millor als giroscopis que directament mesuren l'angle<sup>17</sup> mentre que els giroscopis que hem vist en aquesta memòria, o el que és el mateix, els que mesuren la velocitat de gir<sup>18</sup>, encaixen millor en els sistemes Strapdown.

<sup>17</sup> Aquests giroscopis se'ls coneix amb el nom angles Integrating Gyros.

<sup>18</sup> En anglès, aquests giroscopis reben el nom de Rate Gyros.

## 5- Una possible aplicació d'un sistema de control d'estabilitat: Sensor d'ones.

### 5.1- Introducció: Per què mesurar ones?

Possiblement el lector, després de veure els exemples d'ús dels sensors d'inèrcia dels que hem parlat en aquesta memòria, haja pensat que un sistema que utilitze aquests sensors sols serveix per a conèixer la posició d'un objecte o facilitar el guiat d'este en l'espai, però la realitat és que un sistema de control d'estabilitat no té perquè utilitzar-se exclusivament per a tal fi, també podria utilitzar-se per a modelar una superfície desconeguda o com evoluciona esta al llarg del temps.

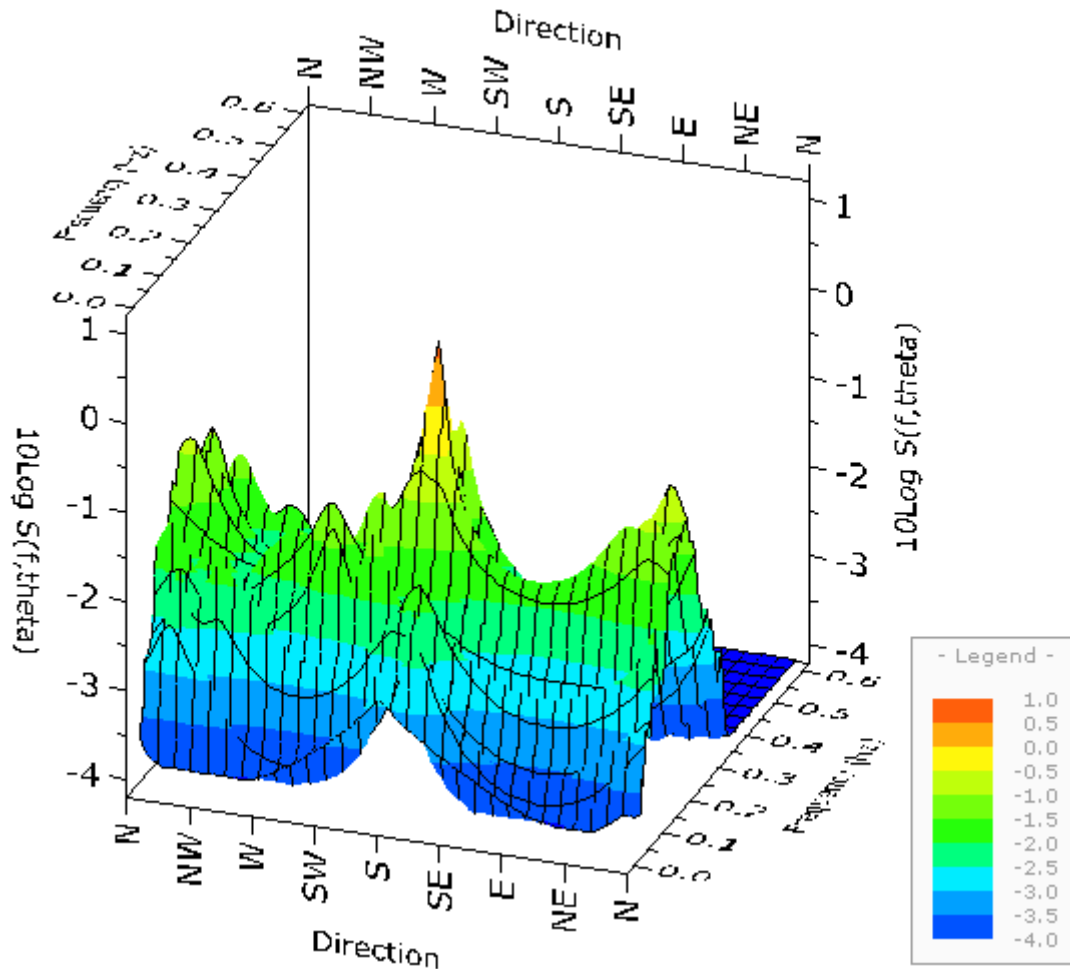


Figura 64: Representació 3D de la distribució de l'energia d'una ona respecte la seva freqüència i la direcció

Per què ens interessaria fer una cosa pareguda? Tot depèn de la superfície que volguérem estudiar. Sí, per exemple, es tractés de la superfície de l'oceà, podríem situar en zones estratègiques aquests sensors, capturar els moviments de la superfície i després, com l'estat de la mar pot considerar-se com una superposició d'un gran nombre d'ones de component sinusoidal de diferents freqüències i altures, utilitzant la transformada ràpida de *Fourier* podríem conèixer la distribució d'energia d'una ona segons la seva freqüència i, a partir d'aquesta informació, determinar com de tranquil·la està la mar en eixa zona, cosa que posteriorment ens podria ajudar a saber quina serà la força que deuriem poder aguantar els materials utilitzats en la construcció d'un moll en eixa zona, o simplement podríem limitar-nos a analitzar la informació capturada i calcular les probabilitats que ploiga, que es produísca una tempesta, un tifó, etc. pues l'oceà condiciona el clima degut a que, entre altres coses, transporta l'energia continguda en el principal element que les crea: el vent.

## 5.2- Ones: un dels elements inseparables de l'oceà

Les perturbacions de l'aigua, és a dir, les ones, estan íntimament lligades als oceans i és impossible trobar un oceà en el que no existisquen aquestes. Per això, com estan per tot arreu i transporten grans quantitats d'energia, interessa entendre les seves característiques i moviments pues les forces generades per aquestes són la principal causa de desgast de les estructures costeres o de la modelació de les costes i, a més, sí estes tenen unes dimensions considerables, poden suposar un perill per als vaixells o inclús per a les persones, pel que mesurar-les i analitzar-les pot arribar a ser més important del que pareix.



Figura 65: Ones trencant sobre un far.

Les ones són generades per les forces que pertorben l'aigua. La gran majoria d'estes són creades pel vent, però poden ser creades per una gran varietat de forces: des de l'atracció gravitatòria de la lluna, passant per terratrèmols subterranis fins a arribar a forces aparentment tan insignificants com pot ser el moviment d'un peix.

En l'oceà, les molècules de l'aire espanten l'aigua sense problemes ja que el vent bufa sobre una superfície "suau". Esta fricció entre l'aire i l'aigua empenya les petites ondulacions que puguen existir sobre la superfície de l'oceà pel que a mesura que continue bufant el vent, les ondulacions aniran augmentant de grandària podent arribar a convertir-se en ones d'uns quants metres d'alçada. Però que una ona siga més gran o més menuda no depèn exclusivament del temps que està bufant el vent, depèn també de la velocitat del mateix i de la distància que porta bufant sense canviar de direcció, de manera que, quan més elevats siguen els valors d'estos paràmetres, més gran serà l'ona. Òbviament, el creixement d'una ona creada pel vent no és infinit ja que en algun moment l'ona trencarà i, per tant, es dissiparà l'energia impresa pel vent en les partícules d'aigua que la formen.

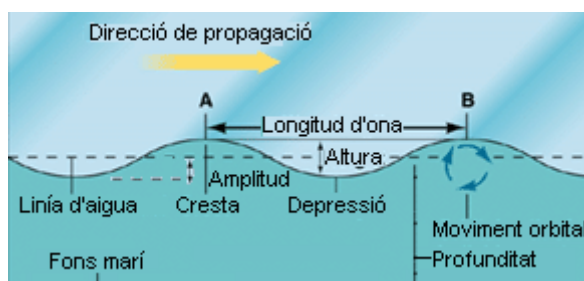


Figura 66: ona superficial ideal

Sí ens fixem en l'aigua, una ona pot parèixer un enorme paret desplaçant-se al llarg de la superfície del oceà, però en realitat, el que es desplaça, és la perturbació junt a la seva energia ja que l'aigua simplement es balanceja amunt i avall de manera que descriu un patró circular com el que es pot veure en la figura 66. Eixa mateixa figura ens ajudarà a entendre els moviments i comportaments de

les ones pues, al principi, per a simplificar, descriurem les ones com una única funció simple d'un sinus o cosinus. Segons això i com podem observar en la citada figura, una ona superficial ideal pot ser descrita mitjançant els següents conceptes o paràmetres:

- **Línia d'aigua:** El nivell de la superfície del oceà si aquest estigués totalment pla i calmat.
- **Cresta:** És la part més alta sobre la línia d'aigua.
- **Depressió:** És la part més baixa sobre la línia d'aigua.
- **Altura:** La distància vertical entre la cresta i la depressió.
- **Longitud d'ona:** La distància horitzontal entre successives crestes o depressions.
- **Freqüència:** El nombre d'ones que passen per un punt en particular en un període de temps concret.
- **Amplitud:** La mitat de l'altura de la ona o la distància entre la línia d'aigua i la cresta o la depressió.
- **Profunditat:** Distància des del fons de l'oceà fins la línia d'aigua.
- **Direcció de propagació:** La direcció en la que una ona està viatjant.

El moviment i comportament d'una ona sinusoidal simple pot ser descrit totalment quan es coneix la longitud d'ona ( $L$ ), l'altura ( $H$ ), el període ( $T$ ) i la profunditat ( $D$ ). Per exemple, a aigües profundes<sup>19</sup>, la velocitat de l'ona pot ser calculada a partir de la mida de la mateixa. En canvi a aigües poc profundes la velocitat depèn principalment de la profunditat de l'aigua.

De la mateixa manera, l'altura de l'ona està limitada tant per la profunditat com per la longitud d'ona. Açò vol dir que, per a una profunditat i període d'ona donats, hi ha una altura màxima per dalt de la qual l'ona es torna inestable i es trenca. En aigües profundes este valor és funció de la longitud d'ona mentre que a aigües poc profundes depèn tant de la profunditat com de la longitud d'ona.

### 5.3- Tipus d'ones

Com és d'esperar, les ones d'un oceà no tenen una forma sinusoidal i rarament tenen una única longitud d'ona o període. De fet, la superfície d'un oceà és irregular i està formada per la combinació de moltes ones component de diferent període i direcció. Estes ones poden classificar-se com a mínim de quatre formes diferents:

- **Profunditat de l'aigua:** Segons aquesta classificació, si la profunditat de l'aigua és major que mitja longitud d'ona ( $D > L/2$ ) estarem parlant d'ones d'aigua profunda, mentre que si  $D < L/20$  és consideraran d'aigua poc profunda. La principal diferència entre estes dos és que en les ones de poca profunditat, la velocitat de l'ona depèn exclusivament de la profunditat: si és una ona d'aigua poc profunda la velocitat disminuirà si disminueix la profunditat, i si és d'aigua profunda, la velocitat serà major quan més longitud d'ona tinga l'ona.

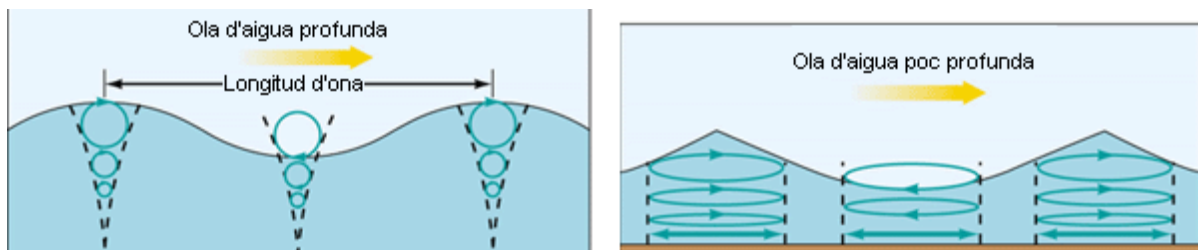


Figura 67: El diferent moviment de les partícules d'aigua d'ones de diferent profunditat modifica les dependències de la seva velocitat

- **Mètode de generació:** Atenent a la força que les origina, podem parlar d'ones d'impacte si han sigut generades per un terratrèmol o qualsevol altra forma "d'impacte" mentre que parlarem d'ones de vent quan estes hagen sigut creades per la força del vent.

<sup>19</sup> Es considera aigua profunda quan la profunditat és major que la mitat de la longitud d'ona.

- **Període de les ones:** Sí ens centrem en el període de l'ona per a classificar-les podem diferenciar entre:
  - **Ones Capil·lars:** Tenen un període inferior a 0.1 segons.
  - **Ones Gravitatòries:** Són les més comuns i tenen un període que oscil·la entre 1 i 30 segons. Solen ser creades pels vents o les tempestes i la seva energia es centra en el període de 10 segons.
  - **Ones Llargues:** Aquestes són generades per tempestes i per terratrèmols. El seu període és superior als 5 minuts.
  - **Marees:** Els períodes d'aquestes són de 12 i 24 hores doncs les generen el sol i la lluna.

• **Relació amb la força que les origina:** Esta classificació distingeix entre dos possibilitats: ones forçades sí estes necessiten a la força que les genera per a continuar actives o ones lliures sí es mouen lliurement després que la força que les genera desaparega, com és el cas de les ones d'impacte.

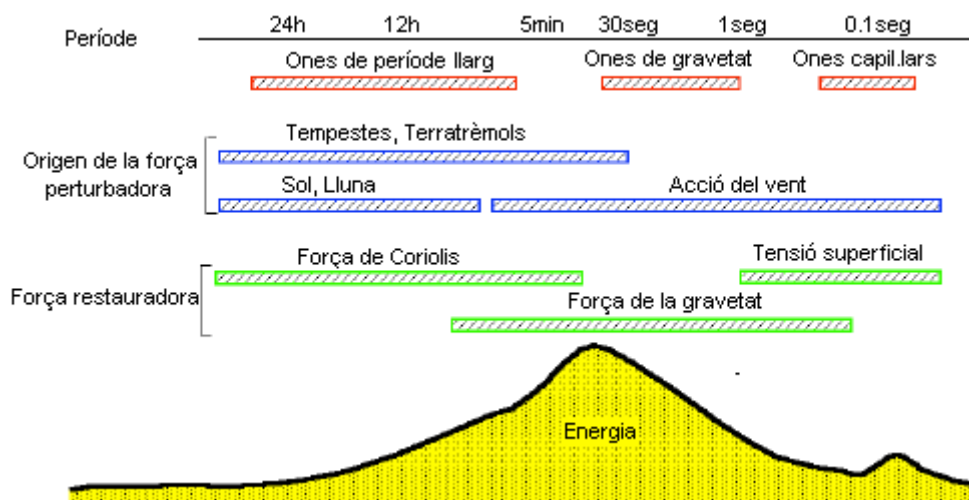


Figura 68: Tipus d'ones i l'energia continguda en cadascuna.

Tot i que la quantitat de tipus d'ones no és molt elevat, nosaltres ens centrarem exclusivament en les ones més comuns: les ones de vent. Com ja sabem, quan el vent bufa sobre l'oceà es transferix part de la seva energia a l'aigua i es generen ones de superfície confuses i sense una direcció definida, pel que sí estiguérem en un vaixell a la deriva intentant calcular el període d'una ona mesurant el temps que tarden en passar dos crestes successives per un mateix punt, acabariem donant-ho per impossible ja que segurament no trobaríem cap cresta que no desapareguera de sobte per a tornar a aparèixer un poc més lluny, al cap i a la fi, les ones es combinen entre elles mitjançant un procés d'interferència com el de la figura 69. Aquest procés d'interferència acabarà creant el que es coneix amb el nom de *mar estesa*.

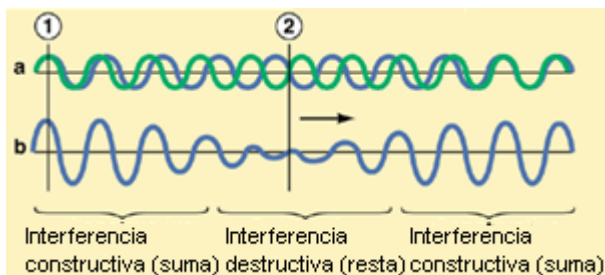


Figura 69: Procés d'interferència en una ona

Quan les ones són creades pels forts vents d'una tempesta, la superfície de l'oceà sol presentar un aspecte caòtic amb moltes ones curtes abruptes de diferents altures. En canvi, en les zones tranquil·les lliures de vents forts, les ones solen ser llargues i amb un aspecte més arrodonit i

uniforme. Aquest fet fa que es diferencie entre dos tipus d'ones superficials: les *Seas* i les *Swells*. Les primeres es referixen a les ones de curt període d'aspecte irregular que, o bé encara les està creant el vent, o bé continuen molt prop de la zona en la que van ser creades. Les de tipus *Swell*, per contra, es refereixen a ones d'aspecte més regular i de període llarg les quals estan lluny de la influència dels vents que les van produir per haver-se'n eixit de la zona en la qual van ser creades. Com estes últimes tenen major longitud d'ona, contenen més energia i, per tant, com tarden més en dissipar-la, poden viatjar més lluny i ràpid que les de tipus *Sea*.

La grandària de les ones creades pel vent, tal i com hem comentat anteriorment, depèn de la profunditat de l'aigua, de la velocitat i del temps i distància que porta bufant el vent. Per a una velocitat constant del vent, sí la profunditat de l'aigua i la distància des de la que porta bufant el vent és el suficientment llarga, les ones portaran el mar al estat anomenat *totalment desenvolupat*. Una vegada en aquest estat, les ones no creixeran més i, tant la seva alçària, com la seva longitud d'ona, seran màximes. Sí el vent no para de bufar, continuarà subministrant energia a la superfície del mar però, com que les ones no poden créixer més, esta energia es dissiparà quan les ones trenquen, és a dir, quan la divisió entre l'altura de l'ona i la seva longitud d'ona siga major a  $0,14^{20}$ . Esta dissipació d'energia dona com a resultat les *whitecaps*: l'espuma blanca que apareix quan les ones trenquen.

| Velocitat (Km/h) | Fetch <sup>21</sup> (Km) | Duració (h) | Altura (m) | Longitud (m) | Període (seg) |
|------------------|--------------------------|-------------|------------|--------------|---------------|
| 20               | 24                       | 2,75        | 0,33       | 10,6         | 3,2           |
| 40               | 176                      | 11,5        | 0,8        | 39,7         | 6,2           |
| 60               | 660                      | 27,5        | 5,1        | 89,2         | 9,1           |
| 80               | 1682                     | 50          | 10,3       | 158,6        | 12,4          |

Taula 11: Exemples de valors necessaris per a obtenir un oceà *totalment desenvolupat*.

En la taula 11 podem veure com, a mesura que la velocitat del vent augmente, serà necessari que el vent bufe durant més temps i durant una distància major per a que els valors de l'altura de l'ona i de la seva longitud augmenten el suficient com per a arribar a l'estat de *mar totalment desenvolupat*, per aquest motiu es diu que, tant la distància durant la que ha de bufar el vent, com el temps que aquest ha d'estar actuant, limiten el *total desenvolupament* del mar.

Una altra conclusió que es pot treure d'aquesta taula és que, a mesura que augmenta la velocitat del vent, també augmentarà la grandària del mar, és a dir, l'altura i període de les diferents ones que el formen, pel que també es pot afirmar que un mar, el qual s'ha desenvolupat totalment a una determinada velocitat del vent, deixa d'estar totalment desenvolupat sí la velocitat del vent augmenta, pues aquest increment de velocitat necessitarà anar acompanyat també d'un augment en el temps i distància d'actuació del vent per a que aquest pugui produir ones de major grandària i, per tant, aconseguir novament un mar *totalment desenvolupat*.

#### **5.4- Matemàtiques dels oceans: Equacions de moviment de les ones**

Les ones, com ja hem vist, són el resultat d'aplicar sobre l'oceà alguna o algunes de les següents forces:

- **Gravitatòria:** és la força predominant i, quan el sol o la lluna es mouen respecte a la terra, produeix la marea, les corrents de marea i les barreges d'aquestes quan estan dins de l'oceà. Dins d'aquesta destaquen els gradients de pressió horitzontal, és a dir, la diferència de pes que existeix entre les aigües de diferents regions de l'oceà i la flotabilitat, la qual produeix un

<sup>20</sup>  $H/L > 1/7$ , aquest valor també es coneix com el *criteri de trencat*.

<sup>21</sup> El paràmetre *Fetch* equival a la distància que ha recorregut el vent des que va començar a bufar.

desplaçament cap amunt o cap a baix d'una parcel·la d'aigua que és més o menys densa que l'aigua que està al seu nivell. Un exemple d'açò és l'efecte que es produeix quan bufa un vent fred sobre el mar. Quan açò passa, la densitat de l'aigua superficial augmenta i fa que aquesta “enfonse” l'aigua que té per baix.

- **Fricció:** és la força que actua en un cos quan este s'està desplaçant sobre un altre cos amb el que està en contacte. En el cas de l'aigua pot ser l'aire o les parcel·les d'aigua i el principal exponent d'esta força és “l'esforç del vent” sobre la superfície de l'aigua. Esta força és la que transfereix un moment horitzontal al mar i crea les corrents. A més, també és la responsable de l'augment de la grandària de les ones pues, al bufar el vent sobre les ones existents en la superfície del mar, es produeix un desequilibri de pressió en estes el qual transferirà energia a les mateixes.

- **Pseudo-forces:** aquestes “forces” són producte del moviment en sistemes de coordenades curvilinis o rotatoris pel que és normal veure els seus termes formant part de les equacions de moviment inercial d'aquests sistemes. El principal exemple d'estes pseudo-forces és la base de funcionament dels giroscopis: la força *Coriolis*.

A part d'estes forces n'hi han altres, com per exemple la sísmica o l'exercida per la pressió atmosfèrica, que també poden originar ones, però comparades amb les anteriors, la seva importància és menor. Siga com siga, independentment d'açò, les ones acaben combinant-se i desplaçant-se en grups entre els quals destaquen els següents:

- **Ones planetàries:** en estes ones, la força de restauració és la rotació de la terra.
- **Ones de superfície o gravitatòries:** són les ones que trenquen en la platja i la seva força de restauració es deu a la diferència entre la densitat de l'aire i la de l'aigua de la superfície de l'oceà.
- **Ones internes:** tenen cert paregut amb les ones de superfície però la seva força de restauració depèn dels canvis de la densitat amb la profunditat.
- **Corrents de marea:** estes corrents horitzontals són producte de les ones internes creades per les mareas.
- **Ones de vora:** sols s'aprecien en zones poc profundes, com la costa, pues la seva amplitud disminueix exponencialment amb la profunditat pel que, al tenir períodes d'uns poc minuts, desapareixen ràpidament.

Com ara veurem, el moviment de les ones pot expressar-se mitjançant equacions matemàtiques. Per a fer-ho, primer dividirem el flux de l'aigua en petites “caixes” com la de la figura 70 i després, utilitzant alguna de les propietats de la “caixa”, com el principi de conservació de la massa, derivarem les equacions diferencials que governen el flux de fluids.

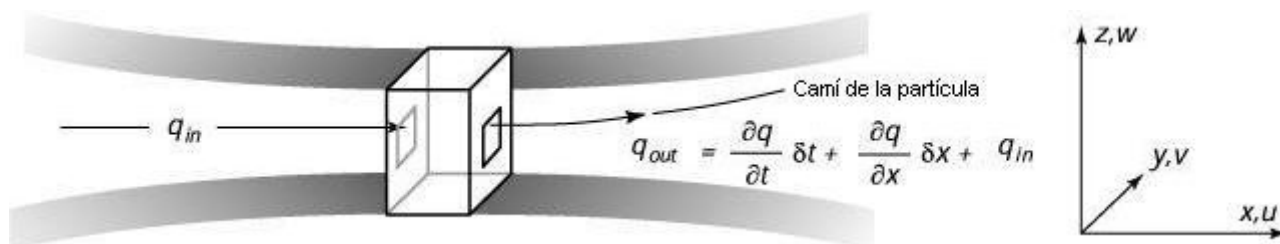


Figura 70: “Caixa exemple” utilitzada per a calcular els fluxos de l'aigua



Quan calcem l'acceleració del flux de la caixa obtindrem una equació que relaciona l'acceleració d'una partícula  $Du/Dt$  amb la velocitat en un punt en concret del fluid. Una vegada obtinguda aquesta equació, també coneguda com *derivada total*, podrem utilitzar-la per a derivar les equacions de dinàmica de fluids proporcionades per la segona llei de *Newton*. Si assumim que  $q_{in}$  és la quantitat de flux d'entrada mentre que  $q_{out}$  és el d'eixida i que  $q$  pot variar contínuament tant en el temps com en l'espai, aleshores la relació entre  $q_{in}$  i  $q_{out}$  és:

$$q_{out} = q_{in} + \frac{\partial q}{\partial t} \delta t + \frac{\partial q}{\partial x} \delta x$$

mentre que la velocitat de variació de  $q$  respecte el volum val:

$$\frac{Dq}{Dt} = \frac{q_{out} - q_{in}}{\delta t} = \frac{\partial q}{\partial t} + \frac{\partial q}{\partial x} \frac{\delta x}{\delta t}$$

com  $\delta x/\delta t$  és la velocitat  $u$ , l'equació anterior es pot reescriure com:

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + u \frac{\partial q}{\partial x}$$

i sí tenim en compte les tres dimensions, la derivada total equival a:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z} ; \quad \frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla()$$

on  $u$  és el vector velocitat i  $\nabla$  és el gradient del flux. Aquest resultat és interessant pues implica que canviar d'un sistema de coordenades referit a una partícula a un altre sistema referit a l'espai tridimensional, transforma una derivada lineal a una no lineal.

Gràcies a aquest resultat podrem calcular el canvi de moment en una parcel·la de fluid. Com indica la segona llei de *Newton*, la relació entre el canvi de moment d'un fluid i la força que se li aplica és:

$$\frac{D(m\mathbf{v})}{Dt} = \mathbf{F}$$

on  $F$  és la força,  $m$  és la massa i  $v$  és la velocitat. Com estem calculant la força en una partícula hem d'utilitzar la derivada total pel que sí assumim que la massa és constant, l'equació anterior pren la següent forma:

$$\frac{D\mathbf{v}}{Dt} = \frac{\mathbf{F}}{m} = \mathbf{f}_m$$

sent  $f_m$  la força per unitat de massa. Si ara tenim en compte les principals forces responsables de la generació d'ones, és a dir, afegim a l'equació els termes de la força *Coriolis*, la de la fricció, gravetat i els gradients de pressió, podem escriure la següent expressió:

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho} \nabla p - 2\boldsymbol{\Omega} \times \mathbf{v} + \mathbf{g} + \mathbf{F}_r$$

és a dir, l'acceleració és igual a la suma de les forces de la gravetat  $g$  i fricció  $F_r$ , menys el gradient de pressió i la força *Coriolis*, on  $\Omega$  és la velocitat de rotació de la terra i val  $7.292 \times 10^{-5}$  rad/seg.

Desenvolupant l'expressió d'acceleració anterior i expressant les seues components en forma de sistema de coordenades cartesià obtenim l'equació del moment:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + 2\Omega v \sin \varphi + F_x \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} - 2\Omega u \sin \varphi + F_y \\ \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial z} + 2\Omega u \cos \varphi - g + F_z \end{aligned}$$

on  $F_i$  són els components de qualsevol força per unitat de massa i  $\varphi$  és la latitud. A més, s'ha suposat que  $w \ll v$  pel que el terme  $2\Omega w \cos \varphi$  s'ha eliminat de la primera equació. Aquesta equació unes vegades rep el nom de l'equació d'Euler i altres el de l'equació de Navier-Stokes i, quan s'utilitza en dinàmica d'oceans, el terme  $2\Omega u \cos \varphi$  pot ser ignorat sempre que no es tracte d'un estudi de la gravetat utilitzant gravímetres instal·lats en vaixells.

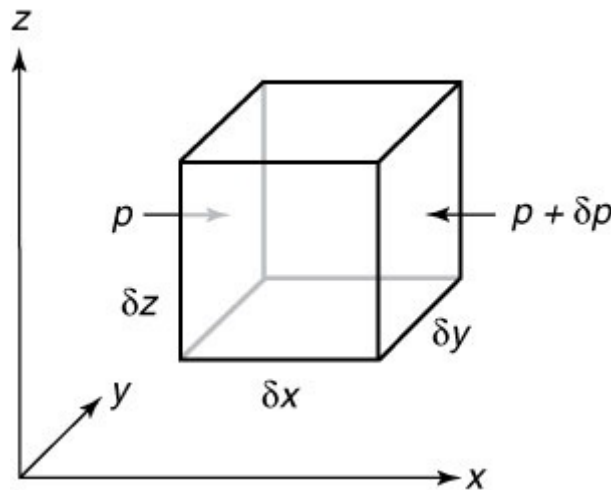


Figura 71: Diagrama del flux utilitzat per a derivar la pressió en l'equació del moment.

Per a derivar la pressió anem a suposar que sobre els laterals del cub de la figura 71 actuen les forces especificades en la mateixa. Així doncs, la força neta  $F_x$  en l'eix X és:

$$\delta F_x = p \delta y \delta z - (p + \delta p) \delta y \delta z, \delta F_x = -\delta p \delta y \delta z$$

com:

$$\delta p = \frac{\partial p}{\partial x} \delta x$$

al substituir tenim:

$$\delta F_x = -\frac{\partial p}{\partial x} \delta x \delta y \delta z \quad ; \quad \delta F_x = -\frac{\partial p}{\partial x} \delta V$$

Al dividir l'equació anterior per la massa  $\delta m$  del fluid de l'interior de la caixa, l'acceleració del fluid en la direcció de X és:

$$a_x = \frac{\delta F_x}{\delta m} = -\frac{\partial p}{\partial x} \frac{\delta V}{\delta m} ; a_x = -\frac{1}{\rho} \frac{\partial p}{\partial x}$$

com és d'esperar, les forces de la pressió i l'acceleració deguda a estes en els altres eixos es deriva de la mateixa manera.

El procés de derivació del terme de la força *Coriolis* en l'equació del moment no és gens simple pel que directament conclourem que l'acceleració d'una parcel·la de fluid en un sistema rotatori s'expressa amb l'equació:

$$a_{\text{fixe}} = \left( \frac{D\mathbf{v}}{Dt} \right)_{\text{fixe}} = \left( \frac{D\mathbf{v}}{Dt} \right)_{\text{rotatori}} + (2\boldsymbol{\Omega} \times \mathbf{v}) + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{R})$$

on  $R$  és el vector distància des del centre de la terra,  $\boldsymbol{\Omega}$  és el vector de la velocitat angular de la terra i  $\mathbf{v}$  és la velocitat de la parcel·la de líquid en un sistema de coordenades referit a la terra. El terme  $2\boldsymbol{\Omega} \times \mathbf{v}$  és la força *Coriolis* i, per últim, el terme  $\boldsymbol{\Omega} \times \boldsymbol{\Omega} \times R$  equival a l'acceleració centrífuga.

Per a derivar el terme de la força de la gravetat tenim que l'atracció entre dos masses  $M_1$  i  $m$  és igual a:

$$\mathbf{F}_g = \frac{G M_1 m}{R^2}$$

on  $R$  és la distància entre les masses,  $G$  és la constant gravitacional i el vector força  $F_g$  està dirigit al llarg de la línia que uneix entre si les dos masses. Aleshores, la força per unitat de massa deguda a la gravetat equival a:

$$\frac{\mathbf{F}_g}{m} = \mathbf{g}_f = \frac{G M_E}{R^2}$$

on  $M_e$  és la massa de la terra. Afegint l'acceleració centrífuga a l'expressió anterior obtenim el valor  $g$  de la figura 72:  $g = g_f - \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times R)$

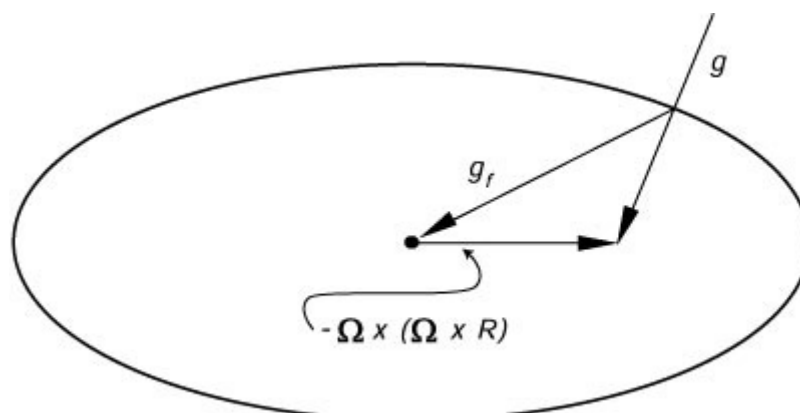


Figura 72: representació de la gravetat  $g$  de l'equació del moment

Com es dedueix de l'equació anterior i s'observa en la figura 72, l'acceleració cap a baix  $g$  d'un cos en repòs sobre la superfície de la terra és la suma de l'acceleració gravitacional entre el cos i la massa de la terra  $g_f$  i les acceleracions centrífugues produïdes per la rotació de la terra  $\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times R)$ .

Destacar que la gravetat no apunta al centre de massa de la terra per efecte de l'acceleració centrífuga. Com a resultat, la terra no és esfèrica sinó el·lipsoide, per això s'ha exagerat la forma de l'el·lipsoide de la figura 72.

Per a derivar l'equació de la conservació de la massa en un fluid assumirem que el flux entra i ix del sistema tal i com mostra la figura 73.

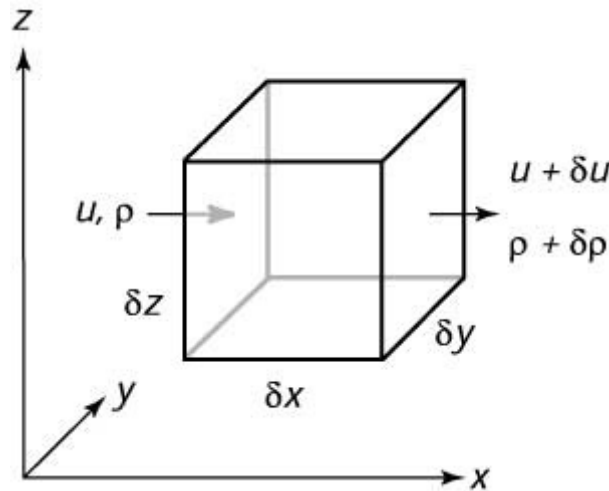


Figura 73: el cub utilitzat per a derivar l'equació de la conservació de la massa en un fluid

el valor del flux d'entrada i el d'eixida val:

$$\begin{aligned} \text{Entrada de flux a la massa} &= \rho u \delta z \delta y \\ \text{Eixida de flux de la massa} &= \left( \rho + \frac{\partial \rho}{\partial x} \delta x \right) \left( u + \frac{\partial u}{\partial x} \delta x \right) \delta z \delta y = \left( \rho u + \rho \frac{\partial u}{\partial x} + u \frac{\partial \rho}{\partial x} + \frac{\partial \rho}{\partial x} \frac{\partial u}{\partial x} \delta x \right) \delta x \delta y \delta z \end{aligned}$$

pel que el flux en l'interior del cub deu ser la resta dels dos:

$$\text{Flux de masa} = \left( \rho \frac{\partial u}{\partial x} + u \frac{\partial \rho}{\partial x} + \frac{\partial \rho}{\partial x} \frac{\partial u}{\partial x} \delta x \right) \delta x \delta y \delta z$$

sí  $\delta x$  s'aproxima a 0, podem dir:

$$\text{Flux de masa} = \frac{\partial(\rho u)}{\partial x} \delta x \delta y \delta z$$

i a l'incloure les tres dimensions:

$$\text{Flux de masa} = \left( \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} \right) \delta x \delta y \delta z$$

el flux de massa deu equilibrar-se amb un canvi de massa

$$\frac{\partial \rho}{\partial t} \delta x \delta y \delta z$$

a més, la conservació de la massa exigeix:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0$$

aquesta equació, coneguda com l'equació de continuïtat, es pot escriure de la següent manera al desenvolupar les derivades parcials:

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} + \rho \frac{\partial u}{\partial x} + \rho \frac{\partial v}{\partial y} + \rho \frac{\partial w}{\partial z} = 0$$

els quatre primer termes es corresponen amb la derivada total de la densitat pel que finalment l'equació de continuïtat per flux compressible es pot expressar també com:

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

La densitat en l'oceà és quasi sempre constant pel que per a simplificar podem suposar que sempre és constant si es compleixen les següents condicions:

- Les velocitats en l'oceà han de ser menyspreable comparades amb la velocitat del so  $c$ . D'aquesta manera ens assegurem que la velocitat no canvia la densitat, doncs a mesura que la velocitat s'aproxima a la velocitat del so, esta produeix *ones d'impacte*.
- El desfasament entre ones deu ser menyspreable comparat amb la velocitat del so  $c$ . Este requisit és necessari quan es té en compte els efectes del so en l'oceà pues implica que el fluid és compressible, cosa que no passa sí parlem de fluids incompressibles doncs en aquests, la velocitat del so és infinita.
- El moviment vertical deu ser menyspreable comparat amb  $c^2/g$ , on  $g$  és la gravetat i  $c$  la velocitat del so. Açò ens permet assegurar-nos que a mesura que augmente la pressió amb la profunditat de l'oceà, els canvis produïts en la densitat per la pressió seran petits.

Aquesta aproximació, coneguda amb el nom d'*aproximació de Boussinesq*, és certa per a fluxos oceànics i ens permet afirmar que les aigües del mar són incompressibles i simplificar l'equació de continuïtat.

Per a veure de quina manera ens ajuda l'*aproximació de Boussinesq*, anem a definir el coeficient de compressibilitat  $\beta$  de manera que:

$$\beta \equiv -\frac{1}{V} \frac{\partial V}{\partial p} = -\frac{1}{V} \frac{dV}{dt} / \frac{dp}{dt}$$

on  $V$  és el volum i  $p$  la pressió. Com en medis incompressibles  $\beta=0$  tenim:

$$\frac{1}{V} \frac{dV}{dt} = 0$$

com  $dp/dt \neq 0$ , al suposar que la massa és constant i que la densitat és igual a massa per unitat volum:

$$\frac{1}{V} \frac{dV}{dt} = -V \frac{d}{dt} \left( \frac{1}{V} \right) = -\frac{V}{m} \frac{d}{dt} \left( \frac{m}{V} \right) = -\frac{1}{\rho} \frac{d\rho}{dt} = -\frac{1}{\rho} \frac{D\rho}{Dt} = 0$$

pel que l'equació de continuïtat per a un líquid compressible queda:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

expressió coneguda com l'equació de continuïtat per a fluids incompressibles. Esta última equació junt a les equacions del moment ens permeten calcular idealment el comportament de les corrents.

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + 2\Omega v \sin \varphi + F_x \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} - 2\Omega u \sin \varphi + F_y \\ \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial z} + 2\Omega u \cos \varphi - g + F_z \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} &= 0 \end{aligned}$$

Intentar resoldre les 4 equacions anteriors, és a dir, les equacions del moment i l'equació de continuïtat per a fluids, és complicat ja que tot i que tenim un sistema de 4 equacions amb 4 incògnites desconegudes,  $u$ ,  $v$ ,  $w$ ,  $p$ , es tracta de 4 equacions diferencials parcials no lineals. Això vol dir que, inclús utilitzant les condicions de contorn habituals de la mecànica de fluids, calcular la solució és molt complicat i això que en cap moment hem tingut en compte els efectes de la vorticitat<sup>22</sup> o la viscositat del fluid. Aquesta complexitat és el motiu pel qual s'intenta simplificar aquestes equacions o utilitzar altres mètodes de modelació de les corrents i, tot i això, continua sense ser fàcil obtenir una resposta numèrica.

## 5.5- Teoria lineal d'ones de superfície de l'oceà

Sí mirem el mar des de la costa podem veure ones en la superfície del mar. Mirant amb més calma podem veure que les ones són ondulacions de la superfície de la mar les quals tenen una altura que pot arribar al metre i una longitud d'ona inferior als 100 metres. Dedicant més temps a l'observació de les mateixes ens n'adonarem que ni la seva altura ni longitud són constants pues unes vegades les ones són el resultat del vent, altres són el resultat de tempestes llunyanes o inclús poden ser producte del moviment de la lluna respecte la terra. Aquesta falta de regularitat és la responsable que les equacions de moviment de les ones siguin tan complexes i que es busquen altres mètodes o teories que expliquen i representen el comportament de les ones. Una d'estes teories és la teoria lineal d'ones.

Tal i com dèiem, el fet que les ones siguin inherentment no lineals complica la resolució de les equacions de moviment, pues sí les condicions de contorn de les equacions són les ones a mesurar i aquestes són no lineals, com les podem solucionar?

Per a començar, suposarem que l'amplitud de les ones sobre la superfície del mar és infinitament menuda de manera que es pugui afirmar que la superfície és totalment plana. A més, per a simplificar matemàticament les equacions, també suposarem que tant la força *Coriolis* com la viscositat són

<sup>22</sup> Rotació d'un fluid

menyspreables i que el flux viatja en la direcció del eix  $x$  i és bidimensional. Si mantenim la rotació, obtindrem ones *Kelvin*<sup>23</sup>, pel que l'elevació  $z$  d'una ona viatjant en la direcció  $x$  és:

$$z = a \sin(kx - \omega t)$$

amb  $\omega = 2\pi f = (2\pi)/T$  i  $k = (2\pi)/L$ . Si, com hem dit,  $ka = 0$ , aleshores  $\omega$  és la freqüència en radians per segon,  $f$  és la freqüència de l'ona en Hertz (Hz),  $k$  és el *nombre d'ona*,  $T$  és el període de l'ona i  $L$  la longitud d'ona de la mateixa.

### 5.5.1- Relació de dispersió

La freqüència d'ona  $\omega$  està relacionada amb el nombre d'ona  $k$  per la *relació de dispersió*:

$$\omega^2 = gk \tanh(kd)$$

on  $d$  és la profunditat de l'aigua i  $g$  és l'acceleració de la gravetat. Com moltes propietats de les ones es poden mesurar amb precisions d'un 5-10%, podem recórrer a dos aproximacions que ens permeten calcular aquesta relació amb una precisió d'un 10%:

- L'aproximació d'aigües profundes es pot utilitzar si la profunditat de l'aigua  $d$  és molt més gran que la longitud d'ona  $L$ . En aquest cas  $d \gg L$ ,  $kd \gg 1$  i  $\tanh(kd) = 1$
- Es pot aplicar l'aproximació d'aigües poc profundes si la profunditat de l'aigua és molt més menuda que la longitud d'ona. En aquest cas  $d \ll L$ ,  $kd \ll 1$  i  $\tanh(kd) = kd$ .

Per a aquests dos límits de profunditat d'aigua, la *relació de dispersió* es redueix a:

$$\omega^2 = gk \quad \text{Relació de dispersió a aigües profundes.}$$

$$d > L / 4$$

$$\omega^2 = gk^2 d \quad \text{Relació de dispersió a aigües poc profundes.}$$

$$d < L / 11$$

### 5.5.2- Velocitat de fase

La velocitat de fase  $c$  és la velocitat a la qual una fase particular de l'ona es propaga, per exemple, a la velocitat de propagació de la cresta de l'ona. És a dir, si en un període d'ona  $T$  la cresta avança una longitud d'ona  $L$ , la velocitat de fase serà  $c = L / T = \omega / k$ .

La direcció de propagació és perpendicular a la cresta de l'ona i cap a la direcció  $x$  positiva. Si apliquem les aproximacions d'aigua profunda i poc profunda per a la dispersió tenim que, dins de la precisió del 5% especificat en la secció sobre la relació de dispersió, la velocitat de fase és:

- $c = \sqrt{g/k} = g/\omega$ , a aigües profundes
- $c = \sqrt{gd}$ , a aigües poc profundes

<sup>23</sup> Les ones Kelvin són ones que apareixen en l'oceà a mode de reacció per un canvi brusca en la seva superfície. La seva força restauradora és la gravetat i es desplacen a una velocitat de grup  $c = \sqrt{gH(\rho_1 - \rho_2)/\rho_1}$ , on  $\rho_n$  és la densitat de les capes,  $H$  l'altura de les ones i  $g$  la força de la gravetat

A aigües profundes, la velocitat de fase depèn de la longitud d'ona o de la freqüència de l'ona pel que les ones més llargues són les més ràpides. A aigües poc profundes la velocitat de fase depèn exclusivament de la profunditat  $i$ , al contrari que passa en les ones d'aigua profunda, estarem parlant d'ones no dispersives.

### 5.5.3- Velocitat de grup

El concepte de velocitat de grup  $c_g$  és fonamental per a entendre la propagació d'ones lineals i no lineals, doncs no sols indica la velocitat a la qual un grup d'ones es desplaça per l'oceà sinó que a més ens proporciona informació sobre la velocitat de propagació de l'energia de les ones:

$$c_g \equiv \partial\omega / \partial k$$

utilitzant les aproximacions per a aigües profundes i poc profundes  $c_g$  val:

- $c_g = g / 2\omega = c / 2$ , a aigües profundes
- $c_g = \sqrt{g d} = c$ , a aigües poc profundes

En el cas de les ones de superfície, la direcció de propagació és perpendicular a les crestes de l'ona en el sentit positiu de l'eix  $x$ . Si es tracta d'un cas més general, com el de les ones *Kelvin*, la velocitat de grup no té perquè ser necessàriament perpendicular a la de les crestes de l'ona.

D'aquest resultat es pot deduir que un grup d'ones d'aigua profunda es desplaça a la mitat de la velocitat de fase de les ones que formen el grup. Com pot ser açò? Sí poguérem mirar de prop un grup d'ones travessant l'oceà, al principi veuríem aparèixer unes crestes d'ona al final del tren les quals anirien desplaçant-se al llarg del tren per a, finalment, desaparèixer al principi del mateix pues cada cresta d'ona es mou al doble de la velocitat del grup.

Vol dir açò que les ones es desplacen en grups regits per la relació de dispersió? Sí. *Munk* i uns col·legues van realitzar en l'any 1963 una serie d'experiments els quals no sols afirmaven que les ones que poden desplaçar-se una gran distància són dispersives, sinó que també demostraven que la dispersió pot utilitzar-se per a rastrejar tempestes. Per a demostrar-ho van registrar durant dies l'activitat de les ones en la zona costera de Califòrnia i després van calcular l'espectre de les ones capturades cada dia ja que, com veurem més avant, gràcies a l'espectre es pot obtenir informació sobre la direcció de propagació de les ones i la seva amplitud, pel que al representar el contorn d'energia de les ones en un diagrama temps-freqüència, es pot obtenir els angles necessaris per a localitzar l'origen de la tempesta.

### 5.5.4- Energia i altura significant d'ona

L'energia  $E$  d'una ona en *Joules* per metre quadrat està relacionada amb la variança del desplaçament de la superfície marina  $\zeta$  per l'expressió:

$$E = \rho_w g \langle \zeta^2 \rangle$$

on  $\rho_w$  és la densitat de l'aigua,  $g$  la gravetat i  $\langle \zeta^2 \rangle$  una mitja temporal o espacial.

En una ona regular, cada partícula està dotada d'una energia cinètica i potencial. Com el període i la longitud d'ona són constants, l'energia d'aquest tipus d'ona és la suma de les energies:

$$E = E_p + E_c = (\rho g \lambda b H^2)/8$$

on  $\rho$  és la densitat de l'aigua,  $H$  l'altura de l'ona i  $b$  l'amplària de la cresta.



A aigües profundes  $E = E_p = E_c$ , és a dir:

$$E = E_p = E_c = \frac{\rho g \lambda b H^2}{16}$$

al ser aigües profundes, la longitud d'ona val:

$$\lambda = \frac{g T^2}{2 \pi}$$

i al substituir en l'expressió de l'energia obtenim:

$$E = \frac{\rho g^2 T^2 b H^2}{32 \pi} = 979,2 b T^2 H^2 \text{ Wseg}$$

Donat que l'energia de les ones depèn del quadrat de la seua altura  $H$ , és evident que la disminució d'aquesta altura amb la profunditat  $h$  és important en l'estudi de la distribució de l'energia de les ones en profunditat. El càlcul de la pressió exercida per una sola ona contra un obstacle, degut a la transferència de la seva energia cinètica sobre el mateix, és de gran interès per l'aprofitament de l'energia de les ones. Esta pressió, a l'igual que passa amb l'energia, varia amb el quadrat de l'amplitud i s'atenua amb la profunditat de forma exponencial.

L'altura significant d'ona  $H_{1/3}$  és l'altura màxima d'un terç de les ones. La necessitat de definir aquest paràmetre no és altra que la varietat d'amplituds de les ones existents en una àrea de l'oceà tal i com es pot observar en la figura 74.

Aquest paràmetre es pot calcular de diferents formes. La més intuïtiva seria registrar durant uns minuts l'altura de, per exemple, 120 crestes d'ona i, una vegada fet açò, triar les 40 ones més grans i calcular la mitja d'aquests 40 valors.

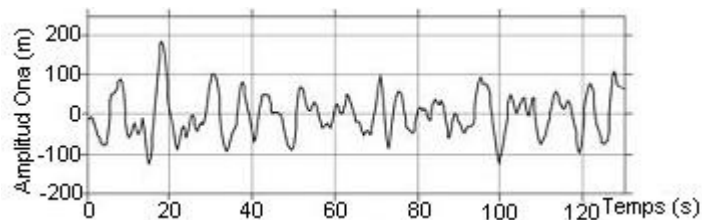


Figura 74: Curt registre de les amplituds d'ona mesurades per una boia en l'oceà atlàntic

Però la manera més útil de calcular l'altura de les ones és a partir del desplaçament d'ona mesurat aprofitant que  $H_{1/3}$  està relacionada amb la desviació estàndard del desplaçament de la superfície de l'oceà per l'expressió:

$$H_{1/3} = 4 < \zeta^2 >^{1/2}$$

on  $< \zeta^2 >$  és la desviació estàndard del desplaçament de la superfície.

### 5.5.5- Trencament de l'ona

Com ja hem comentat, les ones transporten l'energia subministrada per la força que les genera. L'energia que poden transportar està limitada, entre altres, per la profunditat de l'aigua i el vent, pel que una vegada se superen aquests límits, l'ona se saturarà d'energia i, com no serà capaç de contenir-la, acabarà trencant, és a dir, acabarà dissipant de colp tota l'energia que transportava.

Per norma general, una ona trenca quan la component horitzontal de la velocitat de les partícules d'aigua s'iguali a la celeritat, o el que és el mateix  $c = u$ , on  $u$  val:

$$u = \frac{\pi H}{T} e^{\frac{2\pi y}{\lambda}} \cos\left(\frac{2\pi}{\lambda} x - \omega t\right), \text{ a aigües profundes}$$

$$u = \frac{H}{2} \sqrt{\frac{g}{h}} \cos\left(\frac{2\pi}{\lambda} x - \omega t\right), \text{ a aigües poc profundes}$$

La condició de trencament implica  $t = x = 0$  i  $H = H_r$ , on  $H_r$  és l'altura de l'ona a trencar, pel que la part del cosinus val 1 i l'expressió se simplifica considerablement.

## 5.6- Teoria no lineal d'ones de superfície de l'oceà

Fins ara hem derivat les propietats d'una ona de superfície suposant que les ones són infinitament petites  $ka = 0$ . Si les ones són petites  $ka \ll 1$  però no infinitament menudes, les propietats de l'ona es poden desenvolupar en sèries de  $ka$  mitjançant el teorema de Stokes, de manera que estes queden representades per la següent expressió:

$$\zeta = a \cos(kx - \omega t) + \frac{1}{2} ka^2 \cos 2(kx - \omega t) + \frac{3}{8} k^2 a^3 \cos 3(kx - \omega t) + \dots$$

Les fases de les components de la sèrie de *Fourier* que resulten d'expandir l'expressió anterior són tals que les ones no lineals tenen crestes empinades i depressions aplanades. L'amplitud màxima d'una ona d'aquest tipus en aigües profundes<sup>24</sup> és  $a_{\max} = 0.07 L$  ( $ka = 0.44$ ).

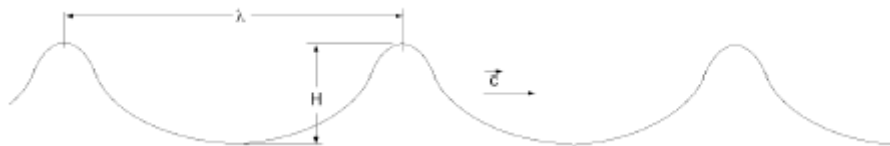


Figura 75: Ona de Stokes

Per a descriure el moviment de les ones no lineals a aigües poc profundes, Stokes proposa la següent expressió:

$$y = \frac{H}{2} \cos\left(\frac{2\pi x}{\lambda} - \frac{2\pi t}{T}\right) + \frac{3}{64} \frac{\lambda^2 H^2}{\pi^2 h^3} \cos\left\{2\left(\frac{2\pi x}{\lambda} - \frac{2\pi t}{T}\right)\right\}$$

on la longitud d'ona  $\lambda$  i la velocitat  $c$  valen el mateix que en les ones lineals.

En aquest cas, la component horitzontal de la velocitat de les partícules d'aigua val:

$$u = \frac{\lambda H}{2hT} \cos\left(\frac{2\pi x}{\lambda} - \frac{2\pi t}{T}\right) + \frac{3}{64} \frac{\lambda^3 H^2}{\pi^2 h^4 T} \cos\left\{2\left(\frac{2\pi x}{\lambda} - \frac{2\pi t}{T}\right)\right\}$$

mentre que la profunditat de l'aigua a la qual trenca l'ona és:

$$H_r = \frac{16\pi^2 h^2}{3gT^2} \left(-1 + \sqrt{1 + \frac{3gT^2}{4\pi^2 h}}\right)$$

<sup>24</sup> A aigües profundes, aquest tipus d'ones empinades també rep el nom d'Ones de Stokes

i la seva energia vindrà definida per:

$$E = \frac{\rho g H^2 \lambda b}{8} \left( 1 + \frac{9}{64} \frac{H^2}{\left(\frac{2\pi}{\lambda}\right)^4 h^6} \right)$$

amb  $c_g = c = \sqrt{g h}$

Comparant aquests valors amb els valors obtinguts en les equacions de la teoria d'ones lineal podem concloure que *Stokes* les modifica amb un factor de correcció que tendix a 0 en aigües profundes però que per a aigües poc profundes val:

$$\frac{9}{64} \frac{H^2}{\left(\frac{2\pi}{\lambda}\right)^4 h^6}$$

L'anàlisi d'aquest tipus d'ona va ser lent fins que, en els anys 60, *Hasselmann* va utilitzar els teoremes de física de partícules d'alta energia per a calcular la serie de sext ordre que representa les interaccions de tres o més ones de superfície. A més, amb la col·laboració de *Phillips* i *Longuet-Higgins* van demostrar que,  $n$  ones lliures en la superfície del mar poden interactuar entre si per a produir una altra ona lliure sí la suma de les freqüències i els números d'ona de les ones que interactuen és zero:

$$\begin{aligned} \omega_1 \pm \omega_2 \pm \omega_3 \pm \dots \pm \omega_n &= 0 \\ k_1 \pm k_2 \pm k_3 \pm \dots \pm k_n &= 0 \\ \omega_i^2 &= g k_i \end{aligned}$$

sempre que permetem que les ones viatgen en qualsevol direcció i  $k_i$  és el vector nombre d'ona que proporciona la longitud i direcció. Destacar que les dos primeres expressions en realitat són els requisits generals per a que es produïska una ona lliure a partir de d'interacció d'altres tres ones i que per a que açò passe, les ones han d'interactuar centenars de períodes i longituds d'ona abans que aparega una quarta ona d'amplitud comparable a la de les altres ones que la generen. Un exemple d'ones que no compleixen aquests criteris són les ones de *Stokes*, doncs els harmònics de major freqüència estan lligats a l'ona principal.

### 5.6.1- Teoria de l'ona solitària o *Tsunami*

Les ones solitàries són un altre tipus d'ones no lineals. Aquest subclasse d'ones es caracteritzen per propagar-se sense canviar de forma, per estar sempre per dalt del nivell del mar i perquè dos ones solitàries poden creuar-se sense interactuar l'una amb l'altra.

Aquestes propietats són el resultat d'un balanç exacte entre la dispersió, la qual tendix a dividir l'ona solitària en un tren d'ones, i els efectes no lineals, els quals tendixen a acurtar i a aplanar l'ona.

L'expressió que representa aquestes ones és la següent:

$$y = H \operatorname{Sech}^2 \left\{ \sqrt{\frac{3H}{4h^3}} (x - ct) \right\} ; \quad c = \sqrt{gH \left( 1 + \frac{H}{h} \right)}$$

Per a aquestes ones la velocitat horitzontal de les partícules d'aigua ve definida per:

$$u = \sqrt{\frac{g}{h}} y = \sqrt{\frac{g}{h}} H \operatorname{Sech}^2 \left\{ \sqrt{\frac{3H}{4h^3}} (x - ct) \right\}$$

En canvi, la condició de trencat de l'ona val  $H_r = 0,714 h_r$  mentre que l'energia de la mateixa en la zona de mar de fons del litoral ve donada per l'expressió:

$$E = 1,54 \gamma \sqrt{(H h)^3} b$$

Com es pot observar, l'energia decreix ràpidament amb l'altura  $h$  pel que esta zona no es considera bona per a la conversió i aprofitament de l'energia de les onades.

## 5.7- Mesurant ones: Espectre d'una ona

Com segurament haurem observat alguna de les vegades que ens hem fixat en les ones del mar, estes no són simples sinusoides. De fet, el mar pareix estar format per ones de diferents longituds i períodes. Aleshores, com podem descriure esta superfície? La realitat és que és molt complicat fer-ho de forma exacta. Siga com siga, tot i que no es pot fer de forma exacta, sí es pot descriure de forma aproximada si simplifiquem alguns aspectes. Aquest procés de simplificació ens porta al que es coneix com l'espectre de les ones del oceà: la distribució de l'energia de l'ona entre les diferents freqüències i longituds d'ona que hi ha en eixe moment en la superfície del mar.

El concepte d'espectre es basa en el treball de *Joseph Fourier*, qui va demostrar que qualsevol funció  $\zeta(t)$  es pot representar sobre l'interval  $-T/2 < t < T/2$  com la suma d'una serie infinita de sinus i cosinus amb freqüències d'ona harmòniques. Aquesta expressió rep el nom de serie de *Fourier* i ve representada per l'expressió:

$$\zeta(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos 2\pi n f t + b_n \sin 2\pi n f t)$$

amb

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} \zeta(t) \cos 2\pi n f t dt, \quad b_n = \frac{2}{T} \int_{-T/2}^{T/2} \zeta(t) \sin 2\pi n f t dt, \quad (n = 0, 1, 2, \dots)$$

on  $f = 1/T$  és la freqüència fonamental,  $n f$  són els seus harmònics i  $a_0$  és la mitja de  $\zeta(t)$  en l'interval.

Sí simplifiquem l'expressió anterior utilitzant:

$$\exp(2\pi i n f t) = \cos(2\pi n f t) + i \sin(2\pi n f t)$$

on  $i$  és el nombre imaginari  $\sqrt{-1}$ , aleshores la serie de *Fourier* es pot escriure de la següent forma:

$$\zeta(t) = \sum_{n=-\infty}^{\infty} Z_n \exp^{i2\pi n f t}$$

amb

$$Z_n = \frac{1}{T} \int_{-T/2}^{T/2} \zeta(t) \exp^{-i2\pi n f t} dt, \quad (n = 0, 1, 2, \dots)$$

aquesta última expressió rep el nom de transformada de *Fourier* de  $\zeta(t)$ .

L'espectre  $S(f)$  de  $\zeta(t)$  és:

$$S(nf) = Z_n Z_n^*$$

on  $Z^*$  és el conjugat complex de  $Z$ .

Òbviament, igual que podem expressar amb series de *Fourier* funcions d'una incògnita, podem expressar també funcions de dos incògnites, és a dir, superfícies.

Per a aplicar aquesta idea a la superfície de l'oceà suposarem que, en un instant en concret, les ones estan congelades en el temps. Utilitzant el desenvolupament de la serie de *Fourier*, la superfície congelada es pot representar com una serie infinita de sinus i cosinus de diferents números d'ona orientats en totes les possibles direccions. Sí ara descongelem la superfície del oceà i deixem que evolucione amb el temps, podrem tornar a representar la superfície utilitzant infinites series de sinus i cosinus, pues les longituds i les freqüències d'ona estan relacionades per la relació de dispersió.

Cal destacar que, mentre considerem distancies i intervals de temps curts, podrem assumir que els coeficients de les series de *Fourier* són constants ja que les interaccions entre ones en eixes condicions són dèbils, però quan no es donen aquestes condicions i estiguem parlant d'intervals de temps de varies hores o d'àrees de centenars de kilòmetres, la cosa canvia i la superfície de l'oceà s'ha de representar com la superposició lineal d'ones sinusoidals reals amb diferents direccions, pel que les series de *Fourier* confirmen que la teoria lineal d'ones de superfície, o el que és el mateix: la superfície del mar està formada per ones independents.

### 5.7.1- Mostreant la superfície de l'oceà

Com calcular les series de *Fourier* que representen la superfície del mar és impossible, ja que deuríem mesurar durant un període de temps llarg l'altura de la superfície de l'oceà  $\zeta(x, y, t)$  en tots els punts d'una zona de, per exemple, 10 kilòmetres, el que es fa és instal·lar un sensor d'ones en un punt qualsevol de l'oceà per a capturar l'altura de la superfície del mar com una funció del temps  $\zeta(t)$ . Sí digitalitzem aquestes mesures de manera que:

$$\zeta_j \equiv \zeta(t_j), \quad t_j \equiv j\Delta$$

$$j = 0, 1, 2, \dots, N - 1$$

on  $\Delta$  és l'interval de temps entre mostres i  $N$  és el nombre total de mostres. La longitud  $T$  del registre valdrà  $T = N\Delta$  i al representar estes mesures obtindrem una cosa pareguda al que podem veure en la figura 75:

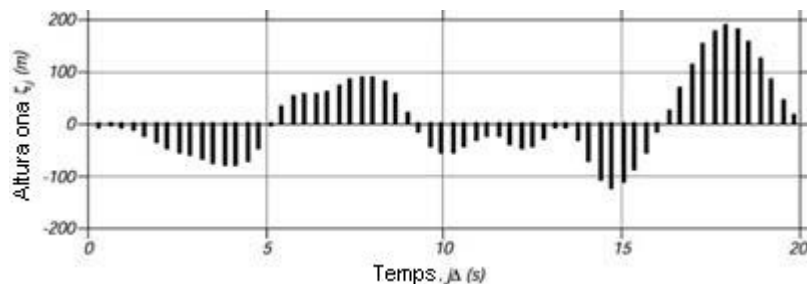


Figura 75: digitalització per a  $\Delta = 0,32$ s dels 20 primers segons registrats per la boia de la figura 74.

Treballar d'aquesta manera ens permetrà mesurar totes les ones de la superfície del mar però no tindrem ni idea de la direcció de desplaçament de les mateixes. A més, hem passat de tenir un nombre de mostres infinit a un finit,  $\zeta_j \neq \zeta(t)$ , pel que entre mostra i mostra no tindrem cap tipus

d'informació d'altura. Açò pot suposar un problema important, pues sí no complim el teorema de *Nyquist*, és a dir, sí les mostres no es prenen com a mínim al doble de freqüència de la component freqüencial més alta de l'ona, es produirà *Aliasing* pel que, com es pot veure en la figura 76, serà impossible recuperar correctament la senyal original.

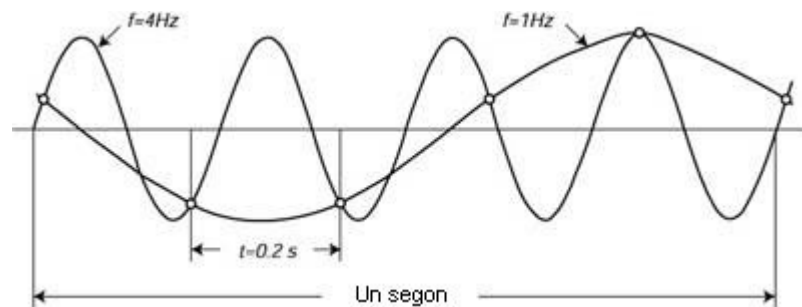


Figura 76: Mostrejar una senyal sinusoidal de 4Hz cada 0.2s transforma la senyal en una ona de 1Hz doncs  $5\text{Hz} < 8\text{Hz}$

Aquest efecte deurà de tindre's en compte tant sí volem mesurar ones amb freqüències per dalt de la freqüència crítica de *Nyquist* del sistema, com sí volem registrar ones amb freqüència inferior a la freqüència fonamental, la qual vindrà determinada per la duració  $T$  del registre que emmagatzemarà la informació de les ones de freqüència compresa entre:

$$\frac{1}{T} < f < \frac{1}{2\Delta}$$

on  $T = N\Delta$  és la longitud del registre i  $f$  la freqüència en Hz.

### 5.7.2- Càlcul de l'espectre d'una ona

La transformada de *Fourier* digital d'un registre d'ones val:

$$Z_n = \frac{1}{T} \int_{-T/2}^{T/2} \zeta(t) \exp^{-i2\pi nft} dt, \longrightarrow Z_n = \frac{1}{N} \sum_{j=0}^{N-1} \zeta_j \exp[-i2\pi jn/N]$$

$$\zeta(t) = \sum_{n=-\infty}^{\infty} Z_n \exp^{i2\pi nft} \longrightarrow \zeta_n = \sum_{n=0}^{N-1} Z_j \exp[i2\pi jn/N]$$

per a  $j = 0, 1, \dots, N - 1$  i  $n = 0, 1, \dots, N - 1$ . Aquestes equacions poden sumar-se ràpidament utilitzant la transformada ràpida de *Fourier*, especialment sí  $N$  és una potència de 2.

El periodograma és l'espectre simple  $S_n$  i ve representat per:

$$S_n = \frac{1}{N^2} [|Z_n|^2 + |Z_{N-n}|^2]; \quad n = 1, 2, \dots, (N/2 - 1)$$

$$S_0 = \frac{1}{N^2} |Z_0|^2$$

$$S_{N/2} = \frac{1}{N^2} |Z_{N/2}|^2$$

En el periodograma,  $S_N$  està normalitzat mitjançant:

$$\sum_{j=0}^{N-1} |\zeta_j|^2 = \sum_{n=0}^{N/2} S_n$$

d'aquesta manera la variança de  $\zeta_j$  és la suma dels termes  $(N/2 + 1)$  del periodograma. Destacar que els termes  $S_n$  per dalt de la freqüència  $N/2$  són simètrics respecte eixa freqüència.

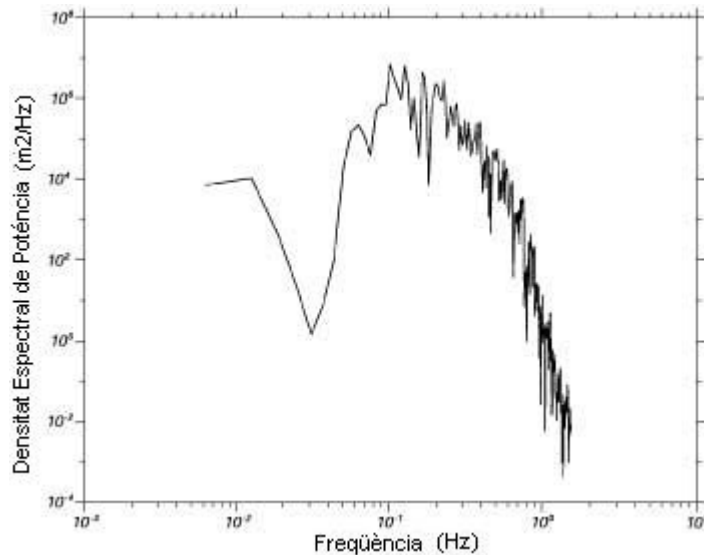


Figura 77: Periodograma dels 164 primers segons de la figura 74 (Freq. Nyquist = 1.5625Hz)

El periodograma és una funció molt sorollosa en la que la variança de cada punt és igual al valor esperat en eixe punt. Normalment, per a reduir la incertesa del valor a cada freqüència, es realitza la mitja d'entre 10 i 30 periodogrames. El periodograma resultant rep el nom d'espectre de l'altura de l'ona i proporciona la distribució de la variança de l'altura de la superfície de l'oceà com a funció de la freqüència en el punt en el que està el sensor. Com l'energia de les ones és proporcional a la variança, l'espectre rep el nom d'espectre d'energia i, per norma general, es necessiten com a mínim unes tres hores de registre per a poder calcular-lo.

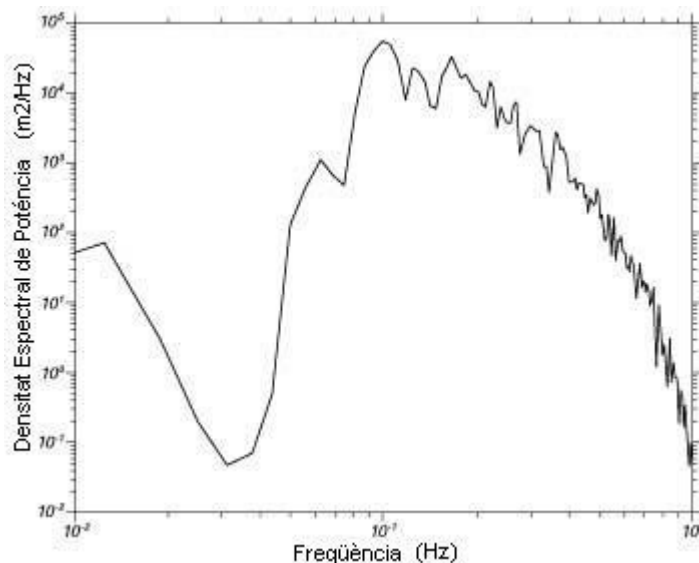


Figura 78: Representació gràfica de l'espectre d'un registre d'ones qualsevol d'onze minuts de duració.

## 6- Solucions possibles i opció triada

Avui en dia hi ha una quantitat enorme de dispositius electrònics que ens permeten fer el que ens proposem en aquest projecte pel que és impossible analitzar i comparar-los entre ells un a un, però tot i que no ens em complicat la vida estudiant les diferents propostes dels principals fabricants, és inevitable tenir que triar un candidat d'entre tots els que estan al nostre abast per a que s'encarregue de realitzar les funcions que precisem.

### 6.1- Hardware

#### 6.1.1- Components necessaris per al disseny

Per al disseny del sistema de control i posicionament d'un objecte o plataforma flotant necessitarem, a part dels típics components passius, una circuit d'alimentació, un processador, un acceleròmetre i un giroscopi de varis eixos, un receptor de GPS i, de forma opcional, un teclat per a controlar el sistema, un *display* per a visualitzar la informació facilitada pel sistema i un dispositiu de transmissió i/o emmagatzemament de les dades proporcionades pels diferents sensors.

#### 6.1.2- Etapa de processament de dades i control

Per a la part de control, com ens interessa disposar d'una placa de circuit imprès el més petita i simple possible, ens hem decantat per un microcontrolador ja que aquests disposen de, en un mateix encapsulat de petites dimensions, els recursos i perifèrics bàsics necessaris per al funcionament d'un sistema digital micro-processat. Dins de tots els fabricants i models de microcontroladors disponibles, ens hem decantat pels microcontroladors de la casa *Microchip*, principalment per la popularitat d'aquests microcontroladors, però també per la familiaritat de l'autor del projecte amb els mateixos. Aquesta popularitat és un factor a tenir en compte ja que ens pot donar una gran quantitat de recursos que faciliten el disseny de la placa i la programació del microcontrolador, pues amb una simple cerca en Internet podem trobar, tant exemples d'ús del mateix el microcontrolador, com gent que ens aconselle respecte a un problema que se'ns haja plantejat o inclús empreses que venen les seves pròpies plaques *demo* per a que qualsevol persona pugui realitzar ràpidament proves dels seus propis projectes sense quasi utilitzar recursos o inclús utilitzar-les com a ferramenta de *debugging* dels seus programes i, d'eixa manera, descartar possibles errades de disseny de la seva placa prototip.

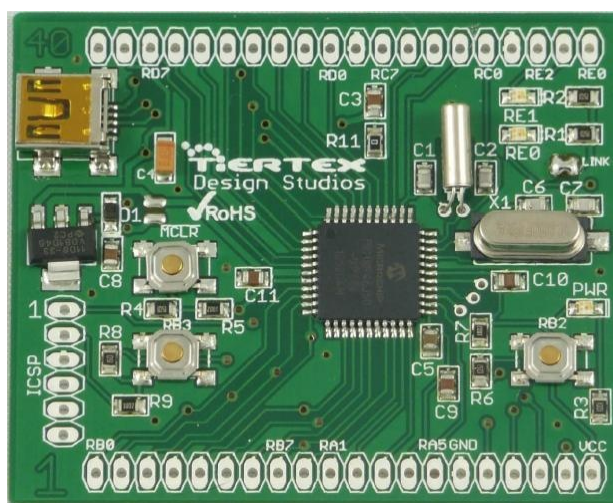


Figura 79: Placa demostració/prototips de la casa Tiertex



En un primer moment, donat que l'autor disposava d'una placa prototip amb un microcontrolador PIC de 8 bits com la de la figura, es va considerar seriosament l'opció d'utilitzar un microcontrolador d'ídntiques característiques per a implementar el disseny, però tal i com veurem més avant, la idea va quedar descartada en les primeres etapes de l'elaboració del projecte degut a les limitacions que imposava el seu ús.

Així doncs, considerant la complexitat de l'aplicació que deu implementar aquest projecte i, donat que la potència dels microcontroladors de 8 bits disponibles actualment podria ser insuficient per a realitzar totes les funcions que deuria de realitzar un sistema de control d'estabilitat d'una plataforma flotant, ens hem decantat per utilitzar un microcontrolador de 16 bits de baix consum de la casa *Microchip* de manera que aquest es limite a capturar les dades proporcionades pels diferents sensors/dispositius i les emmagatzeme o envie al exterior per a que un processador més potent les analitzi i actue conseqüentment.

| Model                           | Preu <sup>25</sup> | RAM   | Pins I/O | Nº Pins | Canals DMA | Comunicacions digitals | USB 2.0             | Port paral·lel       |
|---------------------------------|--------------------|-------|----------|---------|------------|------------------------|---------------------|----------------------|
| PIC24FJ256GB110                 | \$4.70             | 16KB  | 84       | 100     | -          | 4xUART, 3xSPI          | 1xOTG <sup>26</sup> | 1xPMP <sup>27</sup>  |
| PIC24FJ256GA110                 | \$4.45             | 16KB  | 85       | 100     | -          | 4xUART, 3xSPI          | -                   | 1xPMP                |
| PIC24FJ256DA110                 | \$5.18             | 24KB  | 84       | 100     | -          | 4xUART, 3xSPI          | 1xOTG               | 1xEPMP <sup>28</sup> |
| PIC24FJ256DA210                 | \$5.60             | 96KB  | 84       | 100     | 2          | 4xUART, 3xSPI          | 1xOTG               | 1xEPMP               |
| PIC24FJ256GB210                 | \$5.14             | 96KB  | 84       | 100     | -          | 4xUART, 3xSPI          | 1xOTG               | 1xEPMP               |
| PIC24FJ256GA108                 | \$4.24             | 16KB  | 69       | 80      | -          | 4xUART, 3xSPI          | -                   | 1xPMP                |
| PIC24FJ256GB108                 | \$4.49             | 16KB  | 68       | 80      | -          | 4xUART, 3xSPI          | 1xOTG               | 1xPMP                |
| PIC24FJ256GA106                 | \$3.98             | 16KB  | 53       | 64      | -          | 4xUART, 3xSPI          | -                   | 1xPMP                |
| PIC24FJ256GB106                 | \$4.23             | 16KB  | 52       | 64      | -          | 4xUART, 3xSPI          | 1xOTG               | 1xPMP                |
| PIC24FJ256GB206                 | \$4.65             | 96KB  | 52       | 64      | -          | 4xUART, 3xSPI          | 1xOTG               | 1xEPMP               |
| PIC24FJ256DA206                 | \$5.11             | 96KB  | 52       | 64      | 2          | 4xUART, 3xSPI          | 1xOTG               | -                    |
| PIC24FJ256DA106                 | \$4.69             | 24KB  | 52       | 64      | 2          | 4xUART, 3xSPI          | 1xOTG               | -                    |
| PIC18F46J50 <sup>29</sup>       | \$2.58             | 3,7KB | 34       | 44      | -          | 2xUART, 2xSPI          | 1x                  | 1xPMP                |
| DSPIC33EP256MU806 <sup>30</sup> | \$5.22             | 28KB  | 53       | 64      | 15         | 4xUART, 4xSPI          | 1xOTG               | 1xPMP                |

Taula 12: comparativa de diferents microcontroladors *Microchip*

Per a poder triar de la taula un microcontrolador que s'encarregue de gestionar el nostre sistema abans hem de conèixer els requisits de la nostra aplicació. En el nostre cas, un dels aspectes més importants és l'apartat de la comunicacions ja que les dades que capturem dels sensors seran processades per un altre processador i, per això, com s'observa en la taula 13, a l'hora d'escollir el mateix, ens hem centrat en este aspecte i hem obviat determinades característiques comuns als microcontroladors, com puga ser la quantitat de temporitzadors o la de canals del convertidor A/D, ja que no són característiques determinants a l'hora de l'elecció final, com podria ser el cas de característiques que podrien ser més crítiques com, per exemple, la quantitat de memòria RAM.

<sup>25</sup> Preu especificat pel fabricant en la seva pàgina web, en distribuïdors *online* el preu és més elevat.

<sup>26</sup> USB OTG, USB On-The-Go: USB "al vol". Permet que un dispositiu USB intercanvie dinàmicament el seu paper com a host o dispositiu amb els altres dispositius connectats al bus.

<sup>27</sup> PMP, Parallel Master Port

<sup>28</sup> EPMP, Enhanced Parallel Master Port

<sup>29</sup> Aquest PIC s'ha inclòs a mode anecdòtic ja que es va utilitzar en les primeres fases del projecte

<sup>30</sup> Aquest PIC s'ha inclòs per compartir característiques físiques i tècniques amb el PIC escollit

Aleshores, sí a l'hora de triar el dispositiu de control ens centrem en les nostres necessitats de comunicació, podem veure com ja hi han tres candidats a ser descartats: els dos PICs DA i el PIC de 8 bits: el 18F46J50. Que ens porta a descartar-los? Doncs en el cas del PIC de 8 bits és la quantitat de ports de comunicació digital disponibles, pues tot i que deuria de ser suficient per a la nostra aplicació, utilitzar-lo limitaria les opcions d'ampliació del sistema ja que d'entrada tenim tots els ports SPI ocupats i una UART ocupada pel receptor GPS, pel que sols ens quedaria lliure una UART i, en cas de voler afegir alguna funció al sistema, estaríem obligats a tenir que utilitzar un dispositiu compatible amb aquest port. En canvi, el que fa perdre punts als PICs DA a l'hora de ser escollits és l'absència d'un port paral·lel, ja que tot i que és cert que no és necessari una *display* de port paral·lel per a que el nostre sistema funcione, sí és veritat que, degut a la naturalesa del projecte, ens interessa poder desplaçar fàcilment el sistema i visualitzar en temps real les lectures obtingudes, pel que disposar d'aquest element facilitaria la implementació i depuració del sistema ja que no sols ens evitaríem estar lligats al ús d'un PC per conèixer l'estat del sistema, sinó que a més, no caldria dedicar part del nostre temps a la implementació d'una rutina que s'encarregués d'emular el comportament del port per a poder comunicar-se amb una pantalla, ja fos de tipus paral·lel com de tipus SPI.



Figura 80: PIC24FJ256GA108 amb encapsulat TQFP

Tot i que l'apartat de les comunicacions és una de les coses que més pes deu tenir en la nostra elecció, n'hi ha una altra que té quasi la mateixa importància: les dimensions del microcontrolador. Com que interessa que la placa final ocupe el menor espai possible i un xip de molts terminals sol complicar el disseny del circuit elèctric, ens interessa un microcontrolador amb el menor nombre de pins possible, raó per la que els cinc PICs de 100 terminals poden ser descartats sense problemes. El mateix es pot dir dels dos PICs de 80 terminals ja que, a nivell de comunicacions, són pràcticament idèntics les membres de 64 pins i és absurd pagar més per una cosa que no ens farà pràcticament el mateix paper que una altra que costa menys. Aquest detall pot parèixer insignificant ja que la diferència de preu és d'aproximada ment un euro, però sí el sistema arribés a produir-se en serie, esta diferència seria molt important ja que no estaríem parlant d'un euro, sinó de centenars de mils.

Després d'utilitzar les necessitats del nostre sistema com a filtre en l'elecció de candidats ens hem quedat amb tres candidats: el PIC24FJ256GA106, el PIC24FJ256GB106 i el PIC24FJ256GB206. Com podem veure en la taula 11, tots tres presenten característiques de comunicació similars però n'hi ha un que destaca per tenir sis vegades més RAM: el PIC 24FJ256GB206. Aquest fet i el tenir un port paral·lel millorat (EPMP) respecte el dels altres PICs deuria decantar la balança cap al seu costat, però la realitat és que, tot i que aquests "extres" el fan una opció molt a tenir en compte, no és la millor elecció degut a la seva falta de compatibilitat a nivell d'assignació de terminals amb PICs de major potència ja que, a part dels factors anomenats anteriorment, se n'ha tingut un altre que també ha influït en la decisió final: la possibilitat d'aprofitar el disseny elèctric inicial del circuit per a substituir el microcontrolador de baix consum per un altre de major potència per sí de cas el microcontrolador escollit no fou prou potent o simplement se li volgués donar un altre enfocament al sistema. Per això, l'opció escollida per a realitzar el control dels sensors del sistema no ha sigut una altra que la del PIC24FJ256GB106 ja que, com que el PIC256GA106 no disposa de port USB, era l'únic que quedava compatible a nivell de pins amb el DSPIC33EP256MU806, un dels híbrids microcontrolador-DSP més complets que hi havia quan es va iniciar aquest projecte.

### 6.1.3- Etapa de sensat

Dins d'aquesta etapa inclourem tant els sensors d'inèrcia necessaris per a conèixer el moviment experimentat per la nostra plataforma com l'encarregat de localitzar per satèl·lit la posició de la mateixa.

#### 6.1.3.1- Elecció del receptor de GPS

Escollir el receptor de GPS fou de les parts més fàcil del projecte ja que avui en dia la gran majoria d'aquests dispositius es comunica amb l'exterior mitjançant un port asíncron (UART) i, a més, permeten enviar les dades de posicionament d'acord amb les especificacions del protocol NMEA<sup>31</sup>, raó per la que es pot dir que l'única cosa per la que ens hem preocupat a l'hora de l'elecció és el preu. Aleshores, sí ens dirigim a les webs de *Farnell* i *RS-Amidata*, realitzem una cerca amb el terme *GPS Receiver* i ens fixem un preu màxim d'uns 50 euros, tenim les següents opcions:

| Descripció          | NMEA | Comunicació | Connexions               | Preus   |         |
|---------------------|------|-------------|--------------------------|---------|---------|
|                     |      |             |                          | RS      | Farnell |
| NV-08C-M; 1902503   | X    | UART/SPI    | BGA, Sense antena ext.   | -       | 20,21 € |
| GPS 1513R; 704-3307 | X    | UART        | LGA, Sense antena ext.   | 22,49 € | -       |
| IT-03; 537-739      | X    | UART        | LGA, Sense antena ext.   | 30,03 € | -       |
| NV-08C-C; 1902504   | X    | UART/SPI    | LGA, Sense antena ext.   | -       | 34,81 € |
| GPS-610F; 704-3297  | X    | UART        | 10x2mm, Amb antena ext.  | 31,80 € | 42,83 € |
| A-1080-A; 691-2388  | X    | UART        | LGA, Sense antena ext.   | 38,53 € | -       |
| GPS 330R; 704-3304  | X    | UART        | 6x1mm, Sense antena ext. | 41,99 € | 46,95 € |
| GPS 622R; 704-3300  | X    | UART        | 7x1mm, Sense antena ext. | 41,99 € | -       |
| GPS-310FS; 704-3313 | X    | UART        | LGA, Amb antena ext.     | 46,95 € |         |
| IT 321; 1656877     | X    | UART        | LGA, Sense antena ext.   | -       | 48,90 € |

Taula 13: Relació Característiques de Receptors de GPS de menys de 50€

Com podem veure en la taula 13, no n'hi ha cap receptor que no siga compatible amb el protocol NMEA o amb la comunicació asíncrona, pel que teòricament, com no necessitem precisió ni velocitat, ens pot servir qualsevol. En la pràctica, degut a les limitacions que tenim a l'hora de crear la PCB prototip del sistema, és millor limitar-se a receptors que no utilitzen els encapsulats BGA o LGA per a connectar-se a la PCB, així que de 10 possibles candidats ens quedem amb 3: el GPS-610F, el GPS 330R i el GPS 622R. D'aquests tres, al final optarem pel GPS 610F per dues raons: preu i possibilitat de poder utilitzar una antena externa. Possiblement qualsevol dels altres dos receptors haurien sigut millors candidats pel tema de consums energètics o sensibilitat, però com aquestes diferències no són significatives entre uns i altres, tots els receptors comparteixen protocol de comunicació i són compatibles amb els principals serveis de millora del servei GPS, com són el WAAS o el EGNOS<sup>32</sup>, mereix la pena sacrificar un poc de rendiment per a poder utilitzar i instal·lar on desitgem una antena externa, cosa que, per cert, a part de millorar la recepció de la senyal, facilitarà el disseny del circuit elèctric ja que haurem de preocupar-nos menys pel tema de les interferències electromagnètiques que puga produir la part de RF del propi receptor, i a més, no ens forçarà a utilitzar l'antena interna pel que augmentarem de manera considerable les probabilitats de rebre correctament la senyal GPS.

31 Per a detalls sobre NMEA, veure la secció 2.6.1: Un protocol per a la transmissió de missatges GPS: NMEA-0183

32 Per a més detalls sobre aquest serveis, veure la secció 2.2.4- GPS Diferencial i Sistemes SBAS

### 6.1.3.2- Elecció dels sensors d'inèrcia

Sí abans dèiem que escollir el receptor de GPS no havia sigut molt complicat, ara podríem dir tot el contrari de l'elecció dels sensors d'inèrcia ja que actualment la varietat d'aquests és enorme i, per exemple, introduir *accelerometer* en el buscador de productes de *Farnell* o *RS* dona una mitja de resultats superior als 150 components, pel que sí tinguérem la tecnologia necessària per a soldar-los en la placa prototip i ens veiéssim obligats a analitzar-los un per un, segurament acabariem bojos. Afortunadament molts d'aquests components utilitzen el encapsulat LGA o BGA pel que la llista es redueix prou, així que sí ens centrem en el nombre d'eixos a controlar i en quina tecnologia s'adaptarà millor al que necessitem, podrem triar més fàcilment quins sensors utilitzar en el disseny.

Aleshores, sí intentem decidir quins sensors dels oferits per *RS* o *Farnell* utilitzarem en el nostre disseny segons la seva tecnologia de fabricació, veurem que sols tenim dos opcions: la MEMS i la piezoelèctrica. Com és d'esperar, ambdues tecnologies tenen els seus avantatges i desavantatges:

| <b>MEMS vs Sensors Piezoelèctrics</b>   | <b>Sensors Piezoelèctrics vs MEMS</b> |
|---|---------------------------------------|
| Baix cost                               | Major precisió i rang de mesura       |
| Dimensions reduïdes                     | Gran robustesa i fiabilitat           |
| Fàcils d'integrar en un sistema digital |                                       |

Taula 14: breu comparació dels avantatges i desavantatges dels sensors segons la tecnologia utilitzada.

Com mostra la taula 14, la tecnologia MEMS té uns avantatges que la fan interessant per a qualsevol tipus de projecte ja que, per exemple, sí necessitarem mesurar l'acceleració en tres eixos, com els acceleròmetres piezoelèctrics no es venen en configuracions multiaxials, necessitariem tres d'estos, pel que no sols ens caldria més superfície física per a poder utilitzar-los, sinó que a més, el pressupost econòmic augmentaria considerablement, i això és una cosa molt a tenir en compte perquè estos poden valdre unes 200 vegades més que un acceleròmetre MEMS i, tot i que és cert que no tenen la mateixa precisió i/o resolució, també és cert que en aplicacions com la nostra, on aquest factor no és excessivament crític, marca les diferències.

Una vegada triada la tecnologia en la que es basaran els sensors del sistema, hem de decidir la quantitat d'eixos a controlar, o el que és el mateix: sí utilitzarem sensors individuals per a cada eix a mesurar o bé sí utilitzarem un component multiaxial. En aquest cas, la decisió és un poc més complicada pel fet que el preu d'una configuració respecte a l'altra arriba a ser més deu vegades més alta, doncs la combinació individual d'acceleròmetres i giroscopis pot estar com a màxim al voltant dels 40-50 euros, mentre que tots eixos components en un mateix encapsulat pot arribar perfectament als 1000 euros, una diferència suficientment important com per a que meresca la pena pensar-se un poc la decisió. Al igual que hem fet abans, ens basarem en els avantatges i desavantatges de cada configuració per a descartar opcions:

| <b>Configuració Monoaxial vs Multiaxial</b> | <b>Configuració Multiaxial vs Monoaxial</b>        |
|---|--|
| Preu  | Utilitzen menys recursos del microcontrolador      |
| Menor superfície ocupada                    | No compliquen el disseny ni la seva instal·lació   |
| Major varietat de sensors/fabricants        | No solen necessitar adequació de senyal ni filtrat |
|   | Poden incloure altres sensors                      |
|   | Connector ràpid: fàcilment reemplaçables           |
|   | Major precisió/resolució                           |

Taula 15: Avantatges i desavantatges d'una configuració monoaxial d'un sensor respecte a la multiaxial.

Com ja havíem avançat, l'elecció no és fàcil perquè tant el preu com la varietat de sensors disponibles són aspectes importants en qualsevol disseny i això és una cosa que afavoreix a les configuracions monoaxials, però com ja hem vist en l'elecció del processador del sistema, l'opció que pareix millor no sempre és la millor ja que, per exemple, una major varietat de sensors implica tenir que estudiar-los individualment per a fer-los funcionar, igual que per a aconseguir el mateix resultat que una configuració multiaxial, necessitaríem no sols un parell de sensors per cada eix de mesura, sinó que a més hauríem de dissenyar una estructura el més estable i alineada possible amb el nostre centre de referència, pel que sí sumem tots els recursos i hores necessàries per l'estudi dels sensors i per al disseny d'un sistema que obtinga resultats similars als d'una configuració multiaxial, la diferència de preu s'inclina a favor de les configuracions multiaxials i les converteix en l'elecció ideal per als nostres propòsits, pues a més de rendibilitzar-los ràpidament, tenen major exactitud.

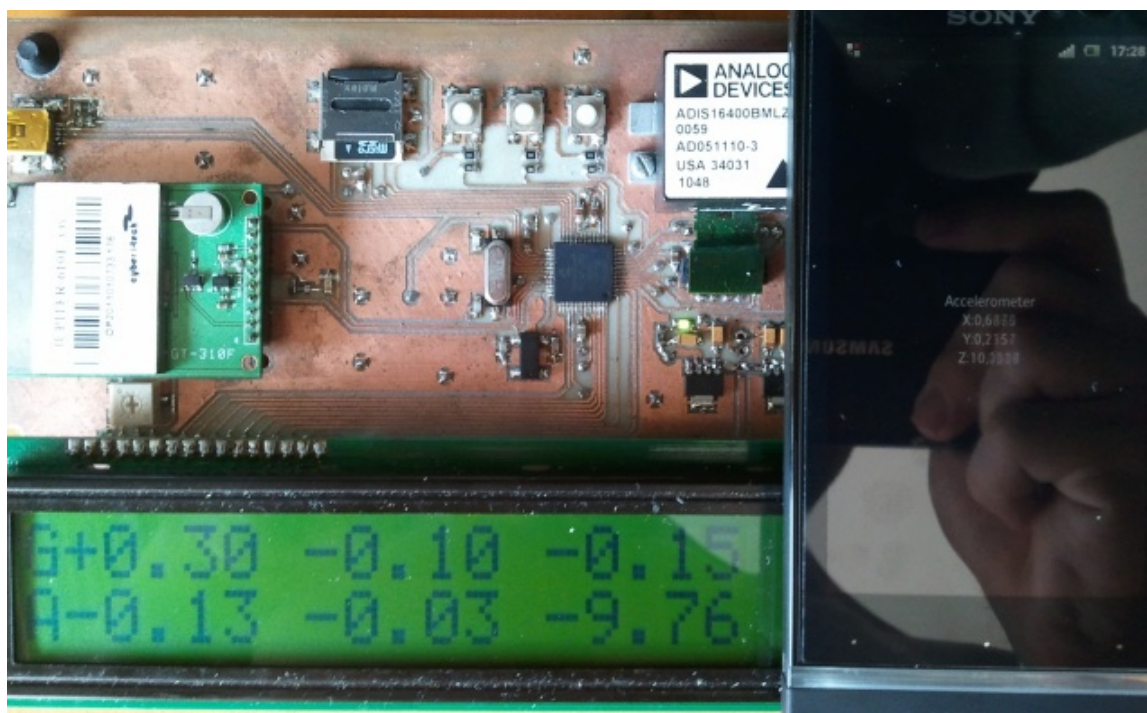


Figura 81: els acceleròmetres d'un *smartphone* de gama mitja poden desviar-se fins  $0.5 \text{ m/s}^2$  respecte als d'una IMU ADIS16400.

De totes les opcions que ens donaven Farnell i RS en l'inici del projecte ens vam decantar per la IMU<sup>33</sup> ADIS 16400 per diferents raons. La més important va ser el fet de l'imminent retirada del mercat d'una IMU que ens va cridar l'atenció pel seu preu: l'ADIS 16350. Un altre detall important va ser el nombre de graus de llibertat<sup>34</sup>, pues en la 16350 "sols" teníem 6 graus mentre que en la 16400 en són 9, és a dir tenim un sensor més per eix: un magnetòmetre. Com veurem més avant la inclusió d'un sensor d'aquest tipus no està feta a l'atzar i per això es va tenir en compte en l'elecció. També van influir altres detalls com el fet que la IMU 16350 no tornaria a estar en estoc fins passat l'estiu, però cap d'aquests detalls de la importància dels altres dos.

Sí actualment tornarem a triar una nova IMU, l'elecció seria la mateixa ja que des que es va iniciar aquest treball de final de carrera fins que s'ha acabat, *Analog Devices* ha introduït més IMU's al seu catalog, però de característiques similars a la de la solució triada, pel que un canvi d'IMU sols estaria justificat sí la nostra IMU deixara de produir-se en breu o bé sí necessitem més graus de llibertat. Independentment del motiu, el canvi d'IMU tindria un impacte mínim en el sistema ja que, de moment, tots els sensors d'inèrcia del catalog de *Analog Devices* són compatibles cap arrere amb altres IMUs del fabricant, pel que segurament els canvis es limitarien exclusivament a modificar les temporitzacions utilitzades en les rutines de lectura/escriptura del dispositiu.

33 IMU, *Inertial Measurement Unit*: Unitat de Mesura Inercial.

34 Per a un poc més d'informació sobre els graus de llibertat es recomana acudir a la secció 3.5: Sensors MEMS

### 6.1.4- Etapa d'alimentació

Per al disseny del circuit d'alimentació, com que anem a dissenyar un sistema mòbil alimentat a bateries, ens hem limitat a utilitzar reguladors de tensió per a alimentar els diferents components.

El fet que el sistema vaja a ser alimentat a bateries ha simplificat l'elaboració d'aquesta part, pues ha fet innecessari el disseny d'una font d'alimentació commutada, i ha sigut el principal motiu d'haver triat reguladors del tipus LDO<sup>35</sup>, doncs aquest tipus de reguladors, gràcies a la poca tensió que cau a la seva eixida, els fa més eficients i, per tant, ideals per a ser utilitzats per equips alimentats a bateries.

Com qualsevol regulador LDO que complisca les especificacions de tensió i corrent d'eixida ens pot servir, per a no dedicar més temps del necessari en l'elecció, hem optat per visitar les webs dels principals fabricants de semiconductors i ens hem quedat amb els reguladors de la casa que més fàcil ens ho ha posat per a veure i estudiar el seu catàleg de reguladors: *National Semiconductor*.

| Part   | Output Current (mA) ▼ | Output Min (Volt) | Input Min Voltage (Volt) | Input Max Voltage (Volt) | Dropout Voltage (Volt) | Packaging                 |
|--------|-----------------------|-------------------|--------------------------|--------------------------|------------------------|---------------------------|
| LP3871 | 800                   | 2.5 1.8 5 3.3     | 2.5                      | 7                        | 0.24                   | SOT-223 TO-263            |
| LM1117 | 800                   | 2.5 1.8 5 1.25    | 2.6 2.65                 | 15                       | 1.2                    | SOT-223 LLP TO-220 TO-263 |
| LP3874 | 800                   | 2.5 1.8 5 3.3     | 2.5                      | 7                        | 0.24                   | SOT-223 TO-263            |
| LP3875 | 1500                  | 2.5 1.8 5 3.3     | 2.5                      | 7                        | 0.38                   | SOT-223 TO-220 TO-263     |
| LP3879 | 800                   | 1 1.2             | 2.5                      | 6                        |                        | PSOP LLP                  |
| LP3961 | 800                   | 2.5 1.8 5 3.3     | 2.5                      | 7                        | 0.24                   | SOT-223 TO-263            |
| LP3962 | 1500                  | 2.5 1.8 5 3.3     | 2.5                      | 7                        | 0.38                   | SOT-223 TO-220 TO-263     |
| LP3964 | 800                   | 2.5 1.8 1.215 3   | 2.5                      | 7                        | 0.24                   | SOT-223 TO-220 TO-263     |
| LP3965 | 1500                  | 2.5 1.8 1.215 3   | 2.5                      | 7                        | 0.38                   | SOT-223 TO-220 TO-263     |
| LP3981 | 300                   | 2.5 2.7 2.8 2.85  | 2.7                      | 6                        | 0.132                  | LLP MINI SOIC             |
| LP3982 | 300                   | 3 2.5 1.8 1.25    | 2.5                      | 6                        | 0.12                   | LLP MINI SOIC             |
| LP3983 | 5                     | 2.5 1.6           | 2.5 2.7                  | 6                        |                        | MICRO SMD                 |
| LP3984 | 150                   | 1.8 3.1 2.9       | 2.5                      | 6                        | 0.06                   | MICRO SMD SOT-23          |
| LP3985 | 150                   | 3 2.5 3.2 2.7 3.  | 2.5                      | 6                        | 0.06                   | MICRO SMD SOT-23          |
| LP3987 | 150                   | 2.6 2.85 2.8      | 3 3.05 2.8               | 6 6.5                    | 0.06                   | MICRO SMD                 |
| LP3988 | 150                   | 3 2.5 2.6 2.85    | 3.2 2.7 3.1 2.8          | 6                        | 0.08                   | MICRO SMD SOT-23          |

Taula 15: una petita llista de la varietat de reguladors LDO de *National Semiconductor*

Al conèixer en el nostre sistema dispositius que funcionen tant a 3.3 volts com a 5 volts, es fa necessari l'ús de dos reguladors diferents: un de 3.3 volts i un altre de 5 volts. La suma dels consums individuals dels components exigeix que la corrent d'eixida siga major de 150mA, pel que per a no quedar-nos curts i, ja posats, poder afegir en un futur més dispositius al sistema sense haver de canviar el circuit d'alimentació, el que hem fet és triar reguladors capaços de subministrar més de 0.5 Ampers a la seva eixida. Com en *Farnell*, de tot el catàleg de *National Semiconductor*, sols tenien dos reguladors que complien els nostres requisits vam optar per utilitzar un LP3874 per a proporcionar els 5 volts i, en el cas del regulador de 3.3 volts, vam optar pel LP3871.

### 6.1.5- Interfície Entrada/Eixida

Tot i que per al funcionament d'aquest projecte no cal una part que permeti a l'usuari interactuar amb el sistema, és una opció interessant perquè d'eixa manera el procés de depuració és més pràctic i, a més, permet a l'usuari conèixer més fàcilment l'estat de tots els components.

<sup>35</sup> LDO, *Low Dropout*; Baixa caiguda (de tensió) a l'eixida.

Com esta és la part menys important del projecte, sols s'han seguit dos criteris a l'hora de triar els elements relacionats amb aquesta part: preu i estoc del component en qüestió.

Per a implementar aquesta etapa del projecte, per tal d'aprofitar-los el millor possible, s'han analitzat els recursos *hardware* que quedaven lliures en el microcontrolador després d'elegir els components de l'etapa de sensat. Així doncs, hem connectat a un dels ports SPI que quedava lliure un lector de targetes micro SD, hem instal·lat tres pulsadors en el port B del micro controlador per a poder interactuar amb ell, hem muntat un connector mini USB en els corresponents terminals del PIC i, finalment, hem afegit un *display* LCD de 20 caràcters i 2 línies per a mostrar les lectures dels sensors i permetre a l'usuari definir algunes funcions del sistema. Ens hauria agradat poder utilitzar una pantalla de més línies ja que açò hauria simplificat el software, però al no tenir-ne una a mà i tractar-se d'un component opcional, descartarem comprar-ne una millor.

## 6.2- Software

A l'hora d'elegir les ferramentes de desenvolupament del programa que executarà el microcontrolador, no hem tingut en compte l'eficiència o la potència de les ferramentes existents, sinó que ens hem fixat en la comoditat i facilitat a l'hora de programar.

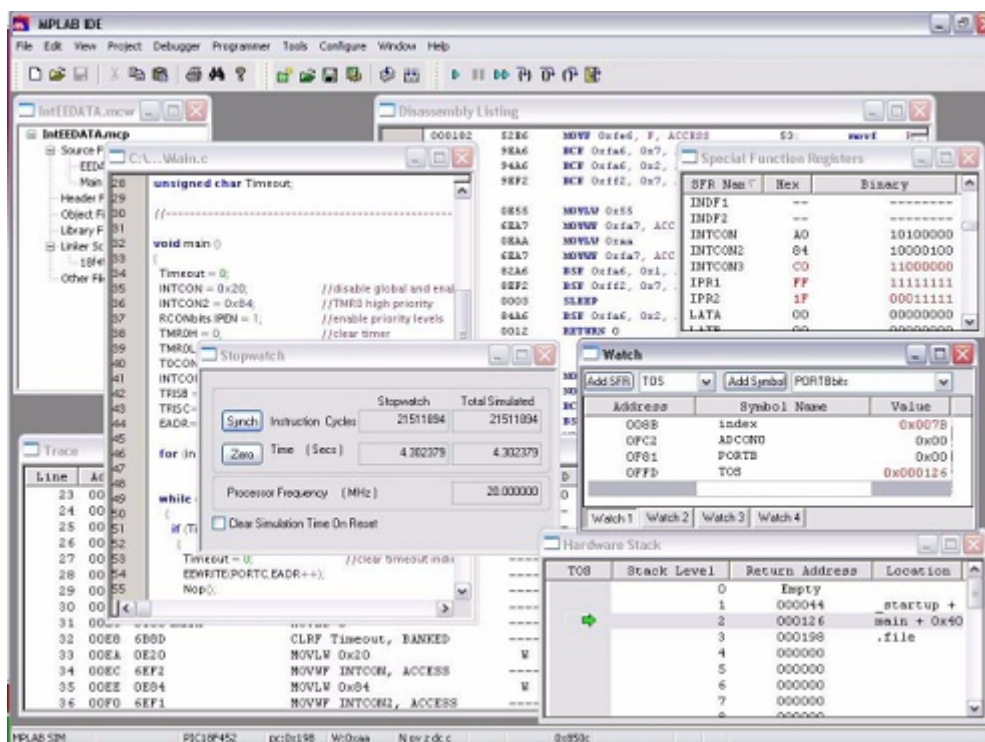


Figura 82: el entorn de programació MPLAB 8 de *Microchip*

Al fixar-nos exclusivament en la facilitat i la comoditat a l'hora de programar hem aconseguit simplificar l'elecció, pues l'opció de programar en llenguatge ensamblador queda directament descartada i sols ens queda l'opció d'utilitzar algun dels compiladors de C que hi han disponibles per a aquests microcontroladors. Com *Microchip* oferix en la seva *web* les ferramentes necessàries per a programar el *firmware* d'un sistema controlat per algun dels seus microcontroladors, no ens ho hem pensat ni un segon i hem utilitzat l'entorn de programació MPLAB 8.8x junt a la versió d'avaluació 3.31 del compilador de C30 de *Microchip* i les llibreries d'aplicació que va alliberar el 3/04/12.

Destacar que la versió d'avaluació del compilador és idèntica a la completa durant els primers 90 dies i, passat este temps, continua funcionant però amb les opcions d'optimització desconnectades, pel que per a fer una prova de concepte és una opció tan valida com la versió de pagament.

## 7- Descripció general dels components escollits

En les seccions anteriors hem anomenat els diferents components que utilitzarem en el nostre disseny, però en cap moment hem donat una descripció que ens pugui ajudar a entendre el seu funcionament ni el paper que tenen en el sistema pel que, per a evitar que el lector tinga que llegir-se les corresponents fulles de característiques per a conèixer-los, hem decidit fer una breu descripció dels mateixos i d'eixa manera facilitar la comprensió d'aquest treball.

### 7.1- El microcontrolador PIC24FJ256GB106

Tal i com hem vist en la secció corresponent, de totes les possibles opcions possibles per a encarregar-se de la part de control nosaltres hem triat el PIC24FJ256GB106. Aquest microcontrolador forma part de la família GB110 i disposa de les següents característiques:

- Encapsulat TQFP de 64 pins.
- 16 Milions d'Instruccions Per Segon @ 32MHz
- 256KB de memòria *Flash* de programa
- 16KB de memòria RAM
- Calendari i Rellotge de Temps Real (RTCC)
- 5 Temporitzadors de 16 bits, 2 dels quals es poden unir per a formar un temporitzador de 32 bits
- 9 Eixides compatibles amb la modulació per ample de polses (PWM)
- 3 Comparadors
- Convertidor A/D de 10 bits
- 4 UARTs amb control de flux, 3 ports SPI i 3 ports I2C
- 1 Port USB 2.0 *Full Speed* (12 Mbps) i compatible amb l'especificació OTG
- 1 Port paral·lel capaç de treballar com a mestre o esclau, el qual permet comunicar-se amb quasi qualsevol dispositiu.
- 1 Mòdul generador de CRC
- Fins 5 fonts externes d'interrupció
- *Watch Dog Timer*
- 29 Terminals de Entrada/Eixida re-assignables per *Software* als perifèrics (PPS<sup>36</sup>)
- Memòria de programa reprogramable per *Software*
- Ports d'entrada/eixida compatibles amb lògica de 3.3 i 5 Volts
- Reprogramable en circuit (ICSP)
- 1mA / MIP i 3 modes de funcionament de baix consum (100nA en mode *Sleep*)

Després de veure les seues característiques podríem pensar que aquest microcontrolador és més que suficient per a poder complir els nostres objectius, però l'ús d'una pantalla LCD de port paral·lel consumeix gran part dels terminals d'entrada/eixida, pel que després de connectar tots els elements del sistema al mateix, sols disposarem d'uns 15 terminals lliures i, d'eixos 15, sols 6 podran ser re-assignats per software a algun dels perifèrics que ha quedat lliure. Així que, sí no es poguera prescindir del *display* LCD, afegir alguna funció més al disseny seria un petit repte. A pesar d'aquest petit inconvenient, podem estar agraïts que aquests dispositius ens deixen configurar al nostre gust quins terminals seran els que utilitze un determinat perifèric en un determinat moment, ja que d'aquesta manera podrem utilitzar de forma més eficient els recursos del processador, de fet, sí no fos per aquesta característica, no seria possible poder substituir més avant el nostre PIC pel DSPIC del que parlarem en la següent subsecció. Això sí, hi ha un preu a pagar i és l'augment de complexitat del programa que executarà el nostre PIC, pel que hi ha que anar en compte a l'hora de realitzar canvis.

---

<sup>36</sup> PPS Periphral Pin Select, aquesta funció permet canviar la funció d'un terminal d'entrada/eixida en temps d'execució



Això des del punt de vista físic. Des del lògic hi han un parell de detalls a tenir en compte. El primer és la quantitat de RAM de la que disposem. Tot i que és una quantitat més que acceptable, sí en la fulla de característiques del sensor d'inèrcia s'hagués detallat el seu procés de calibratge, a l'hora de l'elecció del microcontrolador se li hauria donat un poc més de pes a aquesta característica pues, pels requisits temporals del calibratge d'un sensor MEMS, el consum de RAM es pot disparar i, o bé s'utilitza algun truc per a poder completar correctament el procés, o bé s'utilitza l'assignació de memòria dinàmica amb els riscos d'inestabilitat del sistema que això pot suposar. L'altre punt a tenir en compte són els *Sillicon Errata Issues*<sup>37</sup> de la família GB110. Tot i que actualment el document referent a aquestes errades proporcionat per *Microchip* conté al voltant de quaranta errors en el disseny *hardware* del microcontrolador, a nosaltres sols ens afecten quatre<sup>38</sup>:

- Mòdul USB: el regulador intern de 3.3V no regula. La solució és deshabilitar-lo per programa i alimentar el terminal corresponent amb 3.3 volts.
- Mòdul USB: el flag d'interrupció d'activitat de bus sols s'activa quan el microcontrolador està en mode *sleep*. La solució al problema depèn de com s'alimenta el sistema pel que aquest *bug* s'ha de tenir en compte sí es canvia el disseny del sistema ja que en el nostre cas, com el nostre sistema no utilitza el bus USB per a alimentar-se, sols hem d'evitar que el programa pose en suspensió el dispositiu, però sí passarem a alimentar el sistema des de aquest port, aleshores deuríem de modificar part del programa per a evitar un funcionament aleatori.
- Mòdul UART: la petició d'interrupció en cas d'un error en el mòdul (UERIF) es produeix amb retràs o bé no es produeix. Encara que aquest *bug* és exclusiu de la *versió de silicó* del nostre microcontrolador, no és un problema seriós sí practiquem una programació segura ja que la solució proporcionada pel fabricant no és una altra que llegir els *flags* d'error corresponents en cada lectura del registre de dades d'aquest mòdul.
- Mòdul UART: errors en els *flags* d'error del *buffer* FIFO. Aquesta errada també és exclusiva de la revisió de silicó del nostre microcontrolador pel que en cas d'utilitzar versions més modernes del mateix no estarem obligats a botar a la rutina d'atenció d'interrupció cada vegada que la UART rep un caràcter<sup>39</sup>, pel que podrem fer un ús més eficient d'aquest mòdul.

Òbviament sí volguérem afegir alguna funcionalitat al sistema o simplement ens interessés canviar algun aspecte del disseny inicial, deuríem de consultar la llista actualitzada d'errors proporcionada pel fabricant per a estalviar-nos possibles problemes.

De tots els recursos que ens dona el microcontrolador, els més útils en la nostra aplicació, van a ser els ports de comunicació digital ja que gràcies a ells podrem obtenir dades del sensor d'inèrcia i/o del GPS i, sí ho trobem convenient, deixar-los en una targeta micro SD o bé enviar-los pel port USB a un altre dispositiu. Per aquest motiu, qualsevol altre perifèric del PIC passa a segon pla i sols el mòdul del Relloge en Temps Real i els temporitzadors prenen alguna importància en determinades ocasions respecte als altres perifèrics disponibles.

### **7.1.1- Opcions passades i futures: PIC18F46J50 i DSPIC33EP256MU806.**

En el punt anterior comentàvem que durant el procés d'elecció de l'encarregat de gestionar el sistema s'havia considerat l'opció de triar un dispositiu de menys potència, el PIC18F46J50, i un altre de major potència, DSPIC33EP256MU806. La principal raó de no haver escollit cap dels dos és la potència: un per tenir-ne poca i l'altre per tenir-ne més de la necessària. Açò podria parèixer

<sup>37</sup> Errors en el disseny del *hardware* d'un circuit integrat

<sup>38</sup> Aquestes errades corresponen respectivament a les errades 9, 11, 12 i 13 del PDF PIC24FJ256GB, pàgines 5 i 6.

<sup>39</sup> La UART d'aquests PICS té un *buffer* FIFO pel que es pot fer que el *flag* d'interrupció s'active segons la quantitat de posicions lliures del *buffer*.

absurd, però no ho és ja que inclús l'excés de potència d'un microcontrolador pot ser un problema, doncs al tenir tants recursos al seu abast, s'ha de dedicar més temps al disseny del sistema i a la comprensió del seu funcionament per a, d'eixa manera, poder obtenir un funcionament més eficient del sistema. Així doncs, en el cas d'haver utilitzat aquest microcontrolador en el sistema passariem a controlar, a part del que ja tenim en el 24FJ256GB106, les següents funcionalitats extra:

- 1 Mòdul DSP amb 24KB de Flash i 4KB de RAM addicionals als del PIC.
- 15 Canals DMA
- 2 Mòduls QEI<sup>40</sup>
- 1 Mòdul DCI<sup>41</sup>
- 9 Temporitzadors de 16 bits, 8 dels quals es poden unir per a formar 4 temporitzadors de 32 bits
- 16 Eixides PWM<sup>42</sup>, 9 dels quals poden utilitzar-se per al control de motors PWM.
- 4 Comparadors
- 2 Convertidors A/D de 12 bits
- 4 UARTs amb control de flux, 4 ports SPI i 2 ports I2C

Com podem veure, aquest microcontrolador no sols arriba a doblar la quantitat d'algun dels perifèrics disponibles respecte a un PIC de la família GB110, sinó que n'afegeix altres que podrien resultar interessants a l'hora de controlar l'estabilitat d'una plataforma pues disposa d'un mòdul pel control de motors i un *Encoder* per Quadratura (QEI) el qual podria ser-nos molt útil pel control d'un sistema mecànic. Els canals DMA també ens podrien vindre molt bé per a alleugerir la carrega de treball del processador ja que, mentre els perifèrics podrien estar transmetent-li la informació a aquest mòdul, el microcontrolador podria estar processant els resultats dels càlculs proporcionats pel DSP intern, el qual podria utilitzar-se perfectament per a modelar els moviments de la plataforma, o bé dedicant-se al control d'un altre dispositiu, cosa que seria més que factible ja que passariem dels 16 milions d'instruccions per segon d'un PIC 24FJ als 60 que pot arribar a executar un DSPIC.

Tot el contrari passa amb el PIC de 8 bits ja que, tot i que és igualment vàlid per a la nostra aplicació, la seva potència i l'ajustat nombre de perifèrics i recursos dels que disposa no sols ens limita en un futur, sinó que, com ja veurem més avant, complica el disseny de l'aplicació. Així doncs, en cas d'haver utilitzat aquest microcontrolador, la llista de perifèrics i recursos quedaria de la següent manera:

- 64KB de memòria de programa i 3,7KB de RAM
- 2 UARTs i 2 Mòduls SPI/I2C amb possibilitat d'utilitzar DMA.
- 3 Temporitzadors de 16 bits
- Convertidor A/D de 10 bits i 13 canals
- 2 Comparadors
- 1 Port USB compatible amb l'especificació 2.0 però sense suport del mode OTG ni dels modes d'alta velocitat.

Com podem veure són uns recursos més que justos per a controlar un sistema de les nostres característiques ja que, com ja es va comentar, en cas de voler connectar algun dispositiu més, aquest deuria ser obligatòriament un dispositiu amb UART ja que els mòduls SPI estarien utilitzats pels sensors d'inèrcia o bé per un sensor i un lector de targetes SD, mentre que l'USB, al no poder treballar en un altre mode que no siga el mode dispositiu, no ens permetria assumir el control d'un altre dispositiu.

40 Quadrature Encoder Interface, *Encoder* per Quadratura.

41 DCI, *Data Converter Interface*: facilita la connexió amb *codecs* d'àudio, convertidors A/D o D/A, etc

42 *Pulse Width Modulation*, Modulació per Ample de Polses

## 7.2- El sensor d'inèrcia ADIS16400

El sensor d'inèrcia és sense cap dubte l'element més important del sistema ja que ell és el que ens proporciona tota la informació referent al moviment experimentat pel sistema, pues tot i que el GPS també ens dona informació que ens pot servir per a determinar el desplaçament experimentat, aquest depèn de la correcta comunicació amb els satèl·lits que puga tenir a la vista, pel que sí no hi ha prou satèl·lits a la vista, o simplement hi han interferències en l'ambient, no ens podrem fiar del mateix i, per tant, seria com no tenir res.

Inexplicablement, per a la complexitat que pot tenir un dispositiu el qual el mateix fabricant etiqueta com a ideal per a aplicacions de control d'estabilitat d'una plataforma, la fulla de característiques és molt simple i això és una cosa que s'aprecia ràpidament a l'obrir el PDF corresponent, doncs en menys de 20 fulles "explica" el funcionament d'un sensor que disposa de les següents característiques:

- Giroscopi triaxial amb la possibilitat de treballar en tres rangs:  $\pm 75^\circ/\text{segon}$ ,  $\pm 150^\circ/\text{segon}$  i  $\pm 300^\circ/\text{segon}$ , amb una precisió que va dels  $0.0125^\circ/\text{segon}$  en el rang de  $75^\circ/\text{segon}$  als  $0.05^\circ/\text{segon}$  del mode  $300^\circ/\text{segon}$ .
- Acceleròmetre triaxial capaç de mesurar acceleracions de  $\pm 18\text{G}$  amb una precisió de  $3.33\text{mG}$
- Magnetòmetre triaxial amb el que es poden arribar a mesurar camps magnètics de fins  $\pm 2.5\text{T}$  amb una precisió de tan sols  $0.5\text{mGauss}$ .
- Sensor de Temperatura
- Convertidor A/D de 12 bits
- Comportament programable: múltiples alarmes, freqüència de mostreig, filtrat, auto-test, auto-calibratge, funció dels terminals auxiliars, etc.
- Calibratge manual amb precisió de  $0.0125^\circ/\text{seg}$  pels giroscopis,  $3.3\text{mG}$  per als acceleròmetres i  $0.5\text{mGauss}$  per als magnetòmetres.
- Comunicació SPI a  $2\text{MHz}$  amb la possibilitat d'enviar de colp tots els registres del sensor utilitzant un únic procés de lectura.
- Mode baix consum
- Memòria *flash* (de baixa vida) per a emmagatzemar els registres importants i guardar les configuracions d'usuari com pot ser el calibratge dels sensors.

Però el pitjor de tot no és el fet de la poca informació proporcionada per la fulla de característiques, sinó que a més, conté errors, cosa que, per cert, pareix habitual en les *datasheets* del fabricant ja que buscant informació sobre el problema de comunicació que vam experimentar en l'inici del projecte, ens vam trobar amb una memòria d'un projecte de la Universitat Politècnica de Catalunya<sup>43</sup> en la qual els autors no van poder comunicar-se amb el seu sensor fins que no van detectar una serie d'errades en les especificacions del mode SPI utilitzat pel mateix. Però la cosa no acaba ací pues a part d'estes errades o de deixar-se coses pel camí, de vegades podem veure com no se'ls dona la importància que es mereixen certs detalls que posteriorment resulten ser vitals per al correcte funcionament del sensor, com és el cas de la calibratge del dispositiu o dels efectes del *random walk* en la seva eixida, i, quan no és això, com ells mateixa reconegueren per E-Mail, simplement és el detall d'assumir que l'usuari és un expert en la matèria i no necessita que li aclarisquen o exemplifiquen alguns conceptes/aspectes del funcionament de la IMU, i clar, davant eixa situació no queda una altra que dirigir-se al fabricant i preguntar-los... sí és que treballes en una empresa, perquè sí comets l'error de dir que eres estudiant universitari, et contesten amablement que no donen suport a les universitats<sup>44</sup>, cosa que resulta absurda perquè els qui teòricament utilitzaran en un futur els seus components seran eixos estudiants als que ara no donen suport.

43 Consultar l'entrada de la bibliografia: *Diseño y construcción de bus de datos y sensores para las practicas de NACC*

44 Les conversacions per eMail amb personal de *Analog* es troben en l'apèndix d'aquesta memòria.

## 7.2.1- Especificacions temporals del ADIS16400

Tot i que no és complicat comunicar-se amb un dispositiu que utilitza el bus SPI per a intercanviar informació amb l'exterior, per a fer-ho correctament és importantíssim, no sols complir les especificacions temporals proporcionades pel fabricant, sinó que a més, com ara veurem, s'han de tenir en compte alguns detalls.

| Parameter                               | Description                            | Normal Mode<br>(SMPL_PRD ≤ 0x09) |     |      | Low Power Mode<br>(SMPL_PRD ≥ 0x0A) |     |      | Burst Mode          |     |      | Unit |
|---|--|----------------------------------|-----|------|-------------------------------------|-----|------|---------------------|-----|------|------|
|   |  | Min <sup>1</sup>                 | Typ | Max  | Min <sup>1</sup>                    | Typ | Max  | Min <sup>1</sup>    | Typ | Max  |      |
| f <sub>SCLK</sub>                       | Serial clock frequency                 | 0.01                             |     | 2.0  | 0.01                                |     | 0.3  | 0.01                |     | 1.0  | MHz  |
| t <sub>STALL</sub>                      | Stall period between data              | 9                                |     |      | 75                                  |     |      | 1/f <sub>SCLK</sub> |     |      | μs   |
| t <sub>READRATE</sub>                   | Read rate                              | 40                               |     |      | 150                                 |     |      |                     |     |      | μs   |
| t <sub>CS</sub>                         | Chip select to clock edge              | 48.8                             |     |      | 48.8                                |     |      | 48.8                |     |      | ns   |
| t <sub>DAV</sub>                        | DOUT valid after SCLK edge             |                                  |     | 100  |                                     |     | 100  |                     |     | 100  | ns   |
| t <sub>DSU</sub>                        | DIN setup time before SCLK rising edge | 24.4                             |     |      | 24.4                                |     |      | 24.4                |     |      | ns   |
| t <sub>DHD</sub>                        | DIN hold time after SCLK rising edge   | 48.8                             |     |      | 48.8                                |     |      | 48.8                |     |      | ns   |
| t <sub>SCLKR</sub> , t <sub>SCLKF</sub> | SCLK rise/fall times                   |                                  | 5   | 12.5 |                                     | 5   | 12.5 |                     | 5   | 12.5 | ns   |
| t <sub>DF</sub> , t <sub>DR</sub>       | DOUT rise/fall times                   |                                  | 5   | 12.5 |                                     | 5   | 12.5 |                     | 5   | 12.5 | ns   |
| t <sub>SFS</sub>                        | $\overline{CS}$ high after SCLK edge   | 5                                |     |      | 5                                   |     |      | 5                   |     |      | ns   |
| t <sub>1</sub>                          | Input sync pulse width                 |                                  | 5   |      |                                     |     |      | 5                   |     |      | μs   |
| t <sub>2</sub>                          | Input sync to data ready output        |                                  | 600 |      |                                     |     |      | 600                 |     |      | μs   |
| t <sub>3</sub>                          | Input sync period                      | 833                              |     |      |                                     |     |      | 833                 |     |      | μs   |

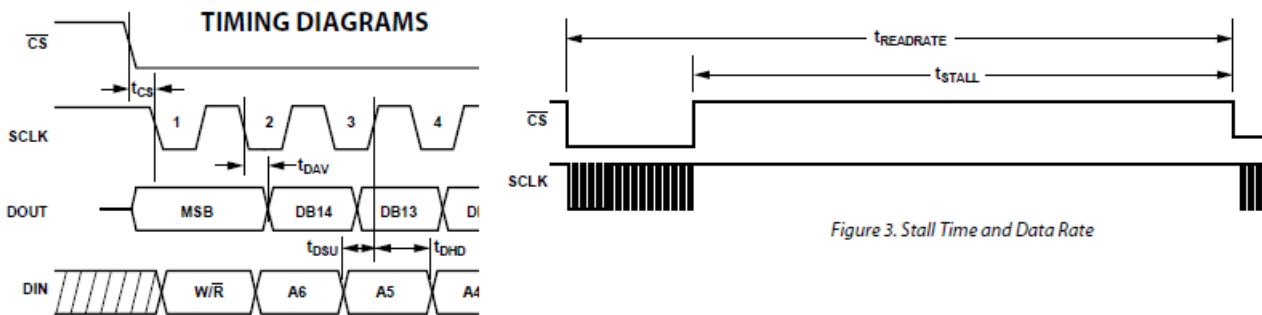


Figura 83: Especificacions temporals de la IMU ADIS 16400

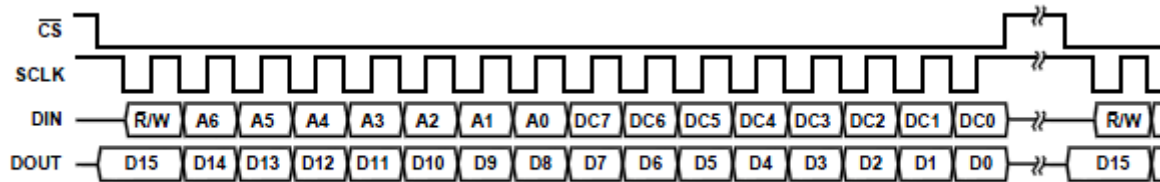
D'acord amb les dades proporcionades per la figura 83, hem de configurar la comunicació SPI de manera que el rellotge estiga inactiu en nivell alt i que les dades siguin vàlides quan el rellotge canvie d'inactiu a actiu. Això, en la nomenclatura SPI, equival al mode 3 (CPOL=CPHA=1), però si s'utilitza un PIC s'ha de tenir en compte que CPHA!=CKE pel que el corresponent bit del registre de control no ha de ser 1, sinó 0. De no no configurar correctament el mode de funcionament del bus, no sols no llegirem res de trellat, sinó que, com ja ens va passar a nosaltres al principi, el sensor pot arribar a perdre tota la configuració interna si en algun moment arriba a confondre una ordre de lectura amb una d'escriptura.

Un altre aspecte a tenir en compte és que la velocitat a la qual es pot accedir al dispositiu canvia depenent de si s'està accedint a ell en mode normal, en mode ràfega o de si està en mode baix consum, pel que ens hem de recordar de canviar la velocitat de rellotge cada vegada que canviem el mode de funcionament del dispositiu.

L'últim punt a tenir en compte és l'especificació temporal de la línia de *Chip Select* ja que una vegada s'ha llegit el sensor, s'ha d'esperar un temps igual o major a T<sub>STALL</sub> abans que el sensor pugui ser llegit de nou. El temps d'espera entre lectura i lectura també és diferent per a cada mode de funcionament pel que, si anem a utilitzar més d'un mode de funcionament en la nostra aplicació, ho hauré de tenir en compte. En la nostra IMU aquest temps d'espera serà d'un mínim de 9μs si estem en el mode normal, de 75μs en el mode de baix consum o 1/f<sub>SCLK</sub> si l'estem llegint en mode ràfega.

## 7.2.2- Format de la comunicació amb la IMU i estructura interna

Des del punt de vista lògic, la comunicació amb el dispositiu es fa utilitzant paquets de 16 bits. Com els ADIS funcionen com a dispositius esclaus dins del protocol SPI, aquests sols posaran informació en el bus de dades quan se'ls active i la senyal SCLK tinga una freqüència vàlida, per això haurem de distingir entre dos tipus de paquets: els que li enviem nosaltres i els que ell ens envia com a resposta.



### NOTES

1. Els Bits DOUT estan basats en l'anterior cicle de lectura

Figura 84: Estructura d'un paquet enviat pel dispositiu mestre

Tal i com podem observar en la figura 84, podem distingir tres parts en els paquets que li enviem:

- La primera part forma el bit més significatiu del paquet: R'/W. Amb un '0' s'indica al sensor que va a ser llegit, mentre que amb un '1' se li diu que va a escriure's una dada en un dels seus registres interns.
- La segona part està formada pels 7 bits següents a R'/W, és a dir, A6-A0, i especifiquen quin registre va a ser llegit o escrit.
- L'última part la formen els 8 bits menys significatius, D7-D0. Aquests bits contenen el valor que s'escriurà en el registre. Com la IMU sols es fixa en aquesta part quan va a ser escrit, l'usuari pot posar qualsevol valor en aquesta part quan vaja a llegir-lo. Independentment de si va a ser llegit o escrit, l'ADIS sempre contestarà la pròxima vegada que l'activem amb un paquet de 16 bits, així que es podria dir que per a efectuar una lectura, en realitat hem de fer dos lectures.

Aquesta manera de treballar fa que no importe si li estem demanant que ens envie la part alta o part baixa d'un registre, pues cada vegada que accedim a una direcció de memòria, l'ADIS sempre ens enviarà els 16 bits referents a la mateixa. En canvi no es pot dir el mateix de l'escriptura ja que escriure un valor en una direcció de memòria sols afectarà als 8 bits d'eixa direcció, pel que si volem escriure un registre sencer haurem de fer dos escriptures: una amb els 8 bits de la part baixa del registre i una altra amb els 8 bits de la part alta.

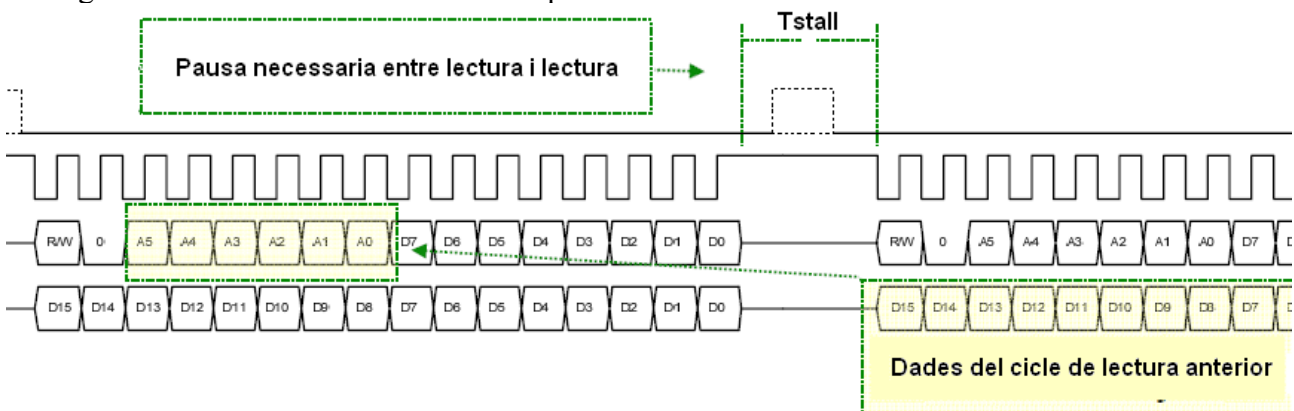


Figura 85: Exemple de comunicació amb l'ADIS: la contestació a una lectura és la resposta a l'anterior lectura

En el que respecta a l'estructura dels paquets que ens envien aquestes IMU's, com el format depèn del tipus de registre al qual s'està accedint, el que anem a fer, aprofitant que estan íntimament relacionats, és passar directament a parlar de l'organització interna d'aquesta família d'IMU's.

Observant la fulla de característiques podem veure que els registres interns ocupen els primers 76 bytes, la resta, excepte la posició corresponent al registre d'identificació, apareixen com a reservats, cosa que s'explica fàcilment al revisar la fulla de característiques de l'ADIS 16407, doncs en aquesta IMU, com hi ha un sensor més per eix, es desplaça varies posicions cap a baix el mapa de registres reduint d'aquesta manera el nombre de posicions lliures reservades. D'esta *datasheet* també obtenim la funció de dos registres reservats pel sistema: nombre del lot al que correspon el sensor i el seu nombre de serie. Siga com siga, dins del mapa de registres podem distingir tres tipus de registres:

- **Eixida dels sensors:** En el cas de l'ADIS 1640x comprenen des de la posició de memòria 0x02 a la 0x1D i ens proporcionen el valor registrat pel sensor corresponent. Excepte el que registra l'alimentació del sistema i el que guarda el resultat de la conversió A/D, tots estan codificats en complement a dos, pel que sí el bit més significatiu de la part que guarda la informació referent al mateix està a 1, aleshores la lectura serà negativa. Com el valor és un nombre sense dimensions, per a obtenir el valor real s'ha de multiplicar el valor resultant de la traducció anterior pel valor escala del registre. Per exemple, sí el registre ZAccel\_Out conté el valor 0x3ED3, al traduir-lo a decimal obtenim un valor de -300<sup>45</sup> el qual, al ser multiplicat pel factor escala, dona un valor de 9,9 m/s<sup>2</sup>, cosa lògica ja que aquest registre guarda el valor de l'acceleració perpendicular a la base del sensor.

| Acceleració | Decimal | Hex    | Binari              |
|-------------|---------|--------|---------------------|
| +18 g       | +5401   | 0x1519 | xx01 0101 0001 1001 |
| +6.667 mg   | +2      | 0x0002 | xx00 0000 0000 0010 |
| +3.333 mg   | +1      | 0x0001 | xx00 0000 0000 0001 |
| 0 g         | 0       | 0x0000 | xx00 0000 0000 0000 |
| -3.333 mg   | -1      | 0x3FFF | xx11 1111 1111 1111 |
| -6.667 mg   | -2      | 0x3FFE | xx11 1111 1111 1110 |
| -18 g       | -5401   | 0x2AE7 | xx10 1010 1110 0111 |

Figura 86: Exemple de traducció del registre XAccel\_Out extret del *datasheet* de l'ADIS 16407

Per últim destacar que es tracta de registres de sols lectura amb una grandària màxima de 14 bits. Per aquest motiu, quan s'envien pel bus SPI, els dos bits més significatius, D15-D14, indiquen respectivament sí hi han dades per llegir o sí s'ha produït algun error.

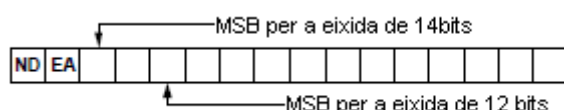


Figura 87: Exemple de l'organització d'un paquet enviat per l'ADIS, els bits ND/EA poden no estar presents

- **Compensació d'off-set:** Aquests registres estan posicionats immediatament després dels registres d'eixida dels sensors i, per a la nostra IMU, ocupen de la posició 0x1A a la 0x31 amb una longitud màxima de 14 bits. Gràcies a aquests registres és possible corregir els errors inicials de *bias* dels sensors<sup>46</sup> i, més avant, quan els efectes de la temperatura i el *random walk* comencen a ser evidents, compensar l'efecte dels mateixos sobre l'eixida dels sensors.

A l'igual que en el cas dels registres d'eixida dels sensors, el valor que es llig o s'escriu va representat en complement a dos i no té dimensions, pel que el valor real serà també el resultat de traduir aquest nombre a decimal i després multiplicar-lo per un factor escala que depèn del sensor a ajustar. La diferència està en que no cal traduir el seu valor per a treure-li trellat al mateix ja que bàsicament el seu ús es limita a sumar o a restar el valor especificat pel mateix al sensor sobre el que actua. A més, no serà necessari mirar el valor

45 El resultat ve de realitzar l'operació 0x3FFF-0x3ED3=0x12C, el qual en decimal equival a -300.

46 Consultar secció 3.5.1 Problemàtica dels sensors MEMS per a més informació sobre aquests errors.

dels dos primers bits del paquet ja que l'ADIS no afegeix els bits ND ni EA<sup>47</sup> quan envia el paquet amb el valor d'un registre d'aquest tipus.

Per a entendre açò tornarem a l'exemple del resultat de llegir el sensor ZAccl\_Out. Com hem vist, llegir aquest registre en condicions normals dona com a resultat un valor de -300, el qual, després de ser traduït, resulta en un valor que coincideix amb la força de la gravetat. Doncs bé, sí volguérem anular el efecte de la gravetat sobre aquest eix tindriem dos opcions: canviar l'orientació del sensor de manera l'eixida fos zero o bé, actuar sobre el registre d'*off-set* d'aquest sensor, pues al escriure el valor oposat a la lectura, és a dir, un valor de 300, ja no obtindrem el valor de la gravetat, sinó un valor nul ja que el ADIS, abans d'enviar el valor de ZAccl\_Out, ha sumat els valors dels dos registres i ens ha enviat el resultat. Això vol dir que, sí en l'estat que nosaltres considerem com a repòs, un sensor ens dona una eixida de 3 LSB's<sup>48</sup>, per a que eixa eixida reflectira correctament l'estat de repòs, el que deuríem de fer és carregar en el corresponent registre d'*off-set* el valor oposat: -3.

- **Registres del sistema:** Es tracta de l'últim grup de registres i comencen en la posició del registre GPIO\_CTRL, el qual ocupa les posicions 0x32 i 0x33<sup>49</sup> en el mapa de registres de l'ADIS 16400. Tots aquests registres s'utilitzen per a obtenir informació del sistema i definir el seu comportament. Com explicar el significat de tots aquests registres seria pràcticament repetir el que diu la fulla de característiques sols farem èmfasis en els que s'interactua més freqüentment:

-GLOBAL\_CMD, Global Commands: escriure un '1' en el bit especificat per la taula 14 del *datasheet* del ADIS16400 fa que el controlador intern del mateix execute eixa ordre, a excepció del comande de lectura en mode ràfega el qual s'activa quan no s'escriu res en el registre, o siga, quan s'envia 0x3E00. Les ordres no són molt variades, però gràcies a aquest registre podrem realitzar calibratges automàtics dels giroscopis, fer un *reset* per *software* o actualitzar la *flash* interna de l'ADIS, funció de la que no cal abusar ja que la quantitat d'escriptures que és capaç de suportar no és molt alta.

-MSC\_CTRL, Miscellaneous Control: Aquest registre és una espècie d'afegit al d'ordres del sistema pues permet activar la compensacions dels efectes de l'acceleració en l'eixida dels giroscopis o activar diversos tests de comprovació del sistema.

-SENS\_AVG, Rang Dinàmic i Filtrat Digital: El ADIS permet utilitzar tres rangs dinàmics diferents, pel que depenent de l'aplicació, sí no necessitem mesurar un canvi d'orientació gran en poc de temps i interessa tenir la major precisió possible, aleshores serà una bona idea utilitzar aquest registre per a canviar la resolució. En canvi, sí volem reduir el soroll a l'eixida dels sensors el que farem serà actuar sobre els tres bits menys significatius d'aquest registre ja que aquests defineixen el nombre d'etapes del filtre digital *Bartlett* incorporat en el sensor. Tot i que la fulla de característiques dona a entendre que hi han disponibles fins 128 (2<sup>7</sup>) etapes, en realitat, pel disseny intern d'aquesta IMU, sols se'ns permet utilitzar-ne 64 (2<sup>6</sup>). Com és d'esperar al realitzar un filtratge, l'ample de banda del sensor disminueix pel que, per a assegurar-nos que les dades són coherents, l'haurém de llegir, com a mínim, a la mitat de velocitat original. A més, també hem de tenir en compte el fet que cada rang dinàmic té unes especificacions mínimes de filtrat pel que sí canviem de rang dinàmic, l'ADIS ajustara automàticament la quantitat d'etapes de filtrat<sup>50</sup>.

47 ND és un *flag* que indica dades per llegir en un registre d'eixida mentre que EA indica la presència d'un error.

48 LSB: *Less Significant Bit* o bit menys significatiu. Les lectures dels sensors normalment s'especifiquen utilitzant aquesta nomenclatura.

49 En la fulla de característiques dels ADIS, quan es mostra l'adreça del registre, sols es mostra la part baixa d'aquest.

50 Per a més detalls sobre el filtrat i rang dinàmic es recomana la lectura de la nota d'aplicació AN-1041 del fabricant.

### 7.2.3- Calibratge dels diferents sensors de l'ADIS 16400

El procés de calibratge és un dels aspectes més importants a l'hora de treure-li tot el partit a una IMU de tecnologia MEMS ja que, degut a les característiques d'aquests dispositius<sup>51</sup>, l'eixida sempre oscil·la, inclús quan el dispositiu està en repòs. Per aquest motiu, en la primera posta en marxa de la IMU, s'ha de realitzar un calibratge dels acceleròmetres i giroscopis per a reduir l'error de *bias* i obtenir d'eixa manera una resposta el més aproximada a la realitat possible.

Sí hem examinat la fulla de característiques haurem vist que l'explicació d'aquest procés brilla per la seva absència, pues al situar-nos en la secció corresponent, sols ens conta per damunt l'efecte d'escriure sobre els registres d'*off-set* i, sí continuem llegint, sols veurem referències a un parell de calibratges automàtics exclusius dels giroscopis: un és el *Autonull Command* o orde de calibratge ràpid, mentre que l'altre correspon a l'ordre *Precision Autonull Command*. Siga com siga, aquests calibratges sols són una solució temporal vàlida per als giroscopis pel que, per a no repetir el que es diu en el *datasheet* i en la nota d'aplicacions AN 1041, no les tractarem en aquesta memòria.

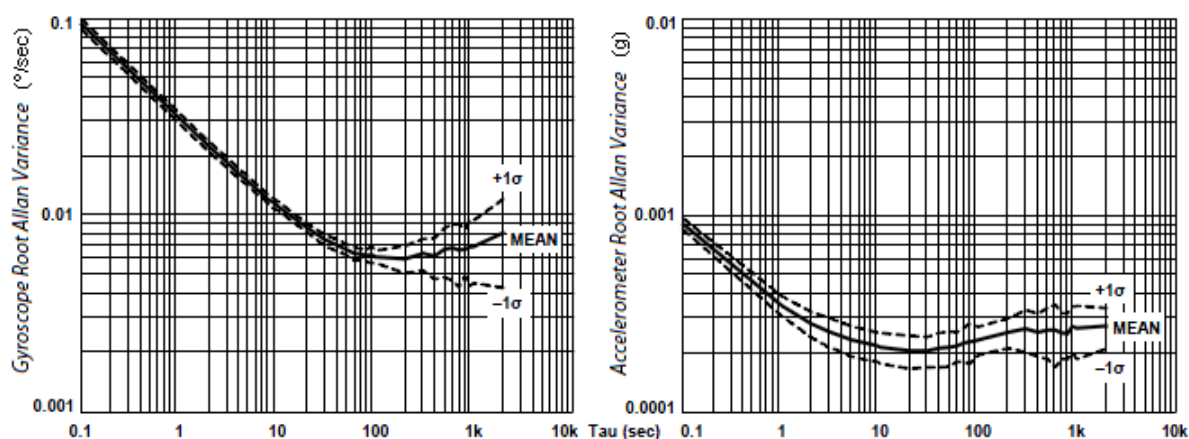


Figura 88: Evolució de la variances al llarg del temps

Per a realitzar de manera òptima aquest procés ens hem de fixar en les gràfiques proporcionades pel fabricant que ens mostren l'evolució de la variances d'*Allan*<sup>52</sup> al llarg del temps. En el cas de la nostra IMU, tenim les gràfiques mostrades per la figura 88.

Com podem observar en les mateixes, a mesura que augmenta el temps, la variances de *Allan* disminueix fins que arriba un punt en el que arriba a ser pràcticament constant per a després tornar a augmentar lentament. Pues bé, eixe període de temps en el que la variances té un valor estable és la quantitat de temps que durarà el calibratge del sensor en qüestió. És a dir, sí tenim tres acceleròmetres amb la variances de la figura 88, aleshores necessitarem uns 25 segons per acceleròmetre per a completar el calibratge d'aquests sensors, mentre que en el cas dels giroscopis necessitarem uns 100 segons per eix.

Cal destacar que, durant eixe temps, hem de prendre tantes mostres per segon com ens entregue el sensor a la seua velocitat de mostreig òptima. Açò vol dir que, sí no hem seguit les recomanacions del fabricant i no treballem la freqüència de mostreig per defecte del ADIS, abans d'iniciar el procés, haurem de restaurar aquesta freqüència al valor original. Aquest valor és d'unes 819 mostres per segon en les IMUs ADIS i implica que, per a calibrar un giroscopi, podem necessitar uns 160KB<sup>53</sup> de memòria, una quantitat que no està al abast de tots els microcontroladors.

<sup>51</sup> Ens referim al *Random walk* i als errors de *Bias* dels que ja hem parlat en la secció 3.5.1 Problemàtica dels sensors...

<sup>52</sup> La variances de *Allan* és un mètode d'anàlisi d'una seqüència de temps per tal d'extreure el soroll intrínsec al sistema com una funció de la mitja del temps

<sup>53</sup> El resultat ve de multiplicar el nombre de mostres per segon, la grandària de la dada proporcionat pel sensor i el període de temps que s'esta prenent mostres:  $819 \times 2 \times 100 = 163800$  bytes  $\approx$  160KB.



Independentment de sí el nostre sistema compleix els requisits de memòria, podria pensar-se que el procés de calibratge d'aquests sensors és tan simple com llegir 819 mostres per segon durant el temps determinat per la variança del sensor, realitzar la mitja de totes les mesures i, finalment, carregar el valor oposat de la mateixa en el registre d'*off-set* corresponent. Però en realitat eixa és la idea bàsica del procés ja que els únics sensors que poden ser calibrats tal qual sense la intervenció d'una persona són els giroscopis.

### 7.2.3.1- Calibratge: automàtic o manual?

A la vista de la gran quantitat de temps necessari per a realitzar el calibratge manual dels acceleròmetres i dels giroscopis és precís preguntar-se: Quin mètode de calibratge utilitzem? Manual o automàtic? Per a respondre més fàcilment a la pregunta compararem els dos mètodes:

| Calibratge manual                     | Calibratge automàtic                              |
|---------------------------------------|---|
| Necessita la participació de l'usuari | No necessita la intervenció de l'usuari           |
| Procés lent però de major precisió    | Ràpid però amb baixa precisió                     |
| Requisits de RAM relativament elevats | Requisits de RAM moderat                          |
|                                       | Sols es pot utilitzar amb els giroscopis          |
|                                       | Escriu en la <i>flash</i> de la IMU al finalitzar |

Taula 16: Característiques dels mètodes de calibratge manual i automàtic.

En un principi es podria pensar que la participació d'una persona en el procés va a ser determinant a l'hora de triar el tipus de calibratge, però en la pràctica, depenent de l'aplicació i de sí necessitem calibrar tots els sensors de la IMU, pot no tenir importància. Com veurem després, ens és indiferent que una persona participe en el procés, pel que a l'hora d'elegir el tipus de calibratge ens hem fixat en la precisió del procés i en un detall que a la llarga pot importar: l'escriptura en la memòria *flash* del sensor inercial. Per què li donem importància a aquest fet? Principalment perquè el fabricant d'aquesta IMU recorda varies vegades en la fulla de característiques que la vida útil de la memòria *flash* de la IMU no és elevada pel que no és recomanable abusar de l'escriptura en ella. Per aquestes dos raons, hem preferit utilitzar el calibratge manual ja que, després de tot, sols es “perd” el temps calibrant en el primer inici del sistema i els requisits de RAM no són un problema sí compliquem un poc la corresponent rutina de programa encarregada del calibratge.

### 7.2.3.2- Calibrant els giroscopis

Dels tres sensors diferents disponibles en l'ADIS 16400, els giroscopis són els únics que no necessiten la intervenció de l'usuari en el procés de calibratge, per eixe motiu, són els sensors que menys complicacions presenten a l'hora de ser calibrats ja que bàsicament el procés consisteix en:

- 1.- Deixar en repòs el sistema sobre una superfície totalment estable com puga ser una taula.
- 2.- Escriure 0x01 en el registre SMPL\_PRD per a que la IMU envie 819 mostres per segon.
- 3.- Escriure en la part alta del registre SENS\_AVG el valor 0x04 per a configurar el rang dinàmic de la IMU a 300°/seg.
- 4.- Escriure 0x02 en la part baixa del registre SENS\_AVG per a establir en 4 el nombre d'etapes de filtrat.
- 5.- Llegir durant 90-100 segons les 819 mostres que ens envia per segon la IMU.
- 6.- Una vegada capturades totes les mostres, realitzar la mitja, invertir el valor calculat, traduir-lo a complement a 2 i escriure'l al sensor.

Després dels quasi 5 minuts que dura el calibratge el dispositiu estarà preparat per a treballar de forma més exacta, cosa que podrem comprovar fàcilment deixant el sistema en repòs i observant com l'eixida dels giroscopis ja no oscil·la entre  $-1^\circ/\text{seg.}$  i  $1^\circ/\text{seg.}$  com passava abans del calibrar-los, sinó que ara ho fa al voltant de valors més petits tal i com s'observa en la figura 89.

|            |          |       |       |       |            |          |       |       |       |
|------------|----------|-------|-------|-------|------------|----------|-------|-------|-------|
| 10-09-2012 | 14:58:28 | 0.100 | 0.700 | 0.050 | 10-09-2012 | 10:46:37 | 0.100 | 0.350 | 0.000 |
| 10-09-2012 | 14:58:30 | 0.500 | 0.800 | 0.050 | 10-09-2012 | 10:46:39 | 0.100 | 0.100 | 0.000 |
| 10-09-2012 | 14:58:32 | 0.650 | 0.050 | 0.350 | 10-09-2012 | 10:46:41 | 0.100 | 0.100 | 0.350 |
| 10-09-2012 | 14:58:34 | 0.600 | 0.150 | 0.150 | 10-09-2012 | 10:46:43 | 0.100 | 0.300 | 0.500 |
| 10-09-2012 | 14:58:36 | 0.650 | 0.050 | 0.050 | 10-09-2012 | 10:46:45 | 0.000 | 0.050 | 0.100 |
| 10-09-2012 | 14:58:38 | 0.800 | 0.050 | 0.300 | 10-09-2012 | 10:46:47 | 0.050 | 0.300 | 0.350 |
| 10-09-2012 | 14:58:40 | 0.550 | 0.150 | 0.350 | 10-09-2012 | 10:46:49 | 0.250 | 0.100 | 0.100 |
| 10-09-2012 | 14:58:42 | 0.500 | 0.050 | 0.250 | 10-09-2012 | 10:46:51 | 0.000 | 0.300 | 0.000 |
| 10-09-2012 | 14:58:44 | 0.350 | 0.050 | 0.400 | 10-09-2012 | 10:46:53 | 0.100 | 0.250 | 0.100 |
| 10-09-2012 | 14:58:46 | 0.300 | 0.350 | 0.250 | 10-09-2012 | 10:46:55 | 0.100 | 0.200 | 0.200 |

Figura 89: A l'esquerra giroscopi per calibrar, a la dreta giroscopi una vegada calibrat.

### 7.2.3.3- Calibrant els acceleròmetres

El procediment a seguir per a calibrar els acceleròmetres depèn de l'ús que anem a fer d'ells, és a dir, depèn de l'aplicació. Estrictament parlant i seguint les suggerències publicades per *Analog Devices* en la fulla de característiques de l'ADIS 16407, els passos a seguir són aquests:

- 1.- Posicionar el sistema de manera que experimente una força de  $+1G$  en l'eix a calibrar.
- 2.- Configurar el sensor inercial per a que ens envie 819 mostres per segon escrivint el valor `0x0001` en el registre `SMPL_PRD`.
- 3.- Escriure `0x04` en la part alta del registre `SENS_AVG` per a configurar el rang dinàmic de la IMU a  $300^\circ/\text{seg.}$
- 4.- Escriure `0x02` en la part baixa de `SENS_AVG` per a que les etapes de filtrat siguin 4.
- 5.- Llegir durant uns 30 segons les 819 mostres que ens envia per segon la IMU.
- 6.- Una vegada capturades totes les mostres, realitzar la mitja i guardar-la en una variable.
- 7.- Posicionar el sistema de manera que experimente una força de  $-1G$  en l'eix a calibrar.
- 8.- Repetir els passos 5 i 6.
- 9.- Restar el valor mig obtingut quan l'acceleròmetre estava experimentant la força de  $-1G$  al valor mig resultant d'experimentar una força de  $+1G$ . Al resultat de la resta se li canvia el signe, es calcula el seu equivalent en complement a 2 i el resultat s'escriu en el corresponent registre d'*off-set* de la IMU.

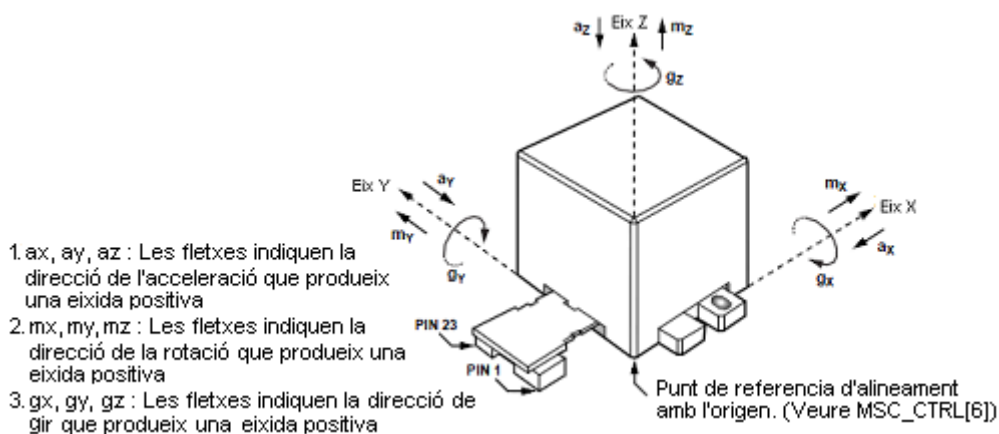


Figura 90: Distribució dels eixos de l'ADIS 16400.

Per a fer-nos una idea de com fer que els acceleròmetres experimenten una força de +1G o -1G, fixem-nos en la figura 90. Com podem veure, sols cal col·locar la IMU de manera que quede totalment perpendicular a la superfície horitzontal de la plataforma a controlar i el signe de l'eixida siga el del sentit de la força a mesurar. Per exemple, si volem calibrar l'acceleròmetre de l'eix X, primer girarem el sensor 90° a la dreta, capturarem 30 segons de lectures, calcularem la mitjana dels valors positius i repetirem el procés amb els valors negatius, sols que girant la IMU 90° a la esquerre com es veu en la figura 91.

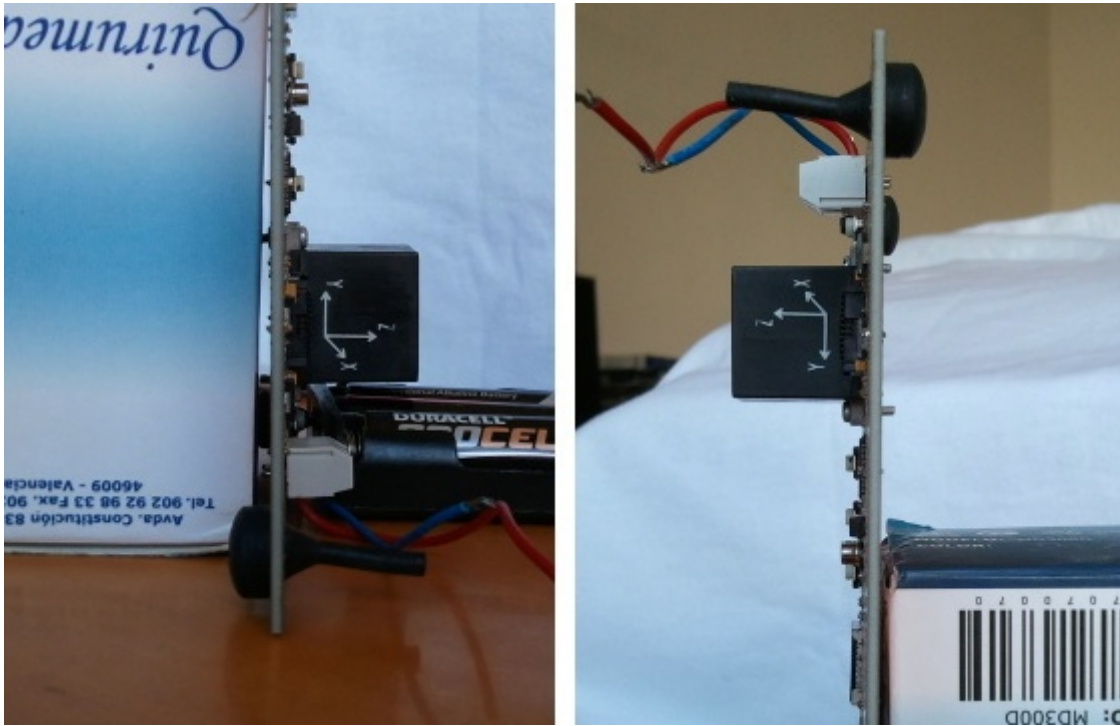


Figura 91: A l'esquerre el sensor orientat de manera que experimenta una força de +1G en l'eix X, a la dreta de -1G

El cas de l'eix Y és similar al de l'eix X, pel que per a mesurar +1G situarem el sensor de manera que la seva base ens done l'esquena, mentre que per a mesurar -1G el col·locarem de forma que la base ens mire. En canvi, per al eix Z, si inicialment la base del sensor queda totalment paral·lela a la superfície terrestre, sols haurem de girar el sensor 180° de manera que la part de dalt del encapsulat es recolze sobre la superfície de la plataforma a controlar per a que experimente una força de +1G, doncs en la posició inicial, l'acceleròmetre d'aquest eix ja està experimentant una força de -1G.

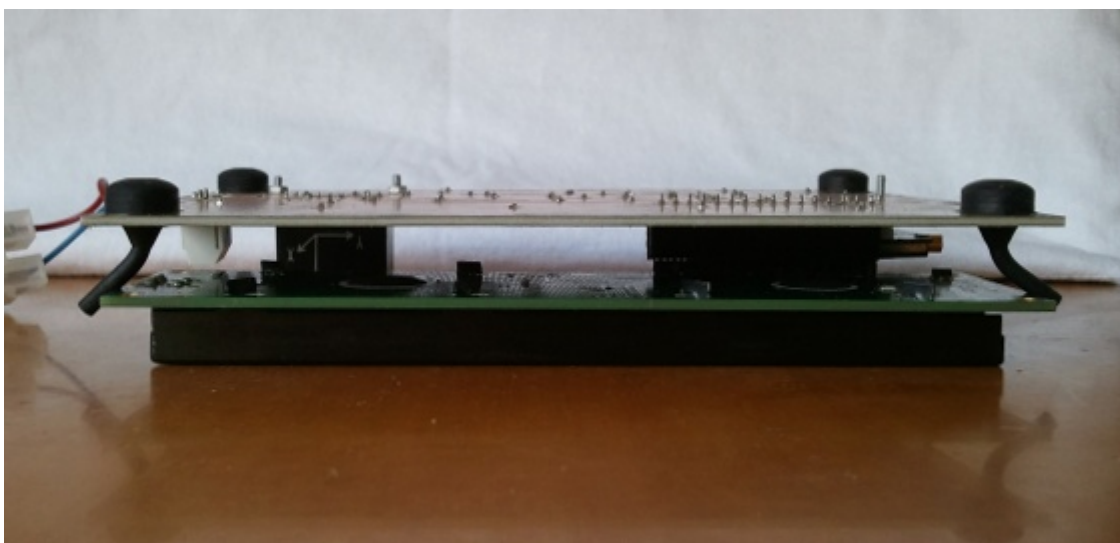


Figura 92: Sensor orientat de manera que experimenta una força de +1G en l'eix Z

Tot i que per a realitzar tot aquest procés, sí el sistema no és totalment autònom, és necessària la intervenció d'una persona per a posicionar el sistema d'acord en l'eix a calibrar, no sempre és recomanable la seva participació perquè en realitat, quan calibrem un sensor d'inèrcia, el que estem fent és establir quina posició en concret del sistema va a ser la que done una lectura 0, pel que sí la nostra intenció és detectar moviment, com ja vam insinuar al introduir els acceleròmetres, no interessa inclinar el sensor per a que experimente les forces de  $\pm 1G$  ja que de fer-ho estarem detectant, almenys en un eix, una acceleració no nul·la corresponent a la força de la gravetat que, de no tenir-se en compte, pot fer pensar al sistema que està experimentant un moviment en eixe eix quan en realitat el sistema continua en repòs o desplaçant-se a velocitat constant.

#### 7.2.3.4- Calibrant els magnetòmetres

El calibratge dels magnetòmetres és la part més obscura d'aquesta IMU ja que el fabricant es limita a explicar per damunt els registres involucrats sense arribar a entrar en detalls del procés. Aquesta falta de detall fa pensar que no és necessari calibrar aquests sensors o que amb capturar durant uns segons l'eixida dels mateixos i després calcular el valor oposat a la mitja és suficient, però en la pràctica és necessari una calibratge en condicions sí volem conèixer o mantenir l'orientació, o inclús corregir els errors de *bias* produïts per la temperatura en l'eixida dels giroscopis. De fet, esta és una de les raons per les que vam triar l'ADIS16400 i per les que les Unitats de Mesura Inercial triaxials solen incorporar també un magnetòmetre en cada eix.

Bàsicament, el procés de calibratge d'un magnetòmetre consisteix en rotar-lo un mínim de  $360^\circ$  i després representar els valors registrats sobre els eixos X i Y. Idealment la representació obtinguda hauria de ser un cercle centrat en (0, 0) com el de la figura 93, però en la pràctica el resultat que s'obté sol ser una el·lipsoide que no sempre estarà centrada en l'origen de coordenades.

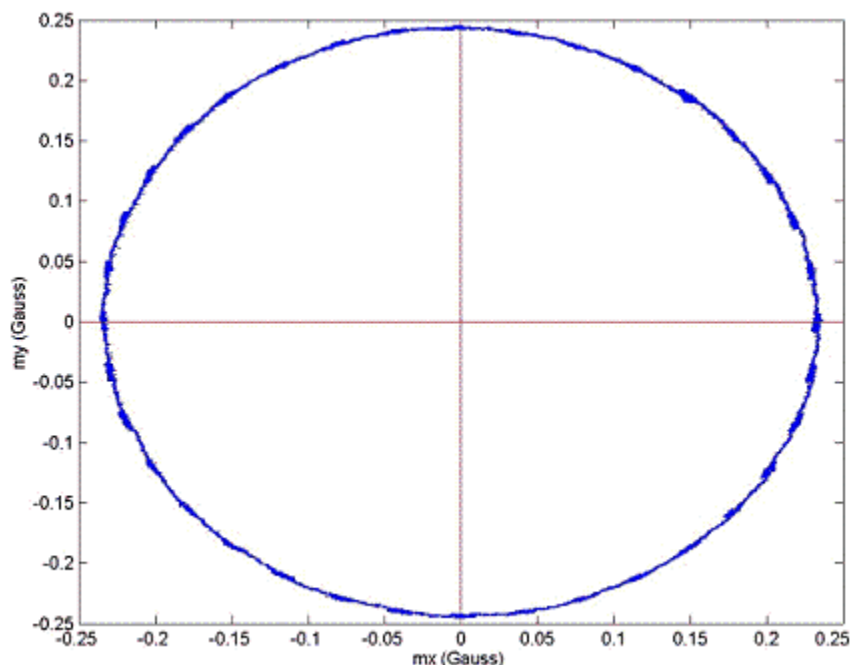


Figura 93: Representació gràfica ideal de les lectures registrades per un magnetòmetre durant el seu calibratge.

Aquesta deformació de la figura obtinguda i el possible desplaçament de la mateixa respecte a l'origen es deu a les influències magnètiques externes del sistema. Aquestes influències normalment es classifiquen en dos tipus: distorsions de tipus *Hard-Iron* i distorsions de tipus *Soft-Iron*.

Les distorsions de tipus *Hard-Iron* són produïdes per materials que produeixen un camp magnètic que s'afegeix al terrestre. Degut a la seva natura, afegeixen un valor constant al camp mesurat pel magnetòmetre i, per això, la figura resultant de representar les lectures del magnetòmetre durant el calibratge no està centrada en (0, 0). Com a exemple d'aquesta situació tenim els *smartphones* actuals. Molts d'aquests telèfons disposen d'un magnetòmetre pel que la lectura del mateix es veurà influenciada pel camp magnètic constant creat per l'imant de l'altaveu i haurà de ser corregida.

Però això no vol dir que les distorsions *Hard Iron* es limiten exclusivament a les interferències magnètiques que estan presents en el sistema de forma constant, sinó que a més, també inclouen els camps magnètics constants que puguen influir de sobte l'eixida del magnetòmetre. Per a veure més clar açò sols cal tornar a l'exemple del *smartphone* i les bases que s'utilitzen amb el sistema de mans lliures del cotxe. Per norma general aquests suports s'enganxen magnèticament al tauler de control del cotxe, pel que l'imant present en la base estarà produint una distorsió de tipus *Hard-Iron* sobre el magnetòmetre del telèfon. L'efecte d'aquest imant deuria de ser corregit pel *firmware* del telèfon a l'hora d'utilitzar-lo com a navegador, però també podria utilitzar-se sense problemes per a saber si el telèfon continua en la base o no i, depenent d'això, connectar o desconnectar el mòdul *bluetooth* intern.

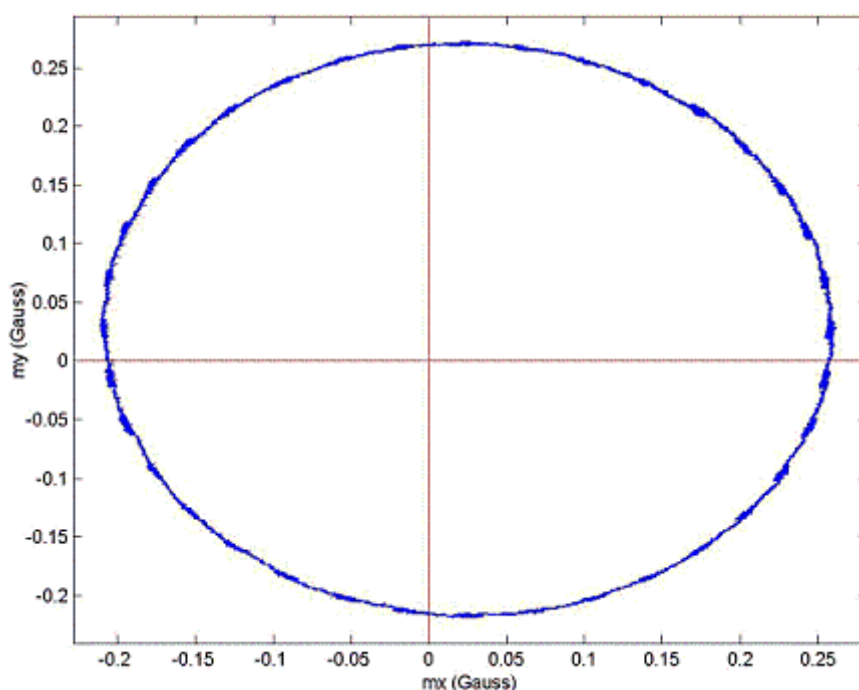


Figura 94: Representació gràfica del efecte de la distorsió *Hard-Iron* sobre un magnetòmetre

Per a compensar els efectes d'aquesta distorsió sobre les lectures del magnetòmetre hi ha prou amb rotar 360° el sensor i després calcular la mitjana del valor màxim i mínim per a cada eix, o el que és el mateix, calcular la distància des del punt (0, 0) al centre de la circumferència utilitzant les següents equacions:

$$\alpha = \frac{(x_{\max} + x_{\min})}{2}, \beta = \frac{(y_{\max} + y_{\min})}{2}$$

En les distorsions *Hard-Iron* aquests valors seran per norma general constants i no dependran de l'orientació ni de la posició de la plataforma de sensat, pel que hi haurà prou amb calcular-los una vegada, cosa que com ara veurem no passa en les distorsions *Soft-Iron*.

En el que respecta a les distorsions *Soft-Iron*, estes també afegeixen un camp magnètic al terrestre, sols que aquest camp extra no és constant i la seva intensitat depèn de l'orientació del material que el produeix respecte al camp magnètic de la terra. Aquesta diferència fa que aquest efecte no pugui ser compensat amb una simple constant com en el cas de les distorsions *Hard-Iron*, doncs aquesta perturbació transforma la circumferència ideal en una el·lipse. Per aquest motiu, com s'utilitzaran equacions matemàtiques més complicades, la carrega computacional augmentarà tant que, de vegades, serà més efectiu des del punt de vista del cost econòmic i de l'eficiència, eliminar els materials *Soft-Iron* que pugui hi haure al voltant del sensor, sempre clar, que això siga possible.

Així doncs, per a eliminar els efectes de les distorsions *Soft-Iron*, primer eliminarem les distorsions *Hard-Iron* i després, una vegada tinguem una el·lipse centrada en (0, 0) i rotada  $\theta$  graus respecte a l'eix X com la de la figura 95, calcularem l'angle de rotació  $\theta$ .

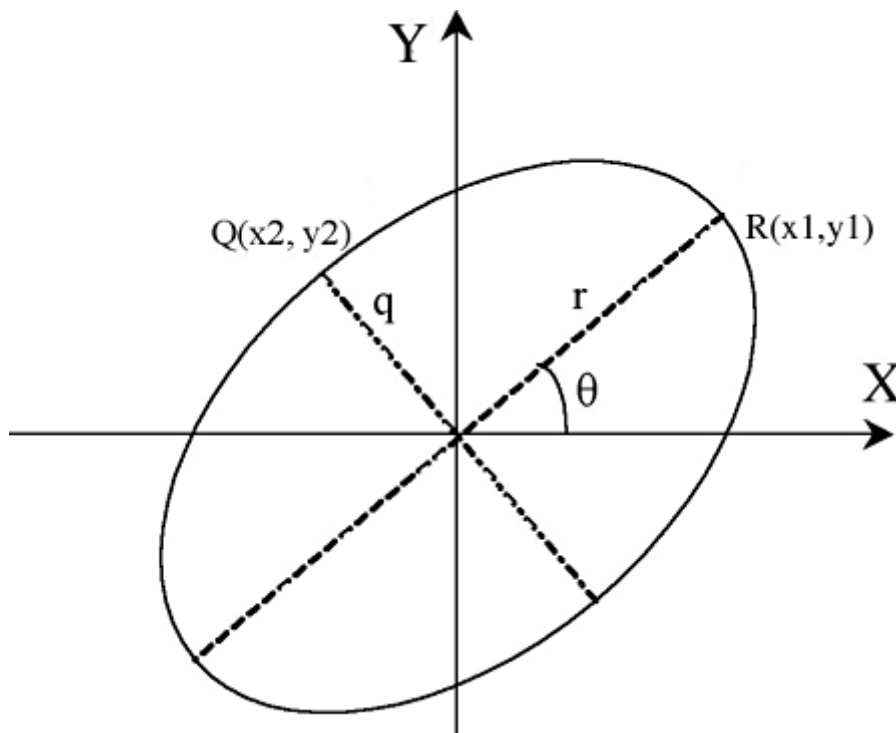


Figura 95: El·lipse centrada en (0, 0) i de rotació  $\theta$  producte de la distorsió *Soft-Iron*.

El valor de  $\theta$  el podem obtenir de l'equació:

$$\theta = \arcsin\left(\frac{y_1}{r}\right), \quad r = \sqrt{(x_1)^2 + (y_1)^2}$$

Sí tenim lectures prou netes podem deduir el valor de  $r$  calculant la magnitud de la mostra de cada punt  $i$ , aleshores, identificar el màxim d'aquests valors per a, d'eixa manera, obtenir les coordenades  $R(x_1, y_1)$ . Després de trobar aquest punt repetirem el procés per a obtenir la posició de  $Q(x_2, y_2)$ , però esta vegada utilitzant el valor mínim. Quan tinguem calculat el valor d'aquests dos punts podrem obtenir el valor de  $\theta$ , el qual servirà per a aplicar la matriu de rotació al vector del magnetòmetre:

$$R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}, \quad v_1 = Rv$$

Després d'aplicar la matriu de rotació, com podem veure en la figura 96, tindrem l'el·lipse alineada amb els dos eixos, pel que el següent pas serà escalar l'eix X de manera que l'el·lipse s'assembla més a una circumferència.

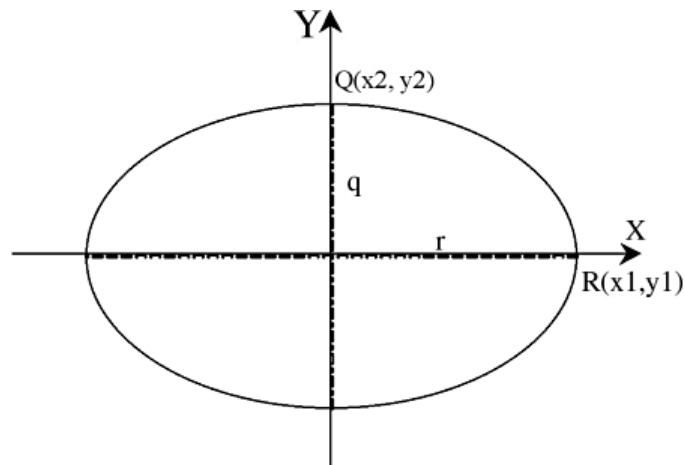


Figura 96: El·lipse totalment alineada després de rotar el vector.

El valor del factor escala  $\sigma$  és la divisió de la longitud de l'el·lipse en l'eix X per la de l'eix Y:

$$\sigma = \frac{q}{r}$$

Com es dedueix de l'expressió, cada valor x del magnetòmetre es divideix per aquest factor escala per a produir el cercle que busquem. Finalment, després de l'escalat, s'ha de tornar a rotar les lectures a la seva posició original utilitzant el valor negatiu de  $\theta$  per a compensar totalment les distorsions *Soft-Iron*.

Tot i que hem simplificat l'explicació del procés, queda patent que, matemàticament parlant, la compensació dels efectes *Soft-Iron* és complexa. Però la complexitat no acaba ací ja que en esta explicació sols hem considerat dos eixos de mesura, pel que sí volem calibrar un magnetòmetre de tres eixos, la dificultat augmenta pues passarem a parlar de circumferències i el·lipsoides.



Figura 97: moviment de calibratge ràpid del magnetòmetre d'un *smartphone*

A més, hem de tenir en compte que el procés no és exclusivament matemàtic sinó que també té una part pràctica. Açò ens obligarà a utilitzar un objecte del que no disposem: una plataforma rotatòria blindada magnèticament per a girar 360° el sensor. Una altra possibilitat seria utilitzar el mateix procediment que utilitzen els *smartphones* per a calibrar els seus magnetòmetres, però no sols perdriem exactitud, sinó que implicaria complicar la part del programa encarregada d'aquest procés, per això nosaltres no hem calibrat els magnetòmetres.

## 7.2.4- Exemple del procés de lectura de l'ADIS16400

Una vegada hem calibrat els diferents sensors de l'ADIS16400 toca comprovar que tot està en ordre i podem accedir a ell sense problemes. Al ser un dispositiu compatible amb el bus SPI el procés de lectura/escriptura és realment fàcil, ja que hi ha prou amb configurar el mòdul SPI del microcontrolador per a que treballi en mode 3 i seguir en ordre els passos corresponents per a cada acció. Hem de tenir en compte que el ADIS16400 és un dispositiu de 16 bits, pel que sí intentem comunicar-nos amb ell mitjançant un mode de transmissió de 8 bits, tot i que no hi haurà cap problema, el procés canviarà lleugerament.

Així pues, sí volem llegir la IMU en mode 16 bits hem de:

- 1.- Desplaçar 8 bits a l'esquerra l'adreça a llegir
- 2.- Sí anem a llegir l'eixida d'un sensor, esperar a que el port d'entrada connectat al terminal *Data Ready* (DIO1) del ADIS es posi en nivell alt.
- 3.- Baixar a '0' la línia de *Chip Select*.
- 4.- Carregar el resultat del pas 1 en el *buffer* d'eixida del microcontrolador.
- 5.- Esperar a que es buide el *buffer* del microcontrolador, és a dir, a que acabi de transmetre.
- 6.- Llegir el *buffer* del microcontrolador i descartar la dada llegida.
- 7.- Pujar a '1' la línia de *Chip Select* durant un mínim de 9 microsegons.
- 8.- Repetir els passos anteriors des de el pas 2 i guardar el valor llegit del *buffer* d'entrada.
- 9.- Esperar a que el terminal *Data Ready* de l'ADIS torne a nivell baix, especialment sí hem configurat el bus SPI a freqüències majors de 1MHz, sí no volem llegir dos vegades el mateix valor.

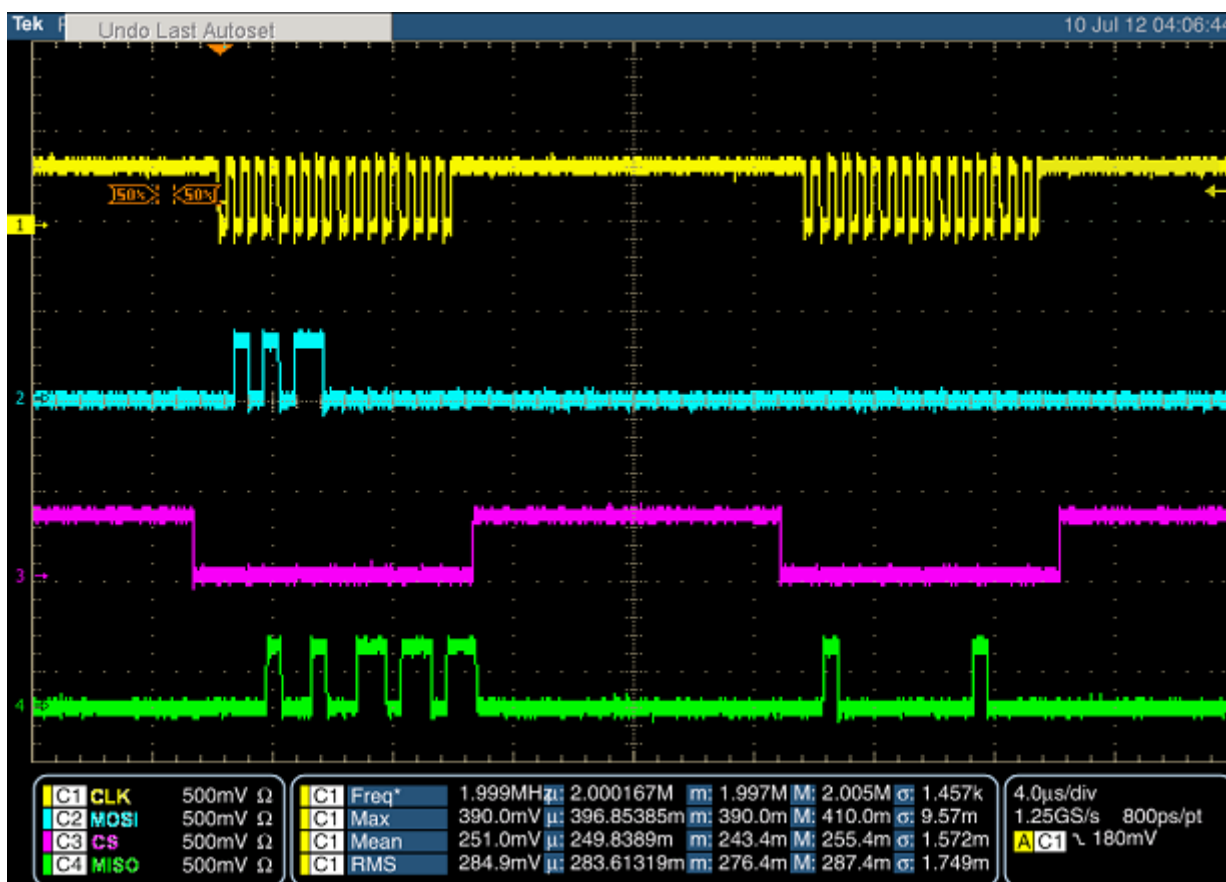


Figura 98: Lectura del registre d'identificació de l'ADIS16400.

En la figura 98 podem veure representat gràficament l'intercanvi d'informació que es produeix en el procés anterior. En aquest exemple hem optat per llegir el registre d'identificació del dispositiu ja



que suposadament és dels pocs que coneguem amb total certesa el seu valor. Per a la representació gràfica de les senyals hem triat el color groc per a la línia de rellotge (CLK), el blau clar per a l'eixida de dades del microcontrolador (MOSI), el fúcsia per a la línia de *Chip Select* i el verd clar per a l'entrada de dades del microcontrolador (MISO). També ens les hem arreglat per a que, a una freqüència de rellotge de 2MHz, es visualitzes el cicle complet de lectura.

Tal i com s'observa en eixa figura, tan prompte com posem a nivell baix la línia de *Chip Select* i enviem la dada 0x5600 pel bus SPI, el ADIS comença a enviar una dada que per a nosaltres en eixe moment no té cap importància ja que no està relacionada amb el registre al que accedirem, pel que esta vegada la deixarem passar, però no ho farem en la segon lectura que realitzem, és a dir, quan en el nostre exemple enviem el valor 0, ja que sí descodifiquem la dada que ens està enviant en eixe moment la IMU veurem que correspon al valor 0x4100, és a dir, al valor decimal 16400, el qual no és altre que la referència de la nostra IMU.

En cas de voler llegir aquesta IMU amb una comunicació de 8 bits, el procediment seria similar sols que, al tenir que enviar 16 bits, deuríem d'enviar i llegir dos paquets de 8 bits en cada activació de la línia de *Chip Select*, tal i com s'aprecia en la figura 99.

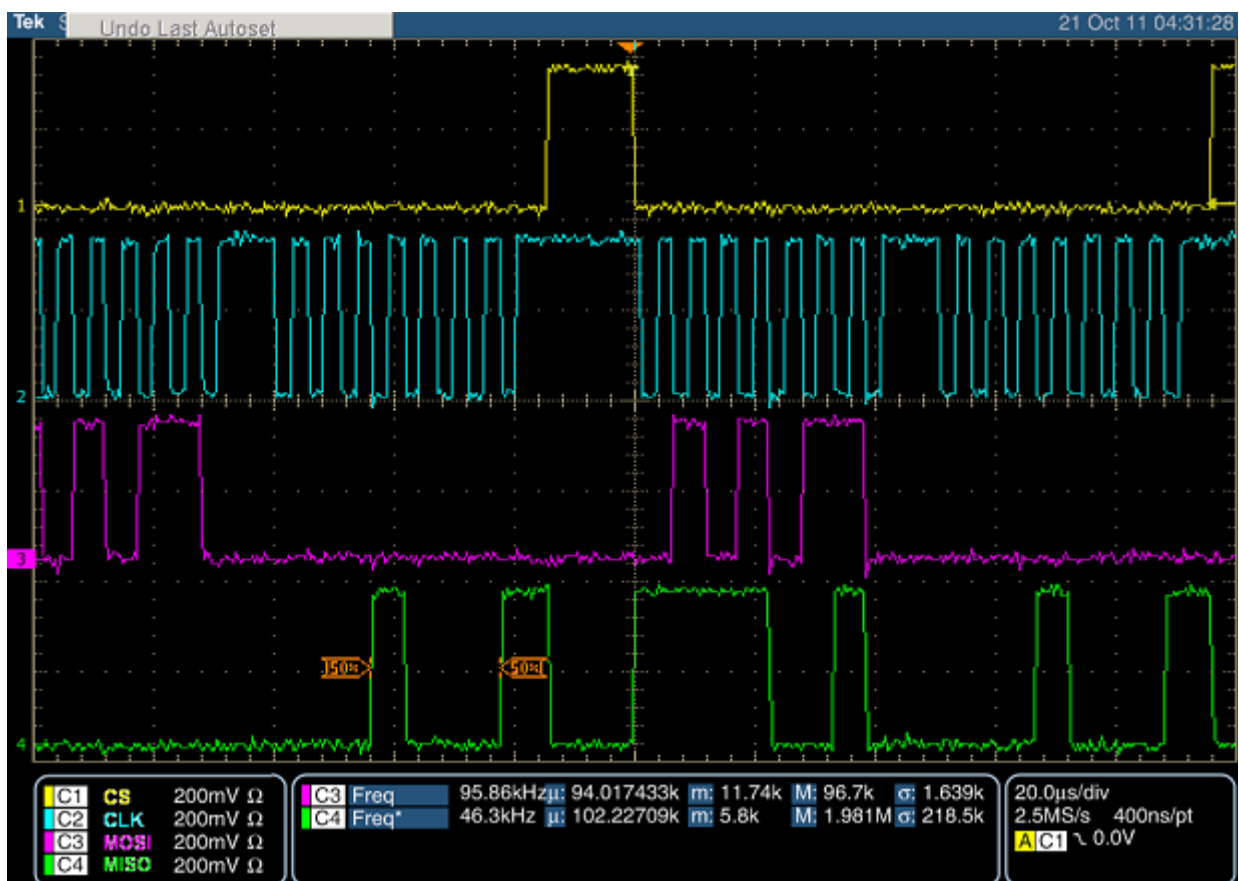


Figura 99: Lectura fallida, en mode 8 bits, del registre d'identificació de l'ADIS16400.

Sí no tenim en compte el fet que es tracta d'una lectura corrupta, la principal diferència d'aquesta figura respecte a l'anterior està en la línia de rellotge CLK. Tal i com s'observa, el temps que passa a nivell alt el octau polse és major que el dels altres polses. Aquest detall podria fer pensar que es tracta del motiu pel que no s'ha pogut llegir el registre d'identificació del dispositiu, però en realitat no té molt a veure ja que l'efecte que produeix eixe polse és un augment de la duració del bit que li correspon, és a dir que, sí per exemple, un bit en condicions normals dura un microsegon, el bit assignat a este polse durarà poc més de dos microsegons. Aquest efecte és normal i es deu al temps que tarda la lògica del microcontrolador en preparar la següent dada i deixar-la en el bus.

El procés d'escriptura d'aquesta IMU és en essència una lectura, però amb algunes diferències:

- 1.- Sumar 0x80 al valor de l'adreça a llegir, desplaçar 8 bits a l'esquerre el resultat i sumar-lo al valor de 8 bits que volem escriure en el registre de l'ADIS.
- 2.- Baixar a '0' la línia de *Chip Select*.
- 3.- Carregar el resultat del pas 1 en el *buffer* d'eixida del microcontrolador.
- 4.- Esperar a que es buide el *buffer* del microcontrolador, és a dir, a que s'acabe de transmetre la dada.
- 5.- Llegir el *buffer* del microcontrolador i descartar la dada llegida.
- 6.- Pujar a '1' la línia de *Chip Select* durant uns 10 microsegons.

Hem de tenir en compte que, degut al format utilitzat per a enviar ordres a aquesta família de IMUs i a la seva organització interna, sí volem escriure un valor major de 255 estarem obligats a fer dos escriptures: una amb la part baixa d'aquest valor i una altra amb la part alta. Òbviament l'adreça que gastarem en cada cas correspondrà a la part del valor que anem a escriure. Això vol dir que sí anem a escriure en la part alta del valor d'*off-set* del acceleròmetre Az, l'adreça que seleccionarem serà la 0x0F. En canvi, sí es tracta de la part baixa del valor, aleshores utilitzarem l'adreça 0x0E.

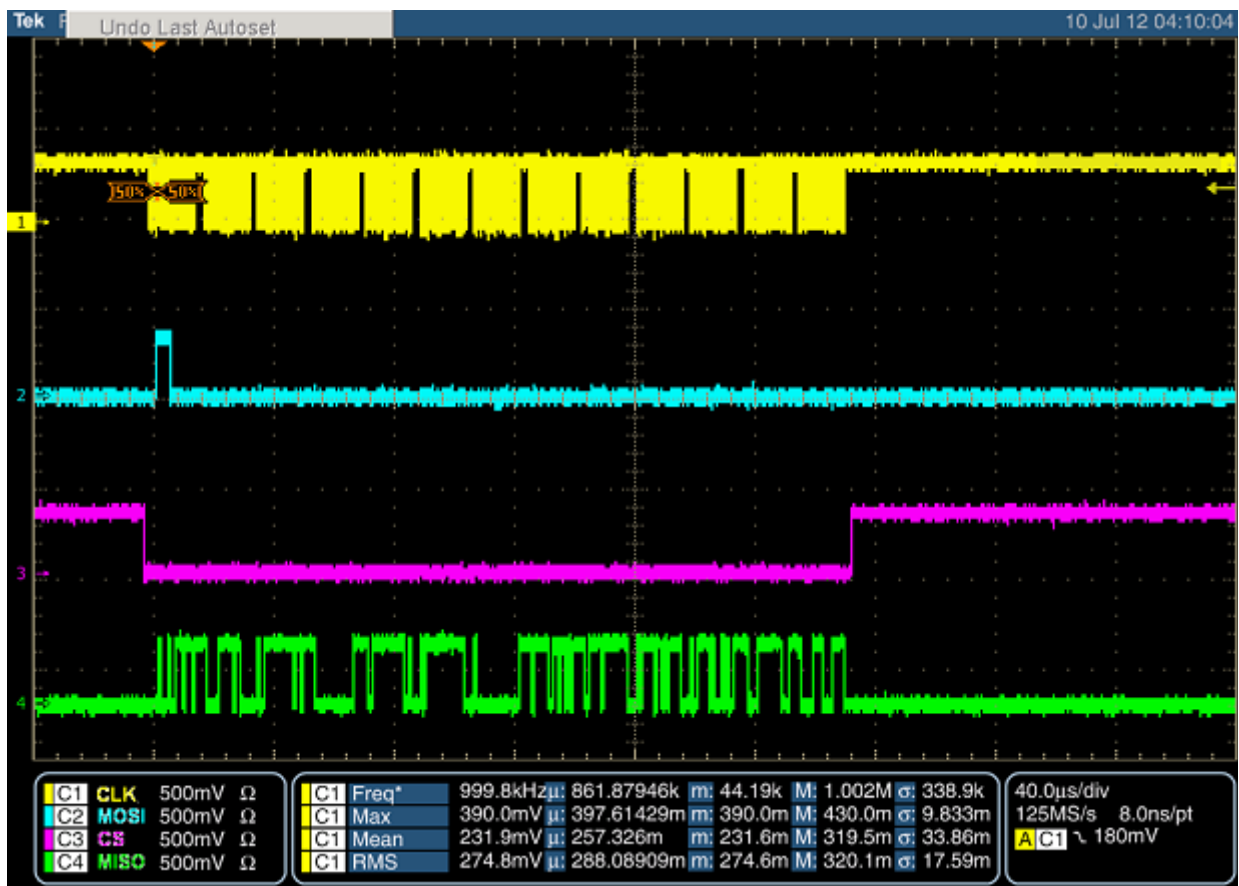


Figura 100: Lectura en mode ràfega de la IMU ADIS16400

Per últim ens queda parlar d'un mode de lectura especial de les IMUs ADIS: El mode ràfega o *Burst*. La diferència entre aquest mode i el convencional és la seva eficiència ja que, com podem comprovar en la figura 100, amb una sola lectura l'ADIS ens envia de colp el contingut dels 13 registres d'eixida dels sensors. Una altra avantatge d'aquesta funció és la coherència de les dades, ja que al llegir tots els registres de colp podem estar segurs que la informació obtinguda correspon a la situació actual del sistema, mentre que sí els llegim un a un, la informació obtinguda de l'últim sensor pot ser no tinga ja res a veure amb la informació subministrada instants abans per un altre sensor.

Per a fer ús d'esta funció primer configurarem la velocitat de rellotge a 1MHz o menys. Després, quan s'active la senyal *Data Ready*, activarem el *Chip Select*, enviarem la dada 0x3E00 i tot seguit enviarem 12 caràcters qualsevol per a que l'ADIS envie el contingut dels dotze registres. Finalment, desactivarem el *Chip Select* i esperarem almenys  $1/F_{SCLK}$  abans de tornar a utilitzar este mode.

### 7.3- El receptor de GPS 610-F

El receptor GPS 610-F és un mòdul GPS comercialitzat per la casa *RF Solutions* amb les següents característiques tècniques:

- Conjunt de Xips Venus 634 del fabricant *Skytraq*.
- 65 Canals.
- 8 milions de proves per segon per a intentar deduir la localització del sistema.
- 29 segons per a localitzar el sistema al aire lliure i 1 segon “en calent”.
- Detecció i eliminació de senyals multicamí<sup>54</sup>.
- Compatible amb els sistemes WAAS i EGNOS<sup>55</sup>.
- Detecció de senyal superior a -161dBm
- Connector MMCX per a poder utilitzar una antena externa activa
- Consum en mode de rastreig: 23mA
- Velocitat de transmissió configurable des de 4800 a 115200 bauds.
- Actualització de posició configurable de 1 Hz a 10 Hz.
- Protocol d'eixida seleccionable entre protocol NMEA i format propietari del fabricant.

A excepció de poder utilitzar una antena externa de la nostra elecció, les principals característiques d'aquest receptor són similars a les habituals en receptors d'aquest rang de preus. Afortunadament per a nosaltres, mentre s'analitzava el funcionament del receptor, vam descobrir de casualitat el fabricant i model del *chipset* del receptor dades que, per cert, no s'especifiquen en la fulla de característiques tot i que es presumeix de la possibilitat configurar certs paràmetres segons les necessitats de l'usuari.

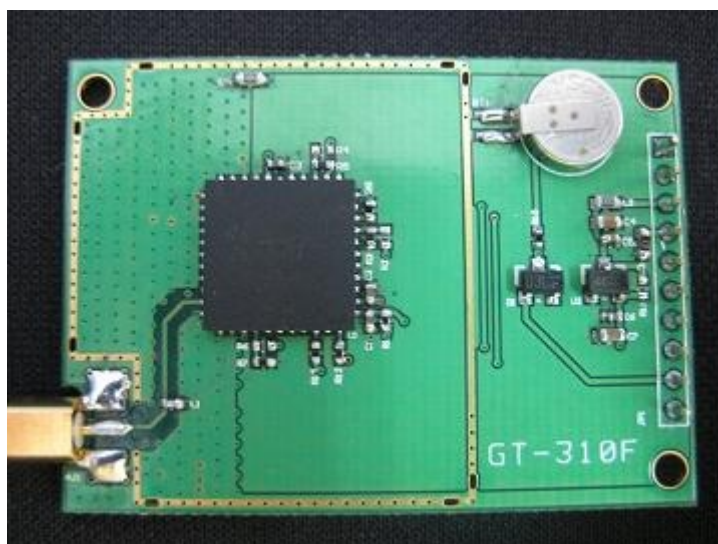


Figura 101: El receptor GPS 610F al descobert.

Sí volem canviar la configuració del receptor i no disposem d'informació sobre les ordres internes del mateix, conèixer el fabricant i el model del *chipset* serà de vital importància pues, sí el fabricant és relativament conegut i seriós, serà més fàcil trobar aquesta informació. Per a sort nostra, tant el

<sup>54</sup> Per a obtenir més informació sobre aquest concepte, consultar secció 2.3- Conceptes Bàsics.

<sup>55</sup> Per a obtenir més informació sobre aquesta característica consultar secció 2.4- GPS Diferencial i sistemes SBAS.

fabricant, *Skytraq*, com el *chipset*, Venus 634, tenen molta presència en la xarxa i hem pogut trobar tota classe d'informació: des de fulla de característiques del *chipset*, passant per muntatges casers fins a arribar al que ens interessa: una descripció tècnica de les ordres internes del *chipset* i inclús una aplicació per a *Windows* que ens permet configurar el receptor sense necessitat d'implementar en la nostra aplicació el protocol de comunicació propietari del fabricant.

Sí no tenim el protocol de comunicació propietari del fabricant en compte, el receptor de GPS és segurament el component més fàcil d'utilitzar d'aquest sistema ja que, per norma general, tan prompte com se li aplica l'alimentació, este es dedica a enviar les cinc frases NMEA que té configurades per defecte a una velocitat fixa per un dels ports series asíncrons dels que disposa pel que, mentre no ens importe l'eficiència del sistema, sols ens hem de limitar configurar la comunicació asíncrona del nostre microcontrolador i, quan ens arribe el caràcter ASCII enviat pel GPS, capturar-lo.

Tot i que nosaltres ens hem limitat a capturar aquests caràcters, cal dir que una vegada es va identificar el fabricant del conjunt de xips del receptor, es va decidir que, sí hi havia prou temps, es dissenyaria una placa electrònica per a adaptar els nivells de tensió del receptor GPS amb els nivells de tensió del port serie d'un PC per a, d'aquesta manera, utilitzar l'aplicació *Windows* de *Skytraq* i configurar el receptor de manera òptima. És cert que podríem configurar el receptor des de el nostre sistema sí implementarem el protocol propietari del fabricant, però com que estem treballant amb un prototip, no és bona idea ja que més avant, pel motiu que siga, podríem canviar de fabricant de GPS i tota aquesta feina quedaria en res, pel que deuriem de tornar a dedicar temps a implementar el protocol de comunicació propietari del nou receptor.

#### **7.4- Altres components: Display de 20 caràcters i 2 línies.**

El *display* LCD és l'únic dispositiu del que podem prescindir en el sistema. De fet, s'ha inclòs en el disseny perquè pràcticament l'únic que havíem de fer per a utilitzar-lo era connectar-lo elèctricament al sistema, pues abans d'iniciar el disseny disposàvem tant del *display* com del codi font necessari per a fer-lo funcionar.

La pantalla que hem utilitzat és la referència FDCC2004B-RNNYBW de la casa *Fordata* i disposa de les següents característiques:

- Controlador compatible amb el controlador de *display* HD44780.
- Possibilitat de regular el contrast de la pantalla i de alimentar-lo tant a 3.3V com a 5V.
- Comunicació via port paral·lel de 4 o 8 bits
- 2 línies amb capacitat per a 20 caràcters per línia (20x2)
- Caràcters de 5x8 i cursor personalitzable
- 80 bytes de RAM
- Joc de caràcters ASCII Americans en ROM
- Possibilitat de crear caràcters nous
- Funcions internes per a esborrar pantalla, desplaçar el cursor, fer-lo parpellejar, etc.

Com aquesta pantalla utilitza un controlador compatible amb el popular *Hitachi* 44780 i aquest controlador està ben documentat, tant en la xarxa com en llibres sobre microcontroladors, no entrarem en detalls del seu funcionament. De fet, per a controlar esta pantalla en el nostre disseny, hem utilitzat el codi present en el llibre *Programming 16-Bit PIC Microcontrollers in C - Learning to Fly the PIC24* de l'editorial *Newnes*, però canviant els temps d'espera de les funcions d'inicialització del *display*, doncs estos són específics per a cada fabricant/pantalla.

L'únic inconvenient d'haver utilitzat aquesta pantalla és la quantitat de caràcters que podem mostrar a la vegada. A l'inici del projecte, com la idea era utilitzar-la quasi exclusivament com a ferrament de depuració, no era un detall important, però a mesura que s'anava avançant i s'afegien noves funcions al disseny, esta particularitat del *display* va anar complicant la programació i, en més d'una ocasió, s'ha trobat a faltar una pantalla de 20x4, en especial a l'hora de mostrar les lectures del sensor, pues al no tenir prou línies, no es poden visualitzar totes les mesures en una mateixa pantalla. A més, una pantalla d'aquest tipus no sols ralentitza el sistema, sinó que a més, a l'haver-la connectat al microcontrolador utilitzant el port paral·lel de 8 bits, complica l'*enrutament* de les pistes de la placa de circuit imprès i augmenta considerablement les dimensions de la mateixa.



Figura 102: Detall del *display* una vegada funcionant en la placa prototip

## 8- Esquema elèctric i distribució dels components del sistema

Per al disseny de l'esquemàtic i del *layout* del sistema hem utilitzat el paquet de *software* *Orcad* en la seva versió 15.7.

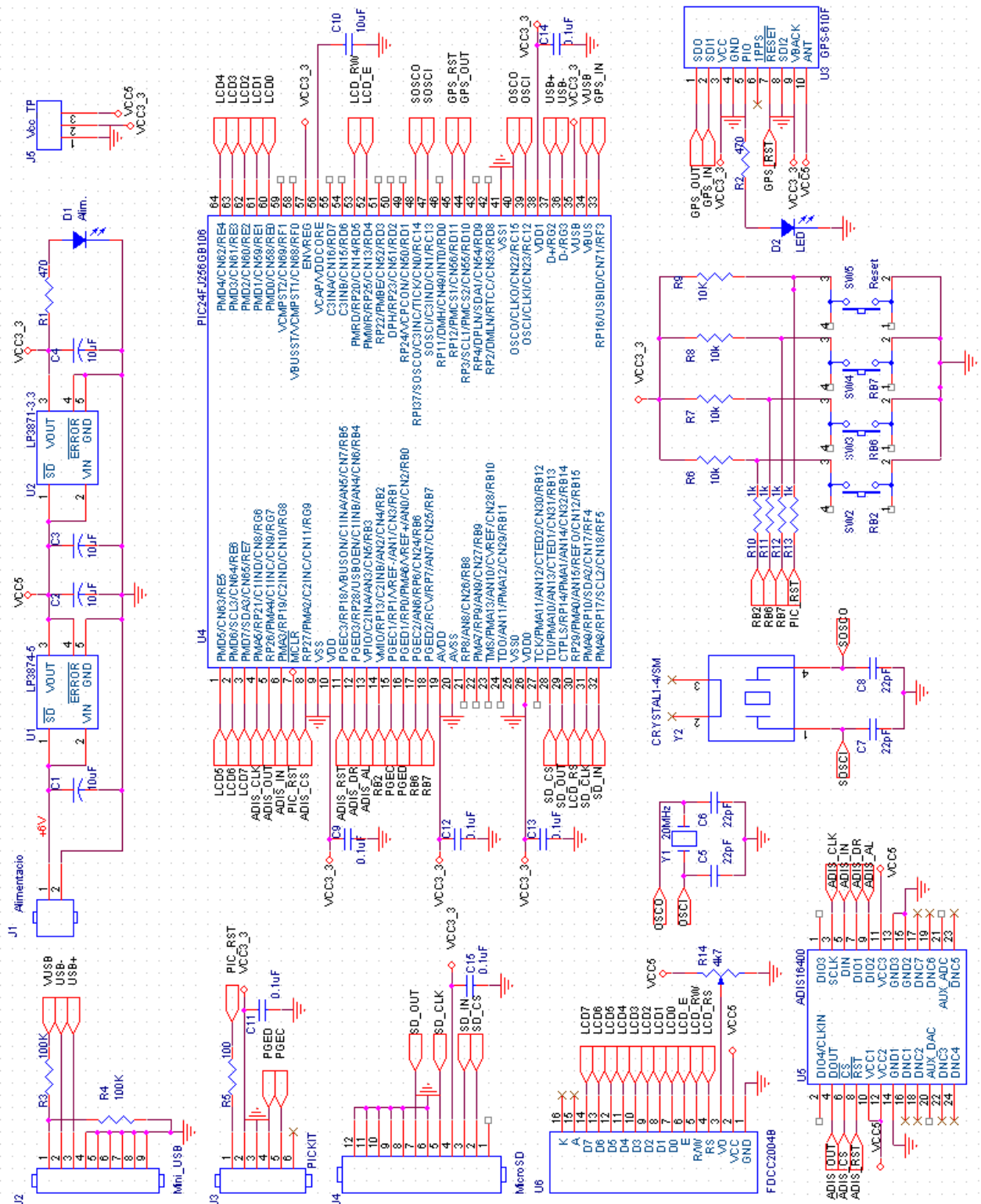


Figura 103: Esquemàtic general del sistema

En la figura 103 tenim l'esquema general del sistema. En el centre es troba la part de control, en els laterals els perifèrics, la interfície amb l'exterior i, en la part superior central, el circuit d'alimentació.

## 8.1- Circuit d'Alimentació

Aquest circuit és l'encarregat de subministrar energia elèctrica al circuit. Com la idea principal és que el sistema funcione a bateries, està integrat per dos reguladors: un LP3874EMP de 5 Volts i un LP3871EMP de 3.3 Volts amb els corresponents condensadors associats als mateixos. Cal dir que podríem haver alimentat el sistema a través del bus USB, però com en un principi ni tan sols anàvem a fer ús d'aquest bus, vam descartar l'opció.

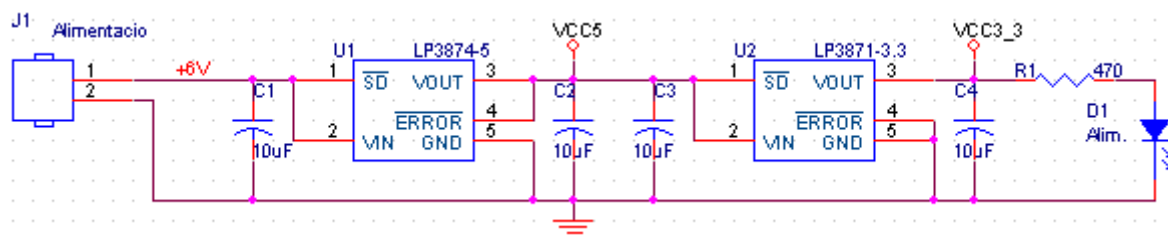


Figura 104: Esquemàtic del circuit d'alimentació

Com s'observa en la figura 104, el circuit d'alimentació és molt simple, consta d'un LED per a indicar que el sistema està alimentat i no inclou cap mesura de protecció. Tot i que els reguladors estan en serie, el regulador de 5 volts no alimenta exclusivament al regulador de 3.3 Volts, sinó que alimenta també a la pantalla LCD i al sensor d'inèrcia ADIS16400. La resta de components, és a dir, microcontrolador, targeta SD i GPS són alimentats pel regulador de 3.3 Volts.

Per al disseny d'aquesta etapa ens hem basat en les recomanacions proporcionades pel fabricant en la corresponent fulla de característiques. Hem de tenir en compte que, a pesar que els símbols que hem utilitzat en l'esquemàtic per als dos reguladors són idèntics, els reguladors LP3874 disposen d'un terminal de sensat de corrent SENSE en lloc del terminal ERROR', per això en el regulador de 5 volts el pin 4 està directament connectat a l'eixida mentre que en el de 3.3 volts, com no s'utilitza aquesta funció, està connectat a terra.

Degut a les especificacions dels reguladors no podem aplicar en el connector J1 una tensió d'alimentació inferior a 5,35<sup>56</sup> volts ni superior a 7 volts. A més, la carrega màxima que podem alimentar entre la línia de 3.3 volts i la de 5 no ha de ser superior als 800 mili ampers. Això vol dir que el circuit el podem carregar amb uns 600 mili ampers més i que amb la carrega actual podrà alimentar el sistema durant unes 7 hores i mitja, pues:

$$I_{\text{SISTEMA}} = I_{\text{GPS}} + I_{\text{ADIS}} + I_{\text{SD}} + I_{\text{LCD}} + I_{\text{PIC}} \approx 200\text{mA}$$
$$\text{Autonomia} = \text{Capacitat bateries} / I_{\text{SISTEMA}} = 1500\text{mAh} / 200\text{mA} \approx 7,5 \text{ Hores}$$

En la pràctica, mentre registrava cada 2 segons en la targeta Micro-SD la posició i les lectures del sensor d'inèrcia, l'autonomia ha sigut de poc més de 6 hores. En canvi, simplement llegint el sensor d'inèrcia s'ha obtingut una duració d'un 7 hores.

## 8.2- Circuit de control

El circuit de control gestiona els perifèrics del sistema i processa la informació proporcionada per aquests. A l'estar format per un microcontrolador, la complicació en el disseny és mínima ja que tots els recursos bàsics del sistema, com puga ser la memòria de programa o els mòduls de comunicació, estan incorporats en el mateix encapsulat del processador pel que bàsicament es podria dir que ens hem limitat a desacoblar mitjançant condensadors l'eixida de 3.3

<sup>56</sup> La tensió d'alimentació en J1 és l'eixida del regulador de 5 volts més la tensió de *Dropout*, la qual depèn del consum de corrent d'aquest regulador. Per a més detalls consultar en la fulla de característiques del LP3874 les especificacions elèctriques.





Per a poder programar el microcontrolador des del mateix circuit i sense dessoldar el microcontrolador, hem connectat les línies de *Reset*, alimentació de 3.3 volts, dades de programació PGED i rellotge de programació PECEG a un connector estàndard de 6 pins. L'assignació de terminals d'aquest connector està feta per a coincidir amb el del programador *Pickit* de *Microchip*, però també deuria servir qualsevol altre que respecte les especificacions de programació del microcontrolador.

Respecte a la font de rellotge del microcontrolador, com interessava que executés el major nombre d'instruccions per segon, vam optar per utilitzar com a font primària d'oscil·lació un rellotge extern de 20MHz. Després, per imposicions del mòdul de Rellotge en Temps Real, hem tingut que connectar un segon oscil·lador de 32,768Hz als terminals 47 SOSCI i 48 SOSCO. El valor de tots els condensadors associats als cristalls, al igual que passa amb tots els condensadors associats al microcontrolador, és el recomanat per *Microchip* en la fulla de característiques del PIC24FJ256GB.

Després de fer aquestes connexions bàsiques hem passat a connectar els restants components als terminals assignats i a deixar com a no connectats tots els terminals que han quedat lliures.

### 8.3- Interfície Entrades/Eixides

Dins d'aquesta etapa podem distingir 6 circuits diferents que, tot i que no estan relacionats entre ells, comparteixen la mateixa finalitat: permetre l'intercanvi d'informació entre el microcontrolador i el món exterior.

#### 8.3.1- Teclat

Aquesta part del circuit permet que un usuari pugui interactuar amb el sistema. Tal i com s'observa en la figura 106 està format per quatre pulsadors i per resistències *pull-up*.

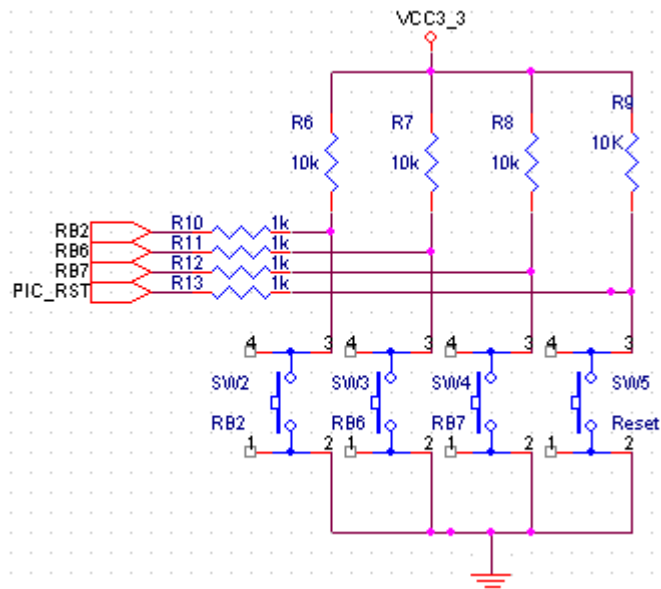


Figura 106: Esquemàtic del teclat

Al dissenyar aquesta part no vam caure en el detall que el microcontrolador disposa de resistències internes de *pull-up* pel que no sols ens hauríem estalviat comprar i soldar les resistències, sinó que també hauríem simplificat un poc el disseny.

En aquest circuit podem distingir dos tipus de polsadors: el de *Reset* i els de control. El primer està connectat al terminal 7 MCLR del PIC, pel que quan es polse, la tensió en el terminal serà zero i es forçara al microcontrolador a fer un *Reset*. En canvi els segons s'han connectat al port B del microcontrolador, concretament als terminals 14 RB2, 17 RB6 i 18 RB7, amb la idea de ser utilitzats per a controlar de forma simple el sistema, encara que inicialment es van utilitzar com a ferramenta de depuració.

### 8.3.2- Display LCD

La pantalla LCD és l'únic component per al que no hem tingut que planificar la connexió al microcontrolador. Açò es deu a que el PIC sols disposa d'un port paral·lel, pel que connectar l'un amb l'altre no ha sigut excessivament complicat.

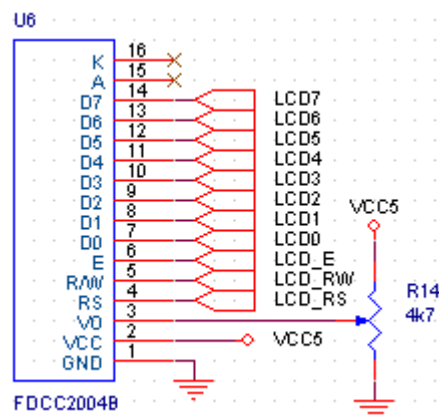


Figura 107: Esquemàtic de la pantalla LCD

Al tenir terminals suficients per a utilitzar la pantalla en mode de 8 bits hem utilitzat la configuració de la figura 106 per a connectar les senyals de control del *display* amb el microcontrolador. Com aquest component no disposa de més de dos registres de control, sols ha sigut necessari connectar un terminal de direccions del port paral·lel del PIC al *display*, concretament al *pin* 4 LCD\_RS.

Per a poder regular el contrast de la pantalla hem utilitzat un potenciòmetre per a realitzar el divisor de tensió necessari per a implementar aquest ajust. En canvi, per a no augmentar el consum de la pantalla, ens hem deixat per connectar l'alimentació de la retro-il·luminació.

### 8.3.3- Connexió amb l'ADIS16400

En principi, per a connectar l'ADIS16400, al tractar-se d'un dispositiu SPI, sols necessitaríem connectar quatre dels seus terminals al microcontrolador però, a canvi d'eixa simplicitat en la connexió, no podríem utilitzar, per exemple, la funció d'alarma que incorpora.

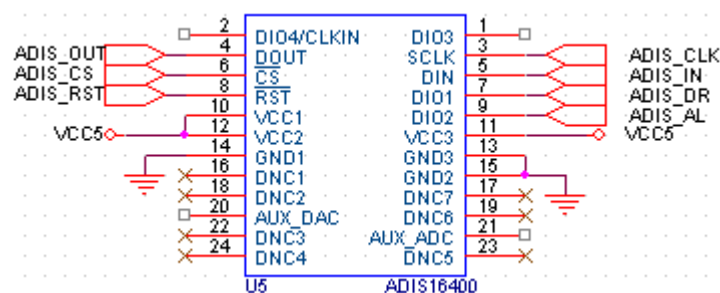


Figura 108: Esquemàtic del sensor d'inèrcia.

Així doncs, a falta de trobar una aplicació per al convertidor A/D-D/A i una funció interessant als terminals auxiliars 1 DIO3 i 2 DIO4, hem connectat tots els altres terminals del sensor al microcontrolador.

Per a poder reutilitzar el disseny amb el DSPIC33EP256MU806 hem connectat els terminals relacionats amb el protocol SPI del ADIS16400, 3 SCLK, 4 DOUT i 5 DIN, als terminals 4 RP21, 5 RP26 i 6 RP19 del microcontrolador. Els restants pins, com són d'ús general, s'han connectat en els terminals que més prop estaven dels anteriors, pel que sí ho necessitarem, podríem desplaçar-los a uns altres terminals d'entrada/eixida.

Tal i com hem dit abans, la connexió del sensor d'inèrcia podria simplificar-se, però perdríem funcionalitats interessants i inclús alguna de necessària com és el cas de la funció *Data Ready*, pues tot i que és cert que podem saber si hi han dades noves comprovant el bit *New Data* del registre d'eixida que acabem de llegir, a l'hora d'utilitzar els filtres, per a assegurar-nos la coherència de les dades que capturem, el fabricant ens diu que la millor forma de fer-ho és contant els polses d'aquest terminal. Per aquest motiu, la connexió mínima recomanable són 5 terminals: SCLK, DIN, DOUT, CS i el terminal al que assignem la funció *Data Ready*.

### 8.3.4- Connexió amb el receptor de GPS

La connexió del receptor de GPS amb el microcontrolador és prou més simple que la del ADIS16400 pues, al ser una comunicació asíncrona, sols són necessàries dos línies de comunicació: una per a la recepció i l'alta per a l'escriptura, encara que esta última, en el cas dels receptors GPS, no sol ser necessària si tenim pensat no implementar el protocol propietari de comunicació del receptor per tal de configurar-lo.

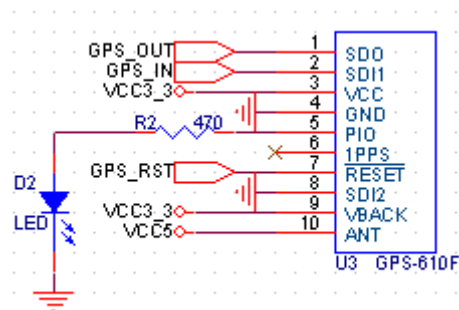


Figura 109: Esquemàtic del receptor de GPS

A part de les línies de comunicació amb el microcontrolador, el GPS disposa d'una serie de terminals dels que podem fer ús segons ens interesse. En el nostre disseny, com no és una eixida present en tots els receptors GPS, no hem connectat la senyal de sincronització horària 6 1PPS i hem connectat l'entrada de dades DGPS 8 SDI2 a massa. En canvi, per sí al final decidíem comunicar-nos amb el GPS, vam connectar l'entrada de dades asíncrona 2 SDI1 al terminal al que vam assignar l'eixida de la UART en el PIC: 33 RP16. Cal destacar que, de no haver fet ús d'aquesta entrada, la deuríem d'haver connectat a la tensió d'alimentació del receptor ja que, en la transmissió asíncrona, quan no hi ha transmissió, la línia d'eixida del transmissor s'ha de mantenir a nivell alt.

Després de connectar les senyals de protocol al microcontrolador, ens queda veure que fem en les restants. Així doncs, per a saber ràpidament si el receptor GPS ha *fixat* ja la posició, hem connectat un díode LED al terminal d'estat 5 PIO. Una altra connexió que hem realitzat és la d'alimentar amb 5 Volts el terminal d'antena activa 10 ANT i, per a reduir el temps de localització inicial, hem alimentat amb 3.3 Volts el terminal 9 VBACK de recolzament d'energia de la SRAM i RTCC.

### 8.3.5- Lector de targetes MicroSD

Les targetes MicroSD tenen dos modes de funcionament: el mode SD i el mode SPI. Com per a poder utilitzar el primer mode s'ha de tenir accés a les especificacions del protocol, hem optat per utilitzar el mode de comunicació SPI. Al ja disposar en el propi microcontrolador d'un mòdul SPI, la connexió amb el mateix es limita a desacoblar l'alimentació de la targeta MicroSD, assignar un terminal genèric d'entrada/eixida del microcontrolador per a realitzar la funció de *Chip Select* i, finalment, connectar els terminals del microcontrolador als que hem assignat les funcions de port SPI als respectius terminals de la targeta MicroSD.

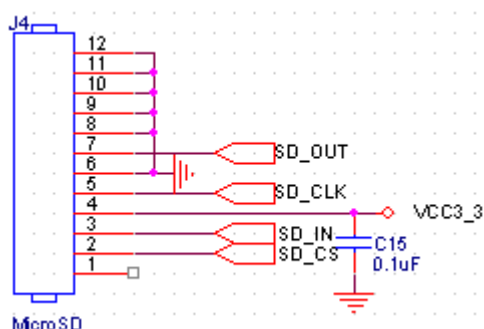


Figura 110: Esquemàtic del lector de targetes MicroSD

Sí ens hem fixat en la figura 110 i alguna vegada hem treballat amb targetes SD de grandària estàndard, haurem observat que la distribució de terminals no sols no correspon amb la d'una targeta MicroSD, sinó que a més, ni disposa del terminal que indica que hi ha una targeta en el lector, ni té el terminal que ens informa que la targeta està protegida contra escriptura. Este detall no afecta al bon funcionament de la targeta, però com veurem més avant, l'haurem de tenir en compte quan accedim a ella des de el microcontrolador utilitzant les llibreries d'aplicació de *Microchip*.

### 8.3.6- Port USB

El bus USB és un protocol de comunicació diferencial que consta de dos terminals de dades més un tercer terminal que pot utilitzar-se per a que el dispositiu mestre alimente als dispositius convidats, o com en el nostre cas, per a detectar que el sistema ha sigut connectat via USB a un dispositiu mestre.

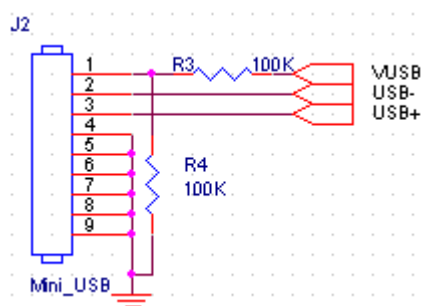


Figura 111: Esquemàtic del port USB

A l'igual que passa amb els dispositius amb comunicació SPI, la connexió del microcontrolador amb aquest bus és molt simple ja que el propi microcontrolador incorpora internament un mòdul de comunicació USB. Gràcies a açò, per a utilitzar aquest bus, sols cal connectar els terminals de comunicació USB- i USB+ amb els terminals 36 D- i 37 D+ del microcontrolador i el terminal d'alimentació VUSB al terminal 34 VBUS del microcontrolador a través d'un divisor resistiu, tal i com recomana *Microchip* en la figura 18-4 de la pàgina 209 de fulla de característiques.

## 8.4- Distribució dels components en la PCB

El següent pas després de crear l'esquemàtic és crear els components i els seus *footprints*. Com el *Orcad* sols té petjades per als components més comuns, vam tenir que dissenyar les petjades de la majoria dels components. Una vegada dissenyats, els vam distribuir sobre la PCB de manera que els oscil·ladors quedaren el més prop possible del PIC i les parts més sorolloses el més lluny possible del sensor d'inèrcia. El resultat de la distribució el podem veure en les figures 112 i 113:

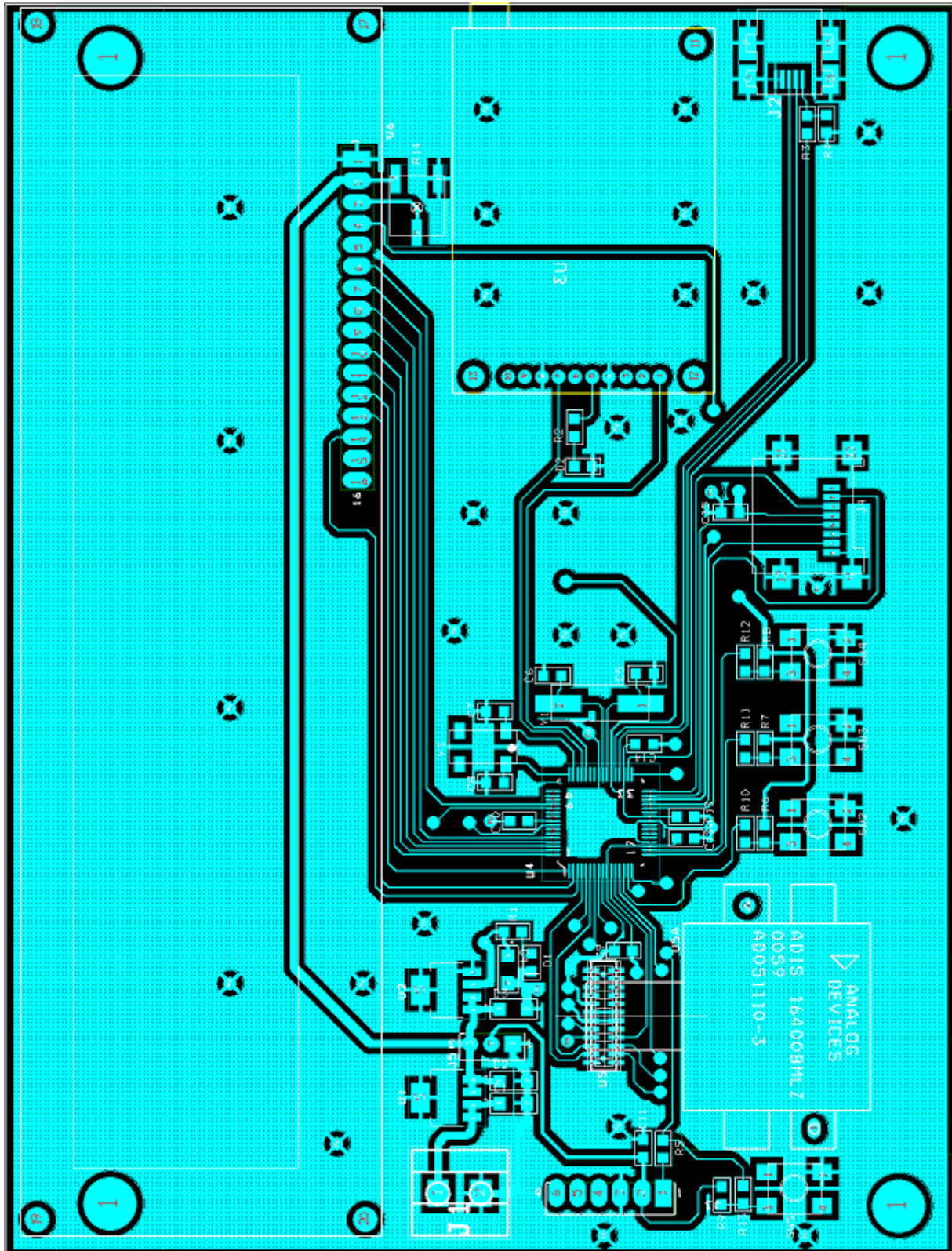


Figura 112: Part superior del *layout* de la PCB

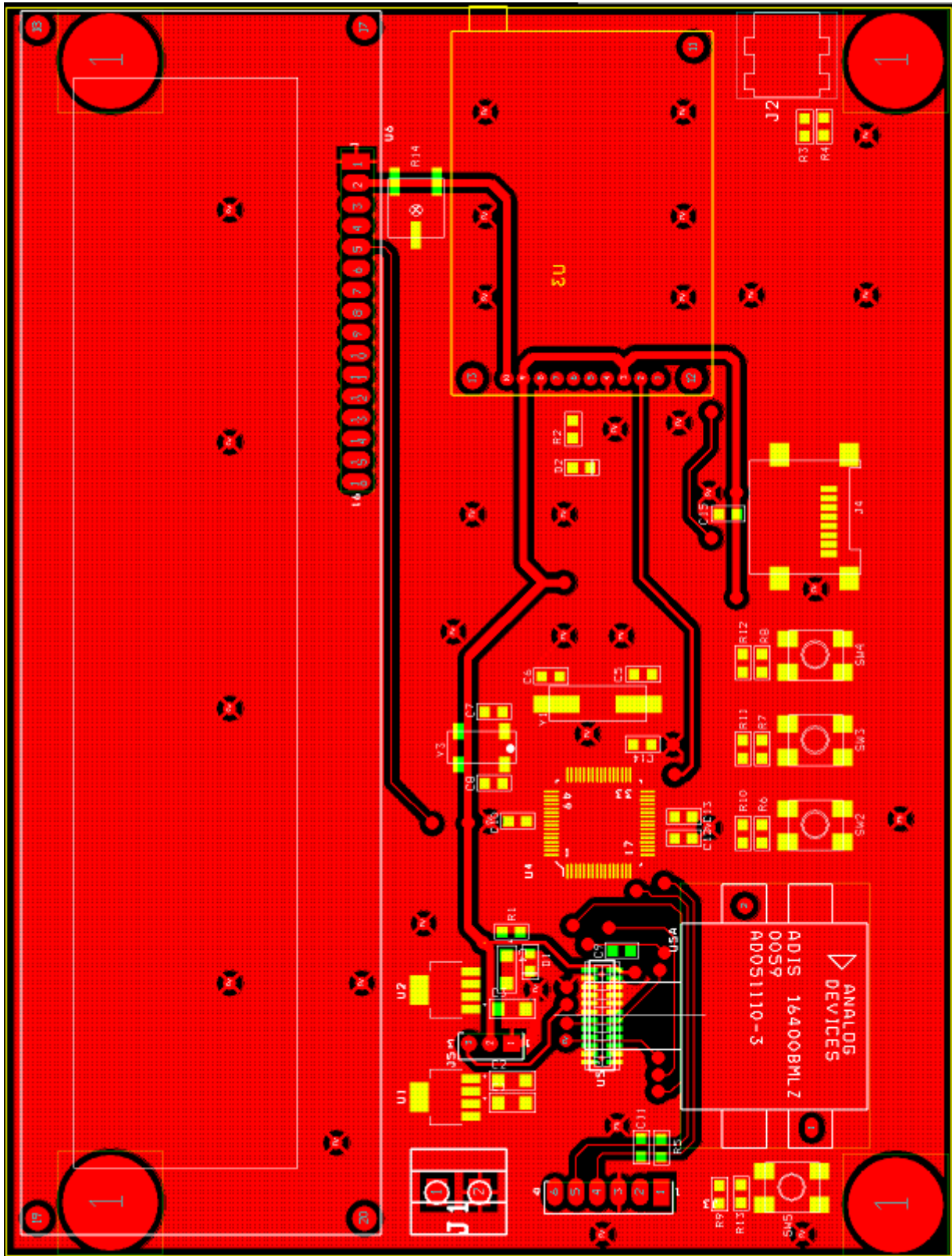


Figura 113: Part inferior del *Layout* de la PCB

El llistat de components i les seves referències en la placa de circuit imprès és el següent:

| Referència | Descripció                    | Referència <i>Farnell/RS</i> | Preu unitari (€) |
|------------|-------------------------------|------------------------------|------------------|
| C1         | Condensador 10uF / V          | <i>Farnell</i> 165-8938      | 1,290            |
| C2         | Condensador 10uF / V          | <i>Farnell</i> 165-8938      | 1,290            |
| C3         | Condensador 10uF / V          | <i>Farnell</i> 165-8938      | 1,290            |
| C4         | Condensador 10uF / V          | <i>Farnell</i> 165-8938      | 1,290            |
| C5         | Condensador 22pF              | <i>Farnell</i> 136-2555      | 0,034            |
| C6         | Condensador 22pF              | <i>Farnell</i> 136-2555      | 0,034            |
| C7         | Condensador 22pF              | <i>Farnell</i> 136-2555      | 0,034            |
| C8         | Condensador 22pF              | <i>Farnell</i> 136-2555      | 0,034            |
| C9         | Condensador 0.1uF             | <i>Farnell</i> 136-2552      | 0,035            |
| C10        | Condensador 10uF              | <i>Farnell</i> 165-8938      | 1,290            |
| C11        | Condensador 0.1uF             | <i>Farnell</i> 136-2552      | 0,035            |
| C12        | Condensador 0.1uF             | <i>Farnell</i> 136-2552      | 0,035            |
| C13        | Condensador 0.1uF             | <i>Farnell</i> 136-2552      | 0,035            |
| C14        | Condensador 0.1uF             | <i>Farnell</i> 136-2552      | 0,035            |
| C15        | Condensador 0.1uF             | <i>Farnell</i> 136-2552      | 0,035            |
| D1         | Díode LED verd (VCC)          | <i>Farnell</i> 122-6372      | 0,092            |
| D2         | Díode LED blau (GPS)          | <i>Farnell</i> 168-6076      | 0,048            |
| J1         | Connector alimentació         | Material del laboratori      | -                |
| J2         | Connector Mini-USB AB femella | <i>Farnell</i> 1696539       | 0,79             |
| J3         | Connector 6 pins (ICSP)       | Material del laboratori      | -                |
| J4         | Lector targetes MicroSD       | <i>Farnell</i> 1344008       | 2,32             |
| J5         | Connector 3 pins (VCC)        | -                            | No instal·lat    |
| R1         | Resistència 470 $\Omega$      | <i>Farnell</i> 109-9798      | 0,015            |
| R2         | Resistència 470 $\Omega$      | <i>Farnell</i> 109-9798      | 0,015            |
| R3         | Resistència 100K $\Omega$     | <i>Farnell</i> 146-9860      | 0,024            |
| R4         | Resistència 100K $\Omega$     | <i>Farnell</i> 146-9860      | 0,024            |
| R5         | Resistència 100 $\Omega$      | <i>Farnell</i> 146-9862      | 0,024            |
| R6         | Resistència 10K $\Omega$      | <i>Farnell</i> 146-9857      | 0,019            |
| R7         | Resistència 10K $\Omega$      | <i>Farnell</i> 146-9857      | 0,019            |
| R8         | Resistència 10K $\Omega$      | <i>Farnell</i> 146-9857      | 0,019            |
| R9         | Resistència 10K $\Omega$      | <i>Farnell</i> 146-9857      | 0,019            |
| R10        | Resistència 1K $\Omega$       | <i>Farnell</i> 933-3711      | 0,037            |
| R11        | Resistència 1K $\Omega$       | <i>Farnell</i> 933-3711      | 0,037            |
| R12        | Resistència 1K $\Omega$       | <i>Farnell</i> 933-3711      | 0,037            |
| R13        | Resistència 1K $\Omega$       | <i>Farnell</i> 933-3711      | 0,037            |

|     |                                  |                  |        |
|-----|----------------------------------|------------------|--------|
| R14 | Resistència variable 5K $\Omega$ | Farnell 168-9924 | 1,39   |
| SW2 | Switch (RB2)                     | Farnell 182-1257 | 0,540  |
| SW3 | Switch (RB6)                     | Farnell 182-1257 | 0,540  |
| SW4 | Switch (RB7)                     | Farnell 182-1257 | 0,540  |
| SW5 | Switch (Reset)                   | Farnell 182-1257 | 0,540  |
| U1  | Regulador LP3874EMP-5.0          | Farnell 8188556  | 1,95   |
| U2  | Regulador LP3871EMP-3.3          | Farnell 8188246  | 3,48   |
| U3  | Receptor GPS-610F                | RS 704-3297      | 31,80  |
| U4  | Microcontrolador PIC24FJ256GB106 | Farnell 1576868  | 6,57   |
| U5  | IMU ADIS16400                    | Farnell 185-2641 | 670,61 |
| U5A | Connector Sammtec ADIS16xxx      | Farnell 204-0664 | 1,840  |
| U6  | LCD FDCC2004B                    | RS 720-0226      | 11,24  |
| Y1  | Cristall 20 MHz                  | Farnell 184-2302 | 0,570  |
| Y2  | Cristall 32,768KHz               | Farnell 161-1824 | 0,690  |

Taula 17: Llista de components del sistema.

Sí sumem el preu dels components, tenim que el cost total dels mateixos és 751,23€ + IVA. A excepció de la pantalla, la IMU, el receptor de GPS i alguns connectors, tots els components són de muntatge superficial i han sigut soldats en una placa de circuit imprès de dos cares.

La placa ha sigut fabricada utilitzant una màquina de control numèric. Podríem haver-la fet utilitzant un mètode d'atac ràpid, però com la majoria de pistes del circuit no eren el suficientment grosses com per assegurar que l'acid no trencaria la continuïtat elèctrica de les mateixes al menjar-se-les, hem preferit no utilitzar-lo.

### **8.5- Placa prototip: errades en el disseny, contratemps i millores.**

El nostre disseny, com qualsevol altre, no ha estat lliure ni de problemes, ni d'errors. A pesar d'açò, podem estar molt satisfets amb el disseny inicial ja que ha funcionat des de el primer moment sense problemes, pel que les millores que es poden realitzar es limiten quasi exclusivament a reduir les dimensions de la PCB, cosa no molt complicada de fer ja que hi hauria prou amb buscar un receptor de GPS més menut i canviar la pantalla per una de dimensions més reduïdes, o bé llevar-la i utilitzar el port USB per a que l'usuari interactue amb el sistema. Això sí, en cas de desplaçar el GPS o utilitzar un altre model, deuríem de tenir en compte en quina part del receptor està la part de radiofreqüència per a evitar que esta etapa del receptor afectara a la resta del sistema.

Una altra millora que podríem aplicar al sistema és, utilitzant el convertidor A/D del PIC, controlar el consum general del sistema i el del port USB. D'esta manera podríem augmentar l'eficiència energètica del sistema i, en cas de detectar algun sobre-consum, detindre el sistema per a evitar mals majors. Per a fer açò sols caldria connectar l'eixida dels reguladors i el pin 34 VBUS del microcontrolador a alguns dels terminals lliures marcats com a ANx del microcontrolador.

Ja posats a jugar amb el consum d'energia i la font d'alimentació podríem aprofitar per a protegir el sistema contra sobre-consums afegint algun fusible o contra inversions de polaritat de les bateries afegint díodes. A més, també podríem veure d'adaptar el circuit d'alimentació per a que el sistema poguera ser alimentat també des de el port USB, pues treballar amb aquest bus mentre la placa s'està alimentant amb bateries resulta un tant incòmode.



L'última millora que se'ns ve al cap és portar els terminals no utilitzats del microcontrolador i del sensor d'inèrcia a un connector d'expansió o inclús veure d'utilitzar la senyal PPS del receptor de GPS per a sincronitzar els elements del sistema. Fent açò podríem afegir noves funcions al sistema sense tenir que tornar a fer una placa des de zero.

En el que respecta als contratemps sols em van tenir un però d'importants conseqüències ja que, degut a un descuit d'ús en el *Orcad Layout*, vam tenir que tornar a fabricar la placa de circuit imprès i, per tant, perdre quasi quatre vesprades de feina.



Figura 114: detall del error que ens va fer tornar a fabricar la PCB

Així doncs, degut a que l'*Orcad* no permet agrupar dos *footprints* en una sola i a que per a l'ADIS16400 se'n necessiten dos, una per al encapsulat i l'altra per al connector elèctric, en una de les vegades que el vam moure per a fer lloc als polsadors, se'ns va oblidar moure una de les dos petjades que el formen i no ens em vam adonar fins que ja teníem la placa feta amb totes les vies i el circuit d'alimentació soldat. Com podem veure en la figura 114, eixe detall va fer que els forats per a fixar sobre la PCB el sensor estiguen desplaçats respecte al connector elèctric, pel que o bé el deixàvem solt però connectat elèctricament, o bé el deixaven fixat a la placa però sense connectar elèctricament. Com en la nostra aplicació no és bona idea que el sensor es pugui moure lliurement i com encara no havíem soldat el microcontrolador, vam decidir tornar a fer la placa però corregint este error i un parell més que vam detectar mentre la revisàvem: les dimensions de les petjades dels condensadors d'alimentació C1, C2, C3 i C4 i el posicionament del receptor GPS, pues quan vam crear la petjada per a aquest component vam invertir la numeració dels terminals de manera que el terminal 1 era en realitat el 10 i al revés. Este error en principi és més important del que pareix, pues tot i que podem invertir la numeració dels terminals instal·lant el receptor en la cara posterior de la placa de circuit imprès, com esta cara és en la seva majoria un pla de massa, faria de pantalla i podria empitjorar la recepció de la senyal GPS.

Desgraciadament, després de tornar a fer la placa vam trobar un parell d'errors més: els forats per a fixar la pantalla LCD en la placa estan lleugerament desplaçats pel que els tornavisos no entren totalment rectes i el connector ICSP per a programar el microcontrolador en el propi circuit té la

numeració dels terminals invertida respecte a la numeració utilitzada en el Kit Pic. Això vol dir que, sí utilitzem un Kit Pic per a programar el microcontrolador, no podrem veure els LEDs ni el pulsador a no ser que inclinem la placa.

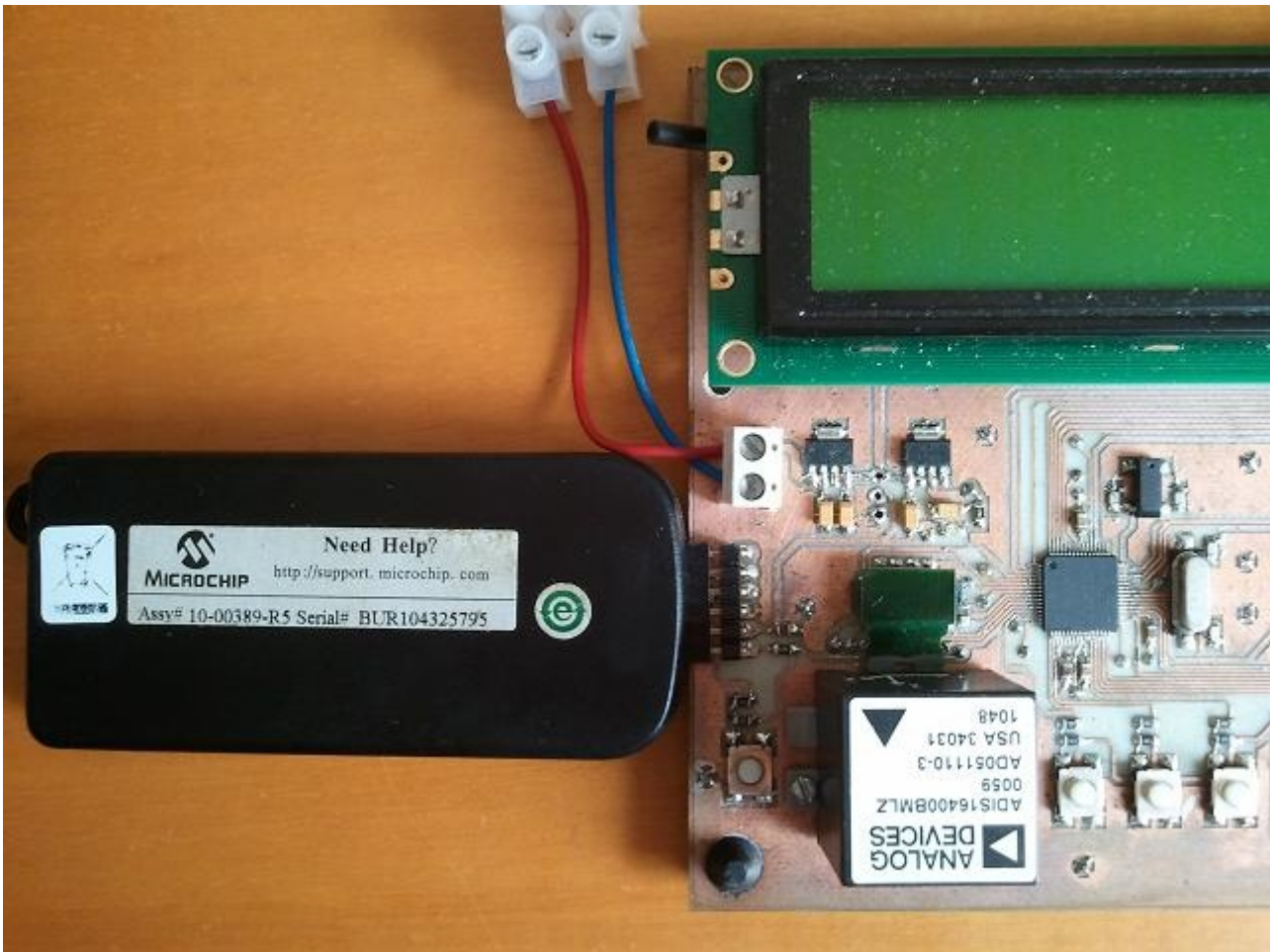


Figura 115: Efecte d'invertir la numeració del connector ICSP respecte a la utilitzada pel Kit Pic

Aquests errors, més que errors, són en realitat defectes cosmètics ja que no afecten al correcte funcionament del sistema, però en cas de tornar a fer una placa nova, seria interessant corregir-los per a millorar la imatge del disseny.

Des de el punt de vista elèctric no hem trobat cap error important encara que sí hi han varies coses que resultaria interessant canviar:

-En la figura 18-4 de la pàgina 209 de fulla de característiques del PIC es diu d'utilitzar un divisor de tensió resistiu tal que  $V_{BUS} = V_{USB}/2 = 2.5$  Volts per a implementar la funció *Bus Sense*. Esta tensió és considerada pel PIC com un nivell alt pels pèls, així que per a assegurar-nos que esta característica funciona correctament, s'haurà de canviar la relació del divisor resistiu de manera que  $V_{BUS} = 3.3$  Volts.

-El terminal 8 SDI2 del receptor GPS no és una entrada de dades normal sinó que es tracta d'una entrada de dades asíncrona. Per aquest motiu, esta entrada no hauria d'estar connectada a terra, sinó a l'alimentació del receptor, però com en la fulla de característiques no emfatitzaven aquest detall, no ens vam donar compte i la connectarem incorrectament.

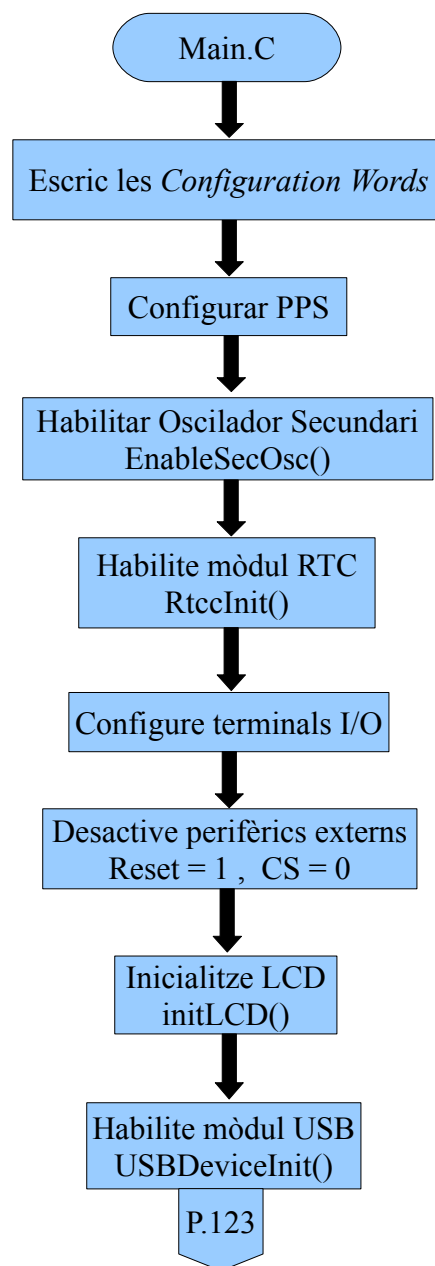
-Ni en la fulla de característiques de l'ADIS16400 ni en la del receptor GPS es diu res, però hauria sigut una bona idea desacoblar la seva alimentació amb condensadors.

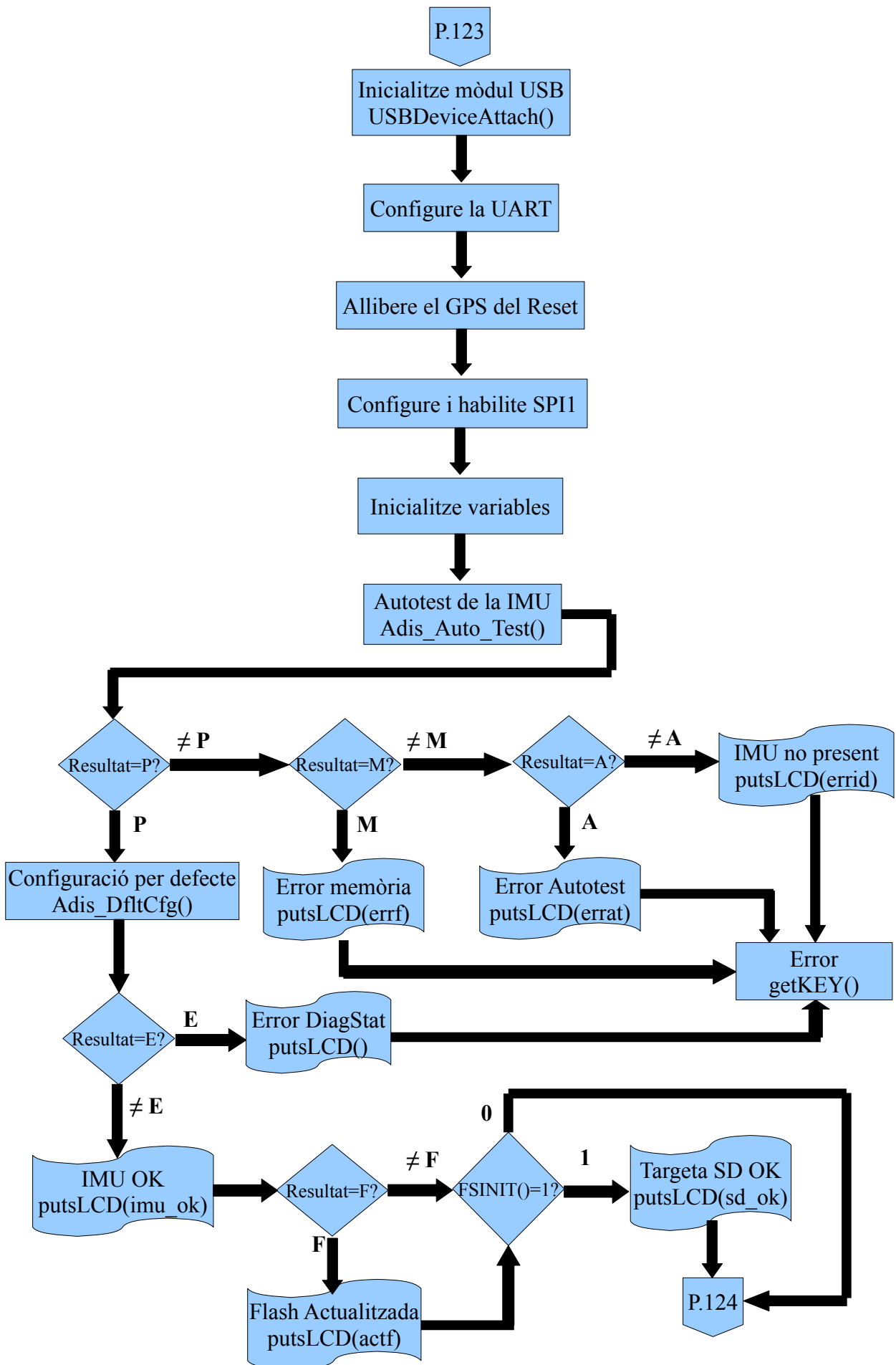
## 9- Programació del *Firmware* del sistema

El programa que executa el PIC24FJ256GB106 l'hem dividit en 7 parts: *main.c*, *adis16x000.c*, *gps.c*, *lcd.c*, *menus.c*, *rtcc.c* i la llibreria de magatzematge massiu de *Microchip*. Per norma general, cadascuna d'aquestes parts s'encarrega de controlar un dispositiu o perifèric del sistema i, a excepció de les funcions extretes de la llibreria d'aplicacions de *Microchip* i del llibre *Learning to Fly the PIC*, totes han sigut escrites des de zero.

### 9.1- Fitxer *main.c*

El fitxer *main.c* s'encarrega bàsicament de configurar els perifèrics del sistema, realitzar una comprovació bàsica del sensor d'inèrcia, llançar el menú del sistema i, en cas d'una petició d'interrupció, guardar els caràcters rebuts per la UART.





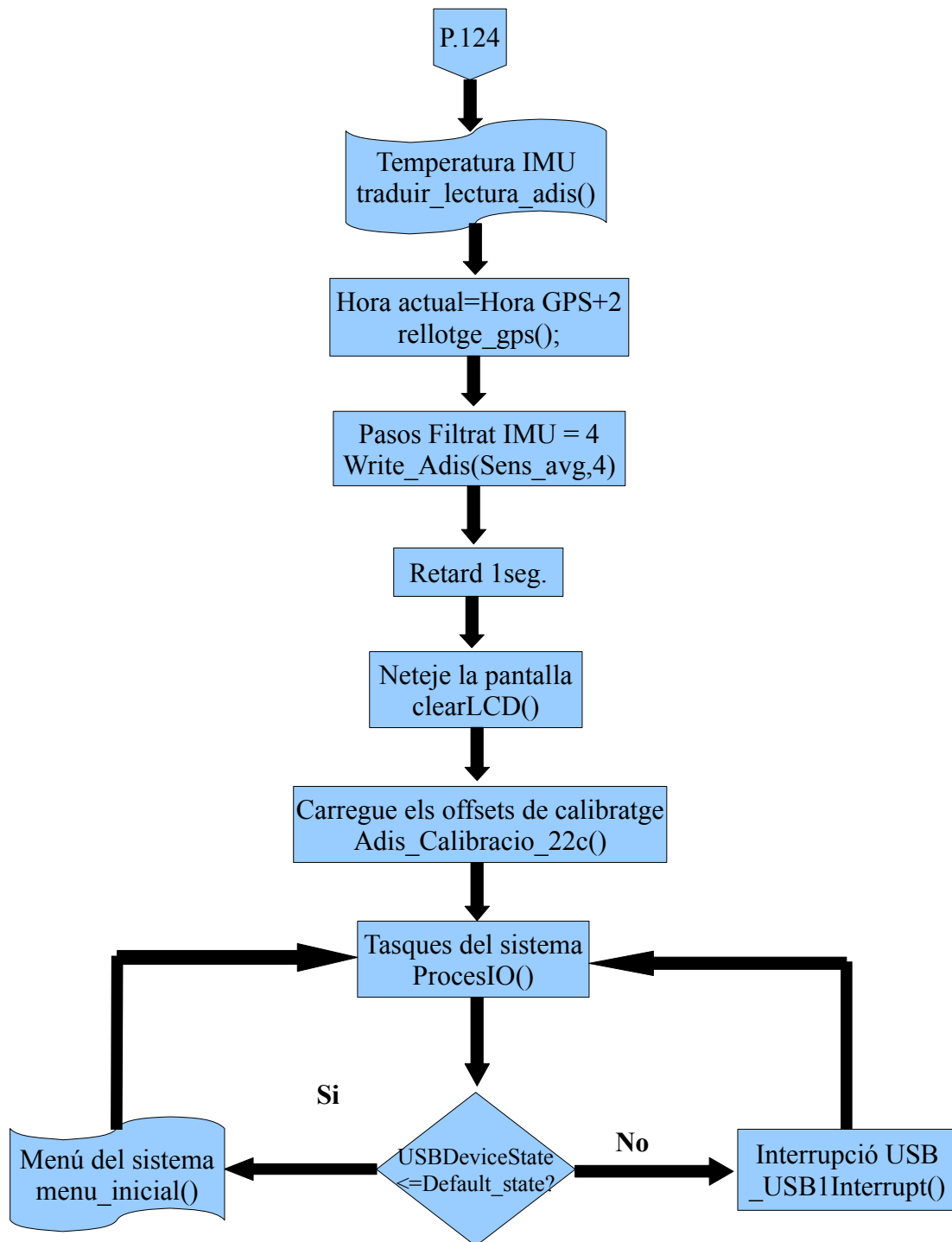


Figura 116: Diagrama de flux de *main.c*

Com podem veure en el diagrama de flux corresponent, abans de fer qualsevol cosa, escrivim els valors de les paraules de configuració del PIC. Aquesta acció és imprescindible abans de fer res ja que estes indiquen al PIC, entre altres coses, quins seran els terminals que s'utilitzaran per a la programació en circuit o quina serà la font d'oscil·lació principal. Òbviament aquest últim paràmetre és vital ja que configurar-lo d'una o altra manera no sols marcarà la diferència entre que el sistema funcione o no, sinó que a més, determinarà la velocitat interna de treball del microcontrolador i dels seus perifèrics. Nosaltres no tinguérem en compte aquest últim detall i vam tenir que refer algunes parts del programa quan vam habilitar el bloc PLL del USB en la part final del projecte.

Totes les altres accions que es realitzen en aquest fitxer són rutinàries i bàsicament es limiten a fer un inici del sistema ordenat i a deixar configurat la UART per a que funcione a 9600 bits per segon amb un bit de parada però sense control de flux i el mòdul SPI1 funcione a la màxima velocitat de rellotge admesa per l'ADIS16400: 2MHz.

## 9.2- Fitxer *menus.c*

Una vegada s'ha acabat d'inicialitzar el sistema i s'ha comprovat que el sensor inercial funciona, la funció *main()*, mentre no es produïska una connexió o desconnexió del bus USB, es dedicarà a cridar la funció que s'encarrega de mostrar el menú inicial del sistema, llegir el teclat i actuar d'acord les ordres d'aquest.

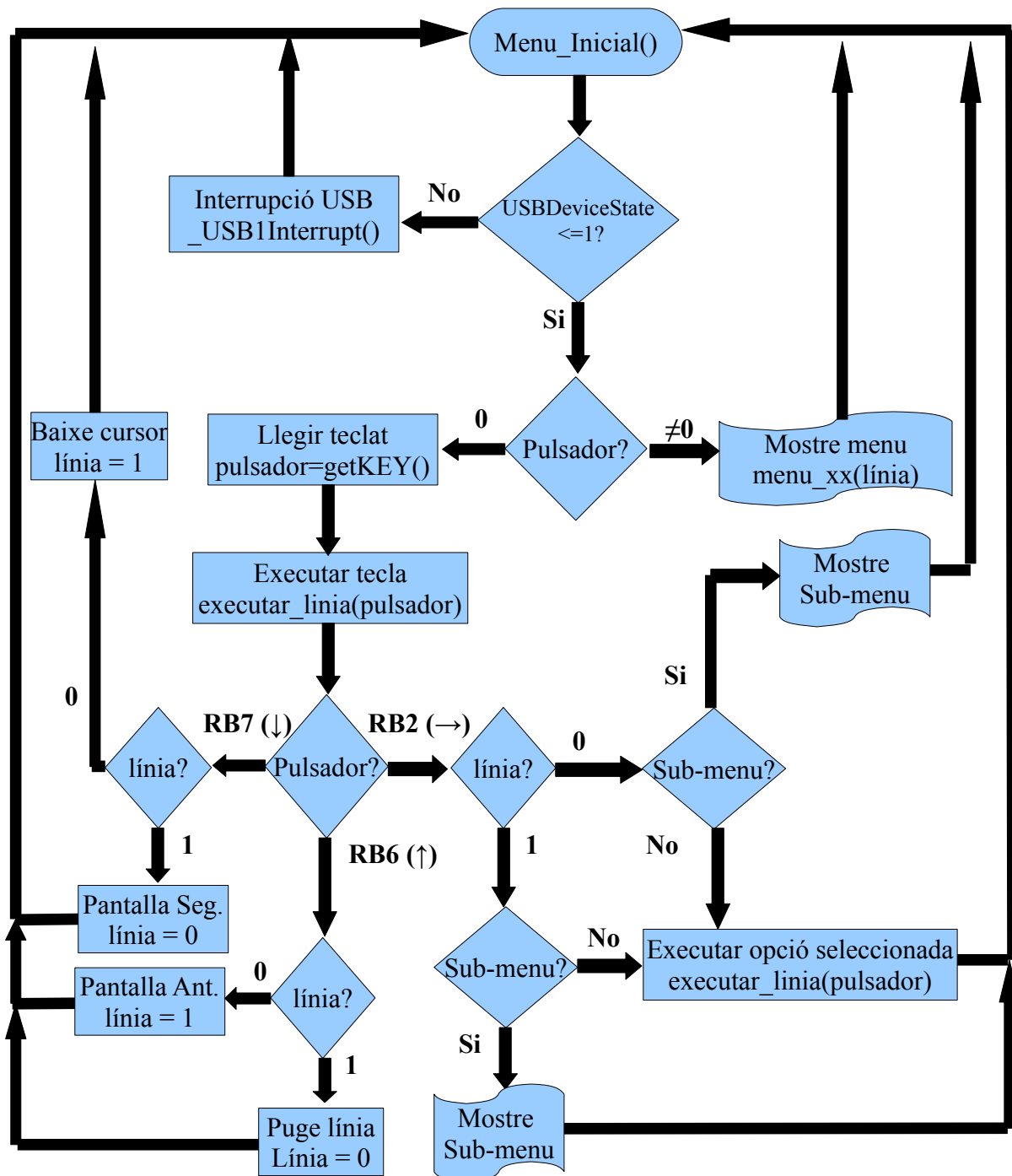


Figura 117: Diagrama de Flux bàsic de *menus.c*

Al no tenir pensat desenvolupar un sistema de control basat en menús, el funcionament d'aquesta etapa és un tant caòtic i està molt poc optimitzat perquè hem tingut que arreglar-nos-les per a poder mostrar tot el que ens interessava en dos línies de text. Així doncs, per a no complicar el diagrama de flux, hem obviat algunes parts.

Sí parem atenció a la figura 117 veurem que, a part de la rutina d'atenció a la interrupció del bus USB, sols anomenem tres funcions diferents: *getKEY()*, *menu\_xx()* i *executar\_linia()*. De les tres funcions, la més important és *executar\_linia* ja que, quan es pressiona fletxa amunt (RB6) o fletxa avall (RB7), esta funció mou el cursor del LCD al mateix temps que relaciona la posició del cursor amb la funció que s'executara quan es pressione la tecla OK (RB2). Tot açò és possible gràcies a *getKEY()* i a *executar\_linia()*, pues mentre la primera li facilita la tecla que s'ha pressionat, la segon s'encarrega de mostrar en pantalla el menú que se li indique. Destacar que, a pesar que estem considerant *menu\_xx()* com a una única funció, en realitat és un grup de funcions d'estructura similar però que mostren text per pantalla que no té res a veure amb les altres funcions *menu\_xx()*.

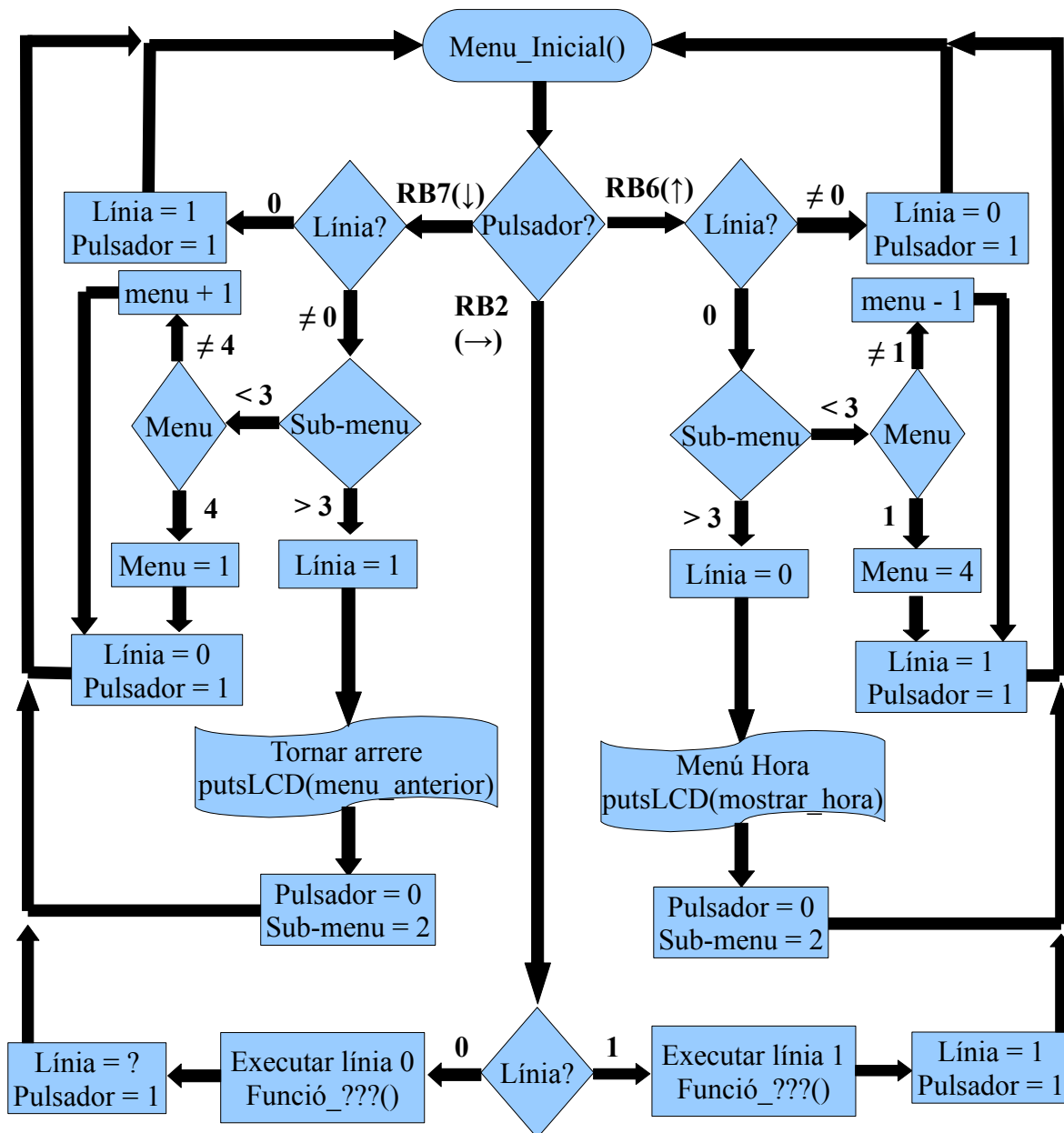


Figura 118: Diagrama de flux simplificat d'executar\_linia()

Tal i com podem veure en la figura 118, mentre no es pressiona el pulsador RB2, el microcontrolador anirà actualitzant internament la posició del cursor i, sí és necessari, presentarà en

pantalla un menú diferent al que s'estava visualitzant. En el moment en el que es pressiona RB2, el microcontrolador executarà una funció que dependrà tant de la línia en la que es trobe com del menú actual. En acabar la funció, el microcontrolador modificarà el valor de la variable *línia* i refrescarà la pantalla per tal de mostrar el darrer menú que vam visualitzar i deixar el cursor en la línia correcta.

Tot i que es pot pensar que són les funcions *menu\_xx()* les que executen les funcions que mostren per pantalla, en realitat no passa d'eixa manera ja que una vegada s'ha pressionat RB2 sobre alguna de les opcions que mostren, és la funció *executar\_linia()* qui cridarà a la funció indicada per la posició del cursor. A més, sí la funció a executar necessita algun paràmetre de configuració, s'encarregarà de demanar-lo a l'usuari i adaptar l'entrada als requisits de la funció a la que es va a cridar. Pot ser hauria sigut més lògic afegir esta part de configuració dins de les mateixes funcions a les que crida *executar\_linia()*, però per a simplificar la neteja d'estes funcions en cas de prescindir de la pantalla, s'ha preferit fer-ho d'aquesta manera.

### 9.2.1- Organització bàsica del menús del sistema

Una vegada descrit breument el funcionament d'aquesta part del *firmware*, passarem a parlar dels diferents menús que ens trobarem al desplaçar-nos per la pantalla principal utilitzant els polsadors RB6 (↑) i RB7(↓).

Degut a que sols podem mostrar dos línies al mateix temps, el menú principal es divideix en quatre parts o pantalles. L'organització d'aquestes és tal, que al arribar a l'última línia del últim menú, sí continuem desplaçant-nos cap avall, apareixerem sobre la primera línia del primer menú i al revés, és a dir, sí estem en la primera línia del primer menú i ens desplaçem cap amunt, apareixerem en l'última línia del últim menú.

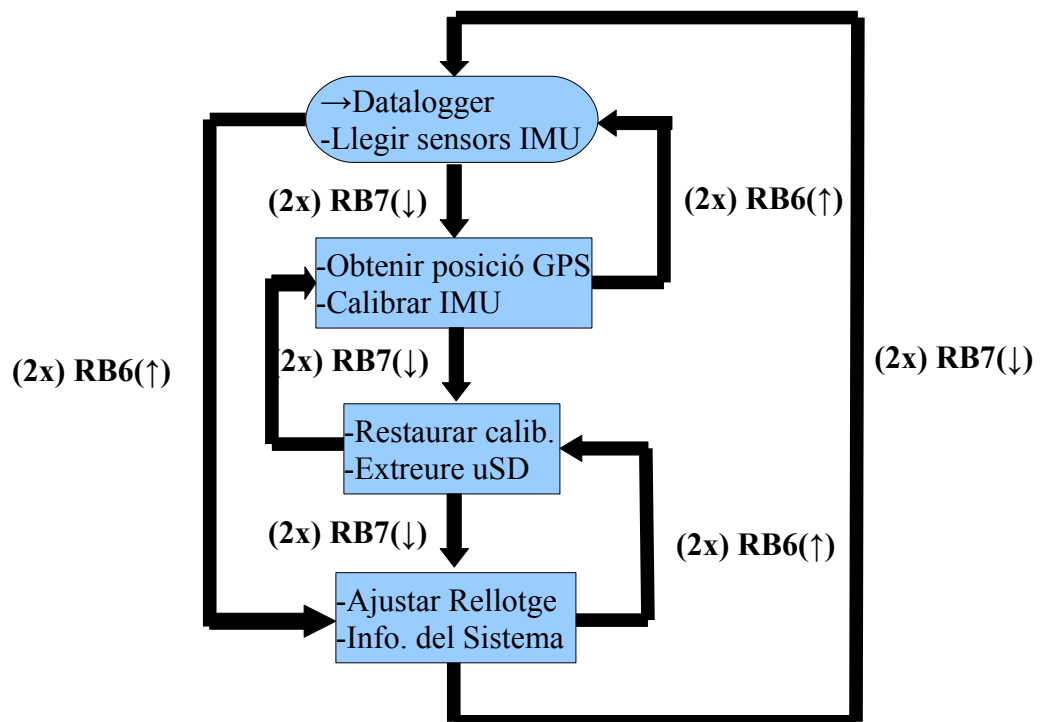


Figura 119: Estructura del menú principal



### 9.2.2- Menú *Datalogger* i Llegir IMU

Com hem vist en la figura 119, tan prompte com es carregue el primer menú podrem triar entre les dos funcions més importants del sistema: la de registrador de dades i la de llegir en temps real el sensor d'inèrcia. El funcionament d'estes dos funcions és en essència el mateix, però el fet que la funció *datalogger* guarde en la targeta SD la posició GPS del sistema defineix una de les diferències entre les dos: la freqüència de lectura dels sensors. Açò es deu a que, per defecte, el receptor GPS envia la informació d'estat cada segon, pel que sí no deshabilitem la funció de registrar la posició actual, en cas que el *buffer* de la UART no continga la frase GPRMC, endarrerirem involuntàriament almenys un segon la pròxima lectura de la IMU, així que per a evitar-ho, la freqüència de mostreig mínima en esta funció és de dos segons. L'altra diferència, com ja haurem suposat, és el fet en sí de guardar les lectures dels sensors en la targeta Micro-SD, pues en cas de no detectar-se'n una, s'avisarà a l'usuari i sí, després de demanar-li que la introduïsca en el lector, torna a fallar la detecció, s'avortarà la funció i serem tornats al menú principal.

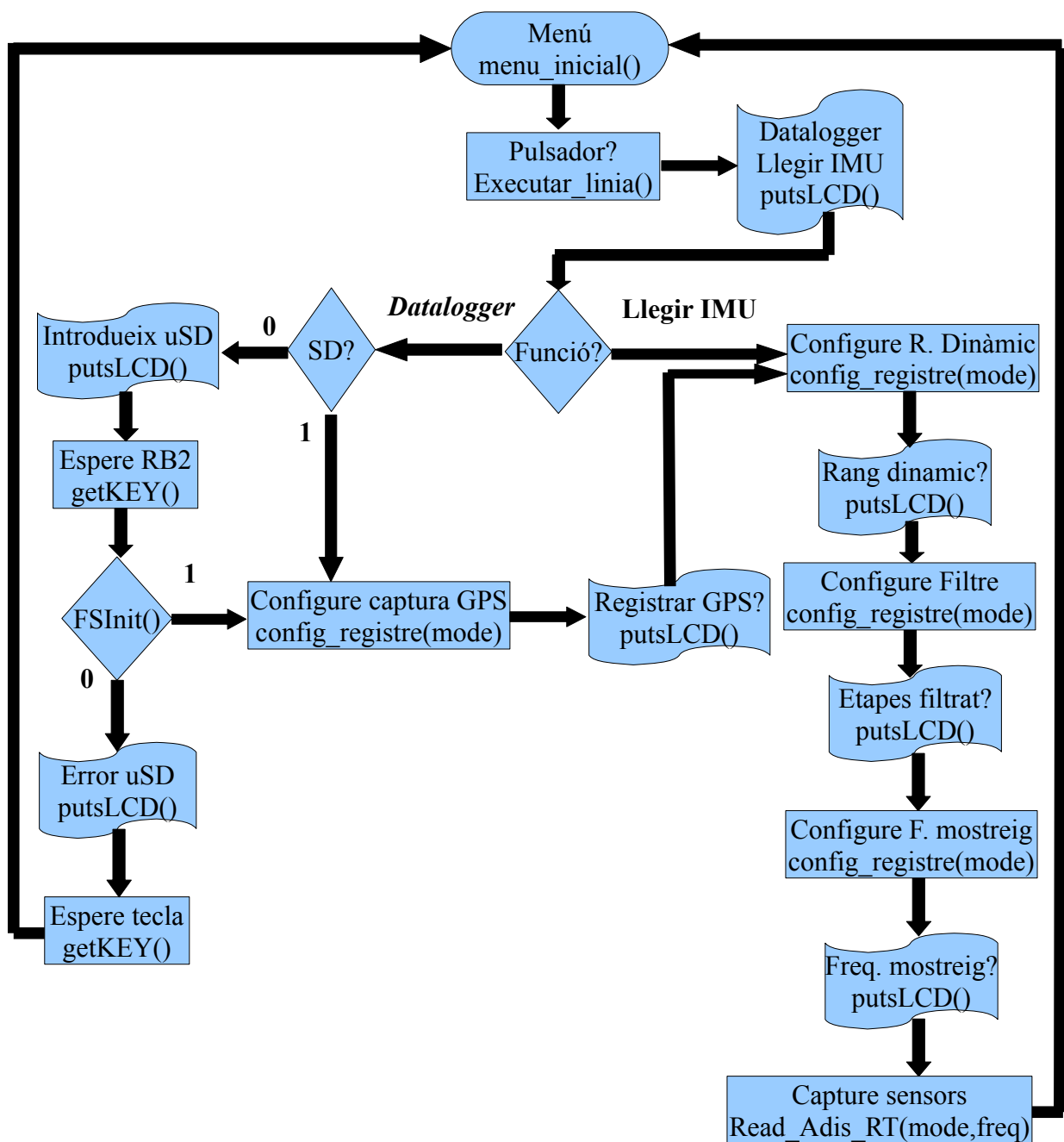


Figura 120: Diagrama de flux simplificat de les opcions de sistema *Datalogger* i Llegir IMU

En el diagrama de flux de la figura 120 podem veure com esta funció és la que s'encarrega de cridar les funcions necessàries per a configurar la funció seleccionada i, en acabar, de cridar la funció que realitzarà la tasca seleccionada. Per a no complicar internament la funció, aquesta utilitza la funció *config\_registre()* per a mostrar els missatges que guien a l'usuari en el procés de configuració dels modes *Datalogger* i *Llegir IMU*. En acabar, el valor retornat per la funció, per norma general, s'utilitzarà com a paràmetre d'entrada de la funció encarregada d'escriure en el sensor d'inèrcia.

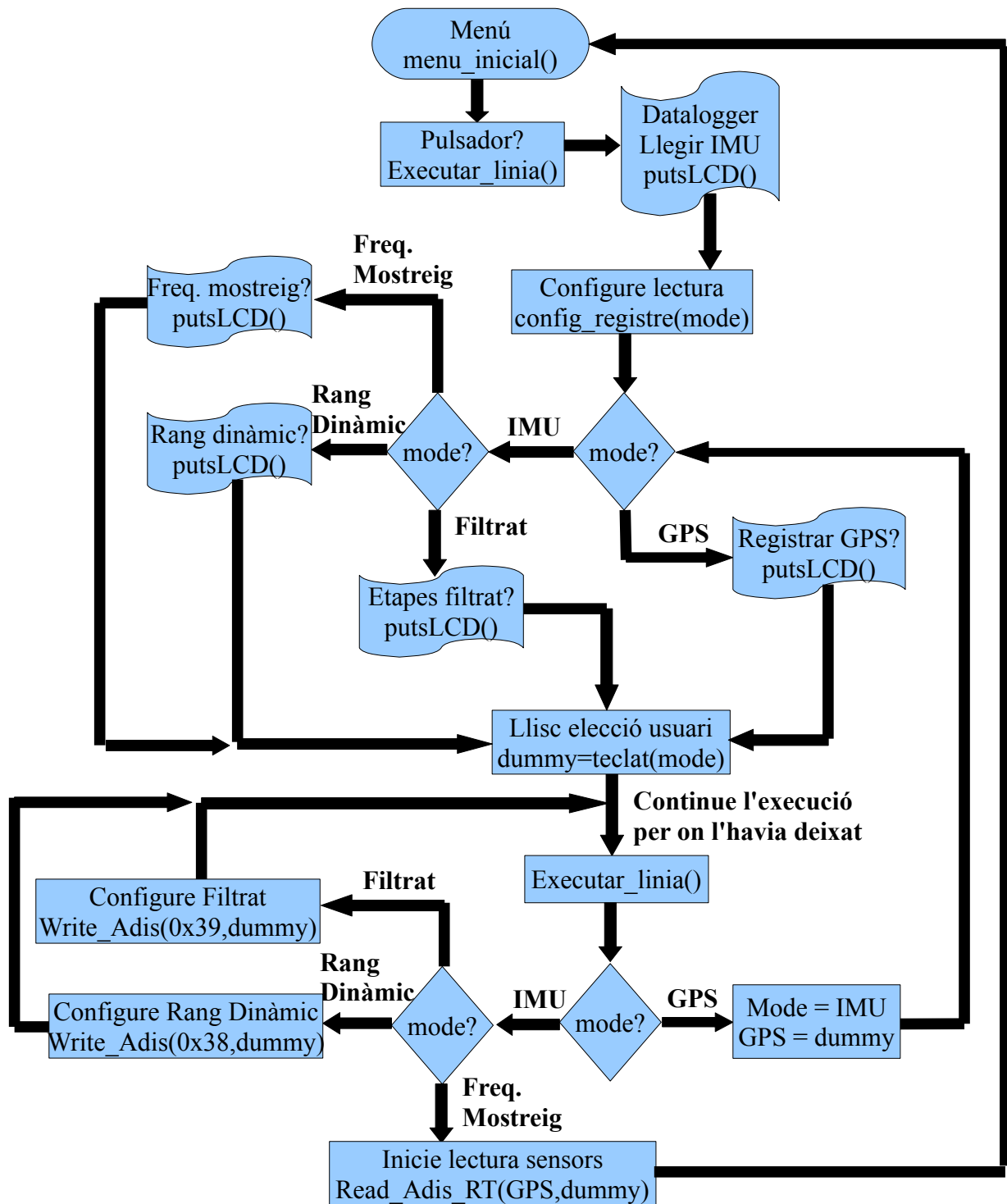


Figura 121: Diagrama de flux simplificat de *config\_registre()*

Tal i com es dedueix del diagrama de flux de la figura 121, la funció *config\_registre()* mostra els missatges de configuració per pantalla mentre que, per a capturar l'entrada del teclat, s'utilitza la funció *teclat(mode)*. Esta funció, a més, s'encarrega d'adaptar i limitar les pulsacions de RB6 i RB7

de manera que el valor *dummy* retornat per la funció pugui ser utilitzat posteriorment com a paràmetre d'entrada d'una funció de configuració. Esta característica és la responsable de, entre altres coses, impedir que l'usuari seleccione rangs dinàmics que no permet l'ADIS16400, la que limita les etapes de filtrat a 6 a l'hora de configurar esta etapa o la que limita la freqüència de mostreig a un màxim de 99 segons.

### 9.2.3- Menú Obtenir posició GPS i Calibrar IMU

Aquestes dos funcions, com el nom suggereix, permeten obtenir la posició GPS i iniciar el procés de calibratge dels sensors de l'ADIS16400. Si ens fixem en la figura 122 apreciarem la simplicitat d'aquest menú.

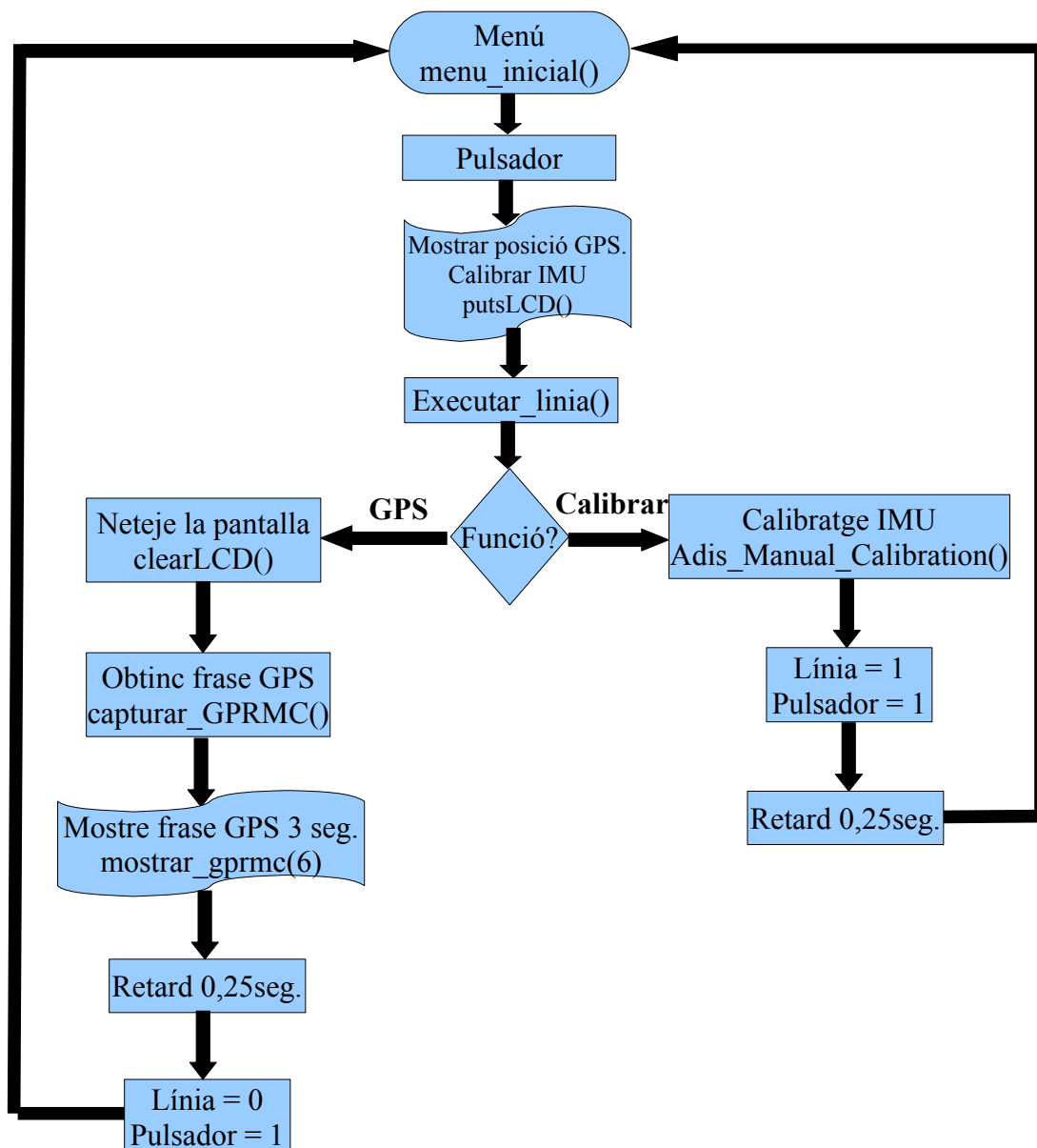


Figura 122: Diagrama de flux del menú Obtenir posició GPS i Calibrar IMU

Tal i com ja havíem avançat, es tracta d'un menú molt simple ja que bàsicament es limita a cridar una funció externa i actualitzar tant la posició interna del cursor com la pantalla. A més, per a evitar algun rebot que no ha sigut filtrat per la funció *getKEY()*, afegim un petit retràs de 250ms, pues d'esta manera també s'evita tornar a carregar per accident l'última funció realitzada.

### 9.2.4- Restaurar calibratge de la IMU i Menú Extraure/Muntar uSD

Tot i que no és imprescindible poder restaurar la calibratge del sensor inercial o poder deshabilitar el lector de targetes, sí és interessant disposar d'aquestes opcions perquè amb la primera opció ens evitem fer un *reset* al sistema mentre que amb la segon ens assegurem que podem extreure la targeta SD del sistema sense cap perill.

Al igual que en el menú anterior, el diagrama de flux d'aquest menú no és excessivament complicat:

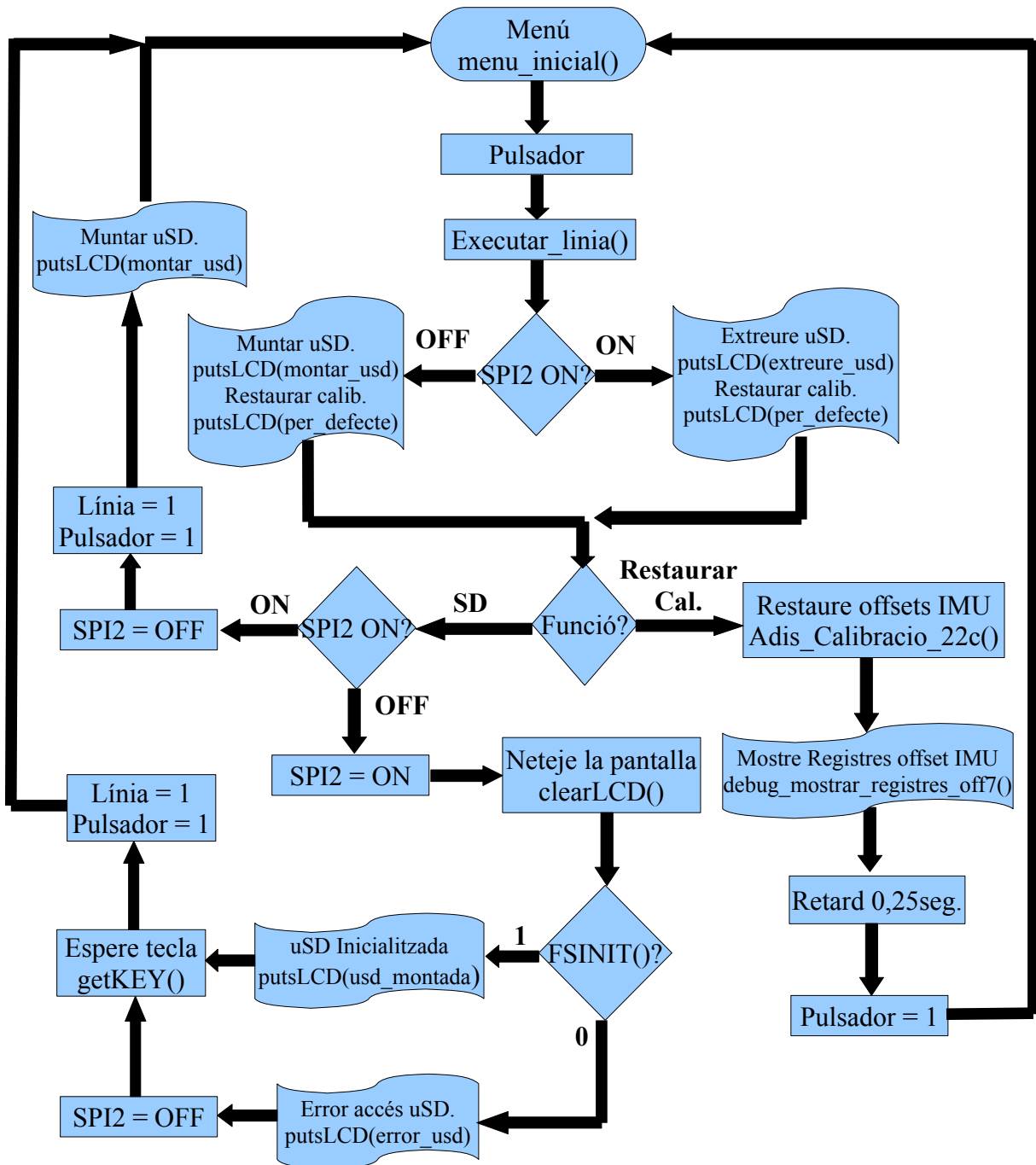


Figura 123: Diagrama de flux de Restaurar calibratge IMU i Extreure/muntar uSD

## 9.2.5- Ajustar RTCC i Informació del Sistema

L'última pantalla del menú ens permet ajustar l'hora i el calendari del sistema o veure informació bàsica del sistema com pot ser la IMU utilitzada, configuració de la mateixa, tipus de targeta micro SD, etc.

Al contrari que passa en els altres menús, el diagrama de flux d'aquest menú és un poc més enrevessat perquè hem habilitat dos opcions dins del menú de configurar el RTCC que no estan a la vista fins que ens desplaçem pel menú. Estes dos opcions són la de poder tornar al menú anterior i la de mostrar hora i calendari. Per a accedir a mostrar *Hora i calendari* sols cal pressionar RB6 en la pantalla inicial de *Configurar RTCC*, mentre que per a mostrar l'opció de *Menú anterior* sols cal desplaçar-se cap a baix fins que aparega la citada opció.

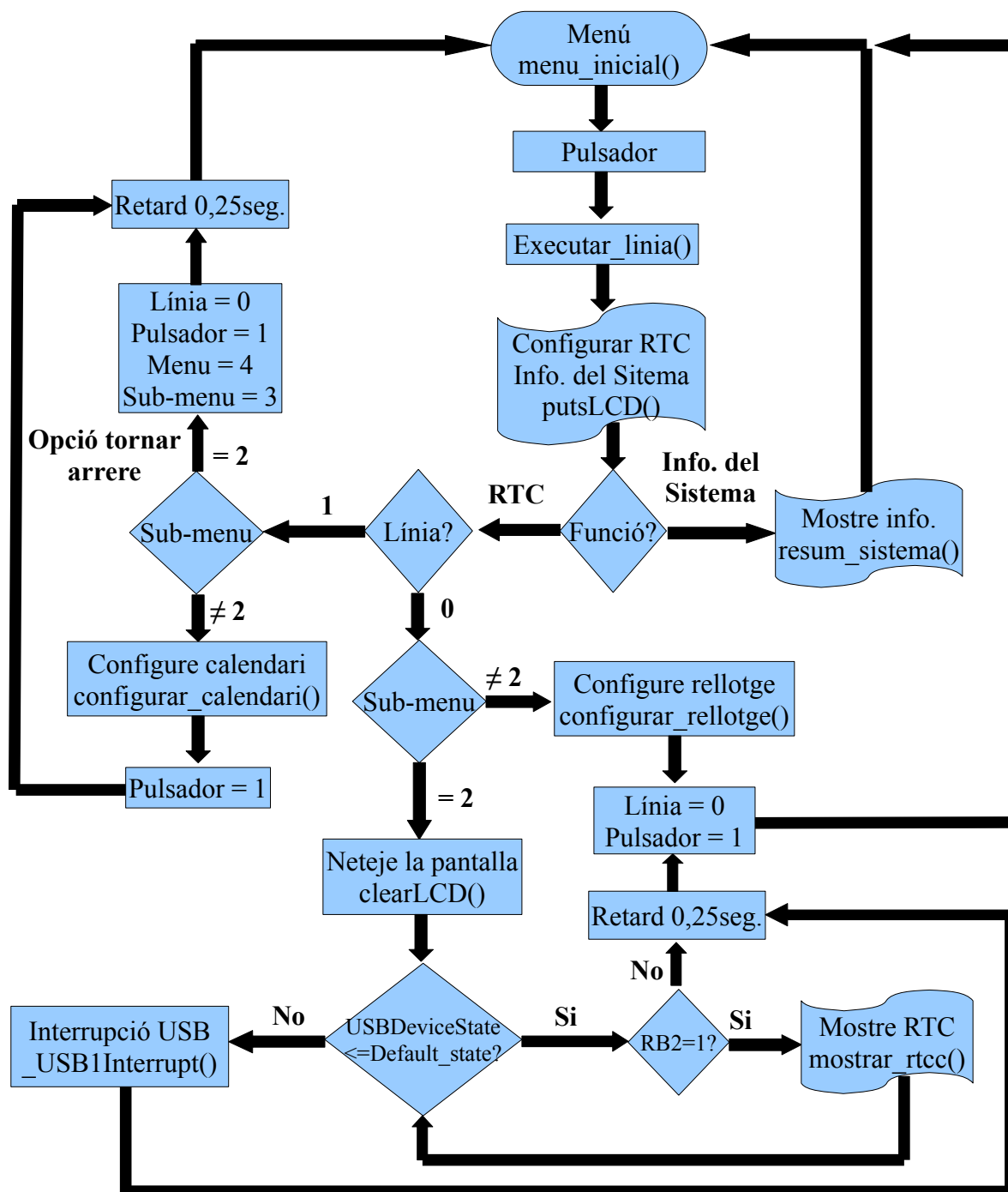


Figura 124: Diagrama de flux d'Ajustar RTCC i Informació del Sistema

Tot i que en el diagrama de flux anterior no s'aprecia, aquest menú té la particularitat de no refrescar tota la pantalla utilitzant una funció *menu\_xx()*, sinó que és la mateixa funció *executar\_linia()* la que escriu en la línia corresponent el text *Menú anterior* o *Mostrar Rellojge*, mantenint intacta l'altra línia.

### 9.3- Fitxer *lcd.c*

Aquest fitxer conté les funcions de manipulació bàsiques del LCD. Com el codi utilitzat no és nostre i la pantalla és un element opcional en el sistema, sols descriurem la part que hem modificat per a adaptar-la a les nostres necessitats: el procés d'inicialització

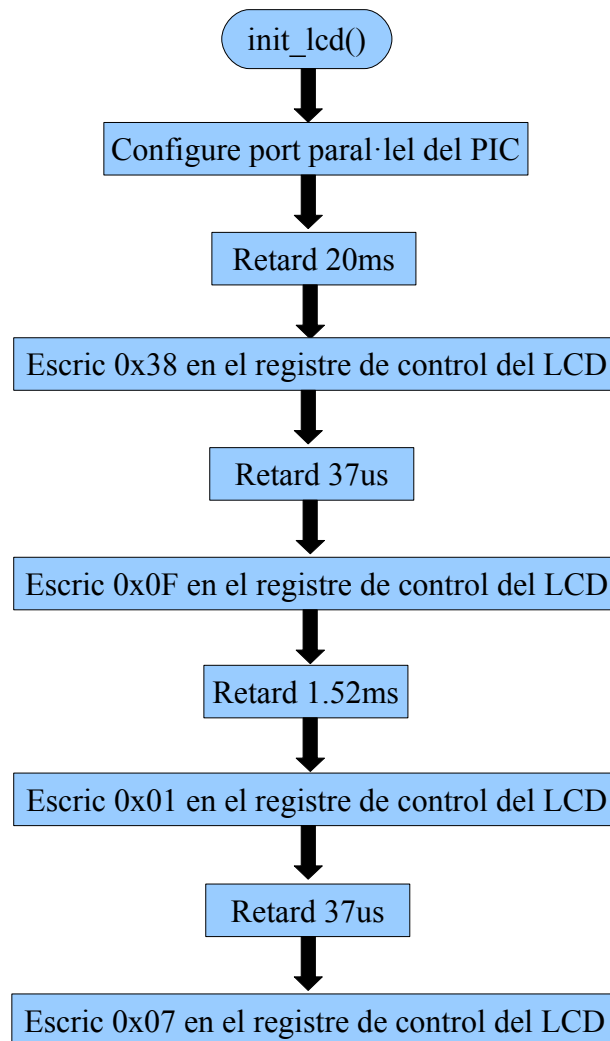


Figura 125: Diagrama de flux de la funció d'inicialització del display LCD.

Com podem apreciar en el diagrama de flux de la figura 125, la inicialització del LCD és tan simple com escriure un dada en el port paral·lel, esperar un cert temps i repetir fins que tinguem configurada la pantalla al nostre gust. Estos temps d'espera varien d'un controlador de *display* a un altre ja que corresponen als temps d'execució dels diferents funcions internes del controlador, pel que han de ser ajustats als valors especificats pel fabricant del *display*. En el nostre cas hem utilitzat els temps necessaris per a configurar la pantalla de manera que utilitze un bus de 8 bits, dos línies amb caràcters de 5x8 amb cursor parpellejant i desplaçament automàticament<sup>57</sup>.

<sup>57</sup> Aquests temps els podem trobar en la taula d'instruccions del manual de programació del controlador del *display*.

Una vegada hem inicialitzat la pantalla, sí volem escriure un caràcter sols hem d'utilitzar la funció *putLCD(caràcter)*. En canvi, sí volem escriure una paraula o cadena de caràcters, haurem de recórrer a *putsLCD(paraula)*. Aquestes funcions són molt bàsiques i no tenen en compte la posició del cursor ni la llargària de la paraula a mostrar, pel que, com no les hem modificat, a l'hora d'utilitzar-les hem tingut que anar en compte per a no menjar-nos accidentalment caràcters de la paraula.

Per a refrescar la pantalla, per norma general no hem corregit la línia o caràcters que contenen informació invàlida, sinó que hem optat per esborrar-la completament amb la macro *clearLCD()* i després escriure-la amb *putsLCD()*.

A part de modificar la seqüència d'inicialització de la pantalla, com hem utilitzat la pantalla més del que ens pensàvem, vam tenir que afegir tres macros:

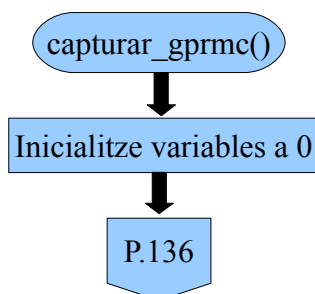
- ***LCD\_L2\_home()***: esta macro posiciona el cursor en l'inici de la segon línia. Principalment l'hem utilitzat en el fitxer menú per a anar sincronitzant el moviment del cursor amb les pulsacions de RB6 i RB7. Equival a escriure *writeLCD(LCDCMD,0xC0)*.
- ***cursor\_arrere()***: és l'equivalent d'escriure *writeLCD(LCDCMD,0x10)* i és útil per a corregir el caràcter anterior al cursor, raó per la que l'hem utilitzat a l'hora de canviar un caràcter o opció en pantalla al prémer RB6 o RB7.
- ***cursor\_avant()***: aquesta macro ens permet avançar el cursor una posició. Això ens ha permès, per exemple, en la funció de configuració del rellotge, avançar al següent camp al detectar la pulsació de RB2. Una altra manera de fer el mateix seria amb l'ordre *writeLCD(LCDCMD,0x14)*.

#### 9.4- Fitxer *gps.c*

Dins del fitxer *gps.c* podem trobar les funcions relacionades amb la manipulació de la informació proporcionada pel receptor GPS. Al no haver alterat la configuració per defecte del receptor, aquest ens envia periòdicament les frases GPGGA, GPGSA, GPGSV, GPRMC i GPVTG. Teòricament hauríem de capturar-les totes, però com ja vam veure en la secció 2.6.1 sobre les frases NMEA, en la pràctica, o almenys per als nostres interessos, hi ha prou amb capturar la frase GPRMC i després separar els camps per a obtenir la informació de longitud i latitud.

##### 9.4.1- Funció *captura\_gprmc()*

Aquesta és la funció més bàsica del fitxer. S'utilitza per a buscar dins del *buffer* de recepció de la UART la frase GPRMC. Una vegada ha localitzat la frase, comprova que estiga completa i, de ser així, comença a separar els camps fins que acaba o troba un error.



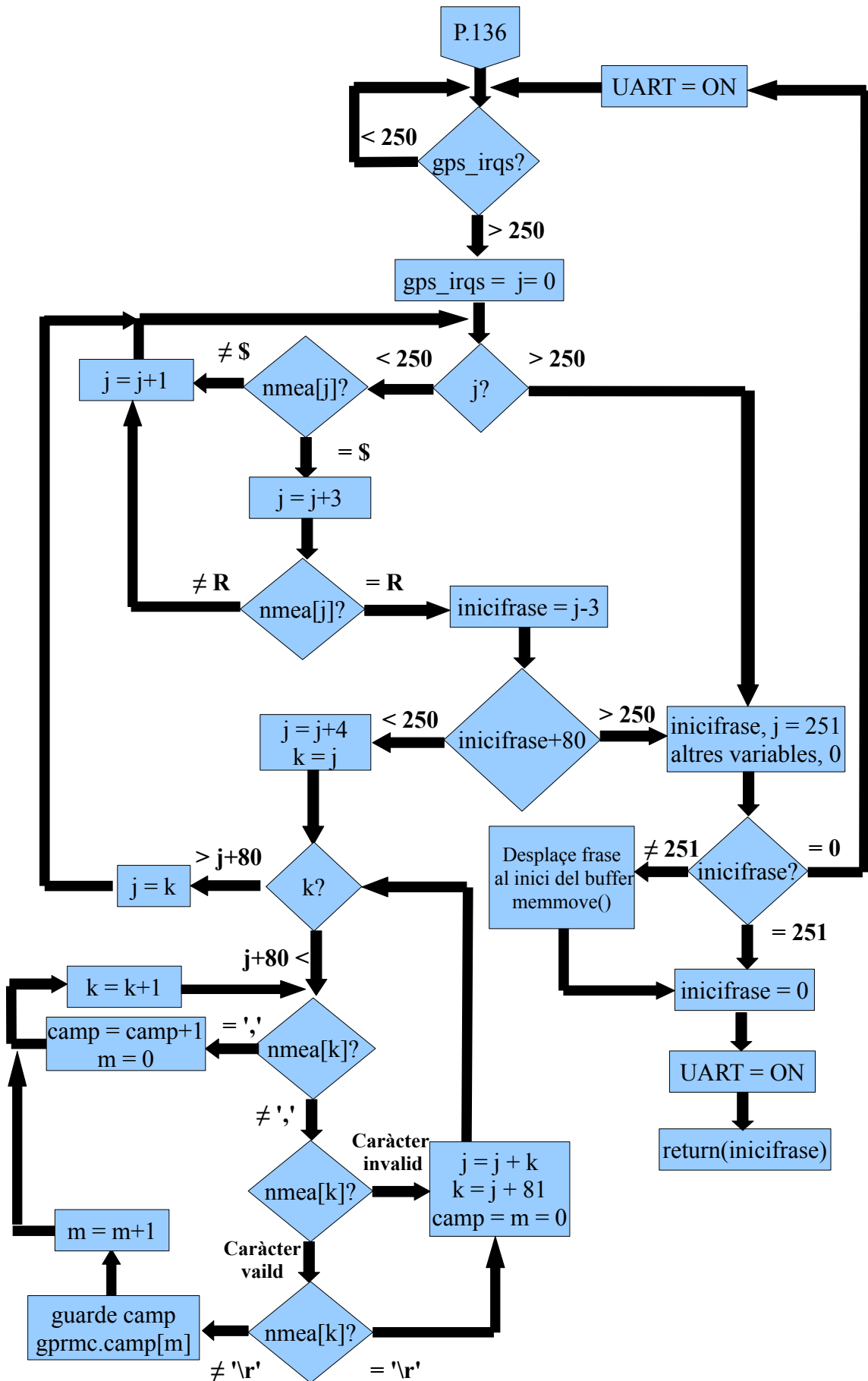
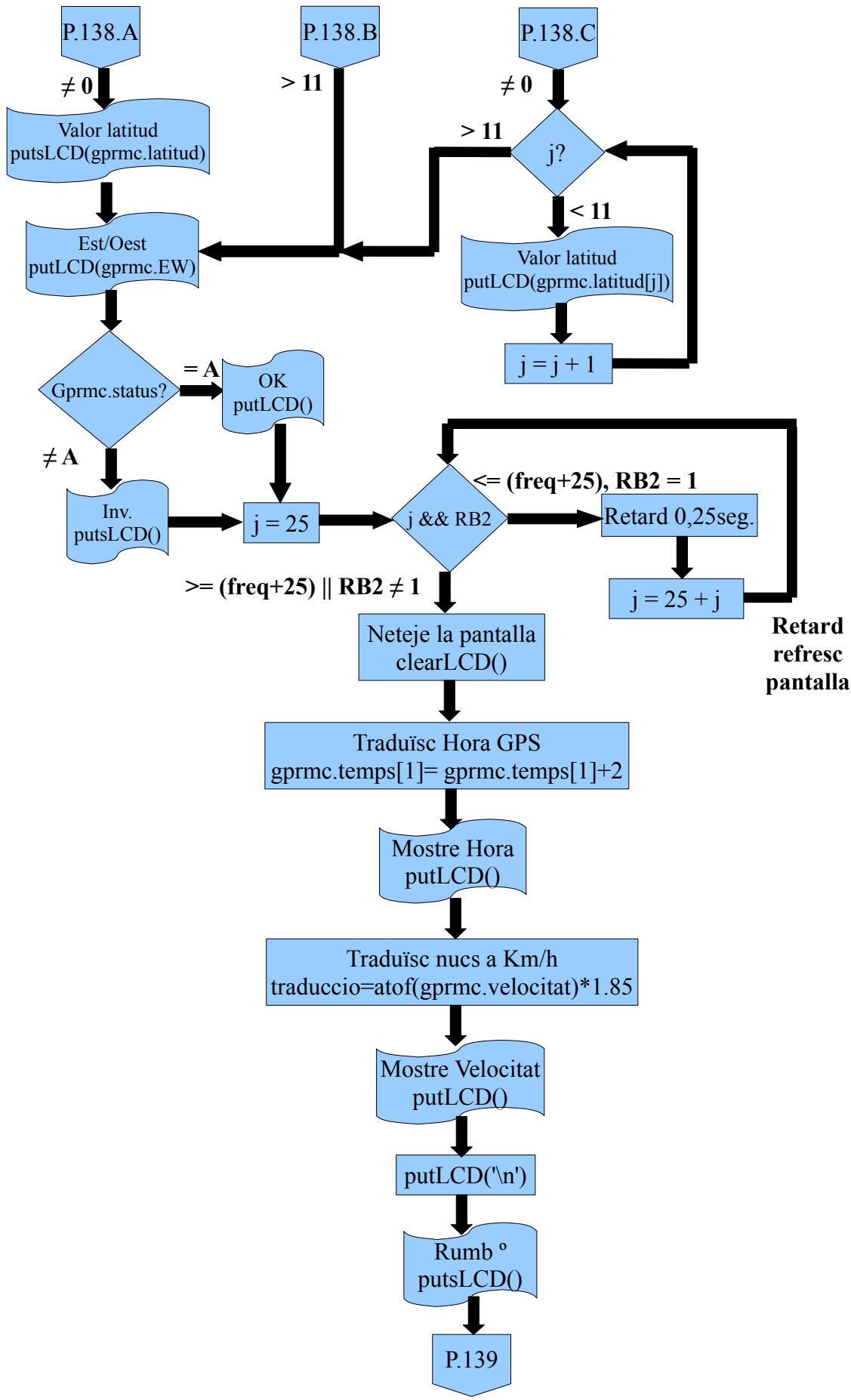


Figura 126: Diagrama de flux simplificat de *captura\_gprmc()*







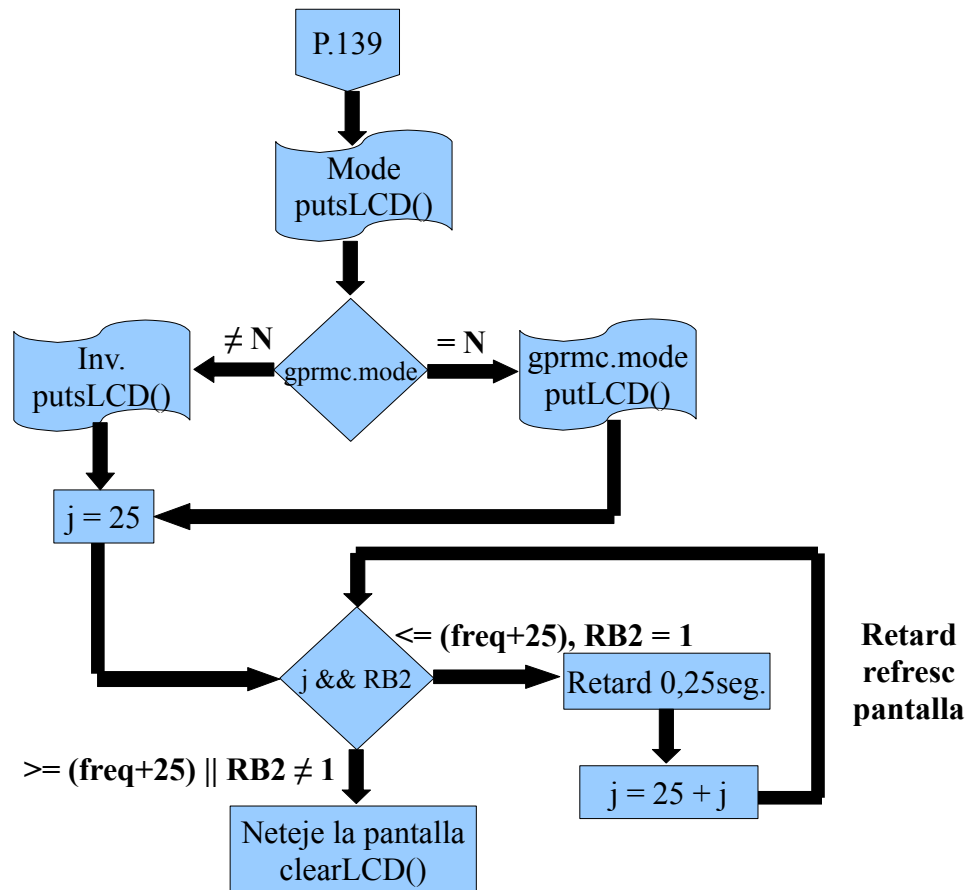


Figura 127: Diagrama de flux simplificat de *mostra\_gprmc()*

En el diagrama de flux de la figura 127 podem veure com la quantitat de línies i caràcters que pot mostrar la nostra pantalla ha complicat la realització d'aquesta funció, pues no sols ens ha limitat la quantitat de temps que apareix la informació en pantalla, sinó que a més ens ha obligat a eliminar caràcters “absurds” com els zeros que poden aparèixer al principi del *array* que conté el valor de la longitud o la latitud. A més, indirectament, estes limitacions també afecten a la velocitat de mostreig, pues resulta absurd mostrejar el sensor a menys d'una vegada per sensor perquè a esta velocitat sols disposem de 0,25 segons per a veure la informació del sensor, pel que a més velocitat, més complicat serà llegir en pantalla la lectura.

### 9.4.3- Funció *rellotge\_gps()*

La funció *rellotge\_gps()* és una funció auxiliar creada amb la intenció d'estalviar-li a l'usuari la feina de posar en hora el sistema. Açò és possible gràcies a que la frase GPRMC conté un camp que indica l'hora en format UTC, pel que afegint a este camp la diferència amb el nostre ús horari, obtindrem l'hora actual.

En la nostra aplicació sols utilitzem esta funció en la inicialització del sistema, però perfectament la podríem arreglar-nos-les per a que s'executarà cada cert temps i d'eixa manera millorar la precisió horària o corregir-la, pues el rellotge intern del mòdul RTCC, degut a les seues característiques, necessita ser calibrat prèviament sí volem millorar la seva exactitud al llarg del temps.

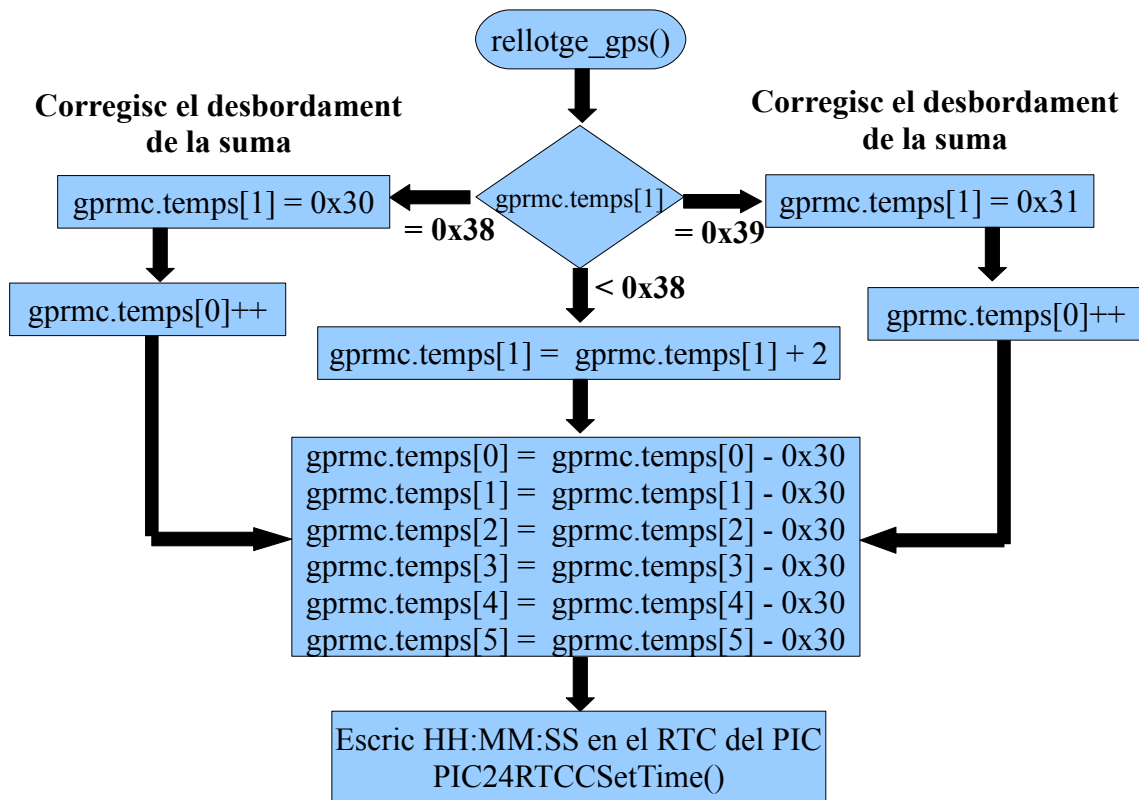


Figura 128: Diagrama de flux de *relloige\_gps()*

A l'igual que passa amb *mostrar\_gprmc()*, esta funció necessita que el *struct gprmc* haja sigut inicialitzat prèviament, pues de no estar-ho, el comportament de la funció serà impredecible. Per últim, destacar que, sí ens oblidem de la funció *PIC24RTCCSetTime()* i del bloc de restes, el diagrama de flux de la figura 128 seria l'equivalent al bloc de traducció d'hora GPS del diagrama de la figura 127.

#### 9.4.4- Rutina d'atenció a la interrupció de la UART

Tot i que la part encarregada d'atendre la interrupció que es produeix cada vegada que arriba un caràcter nou a la UART del microcontrolador està en el fitxer *main.c*, parlarem d'ella en esta secció aprofitant que està relacionada amb el receptor GPS.

Tornant arrere, concretament al diagrama de flux de la funció *captura\_gprmc()*, veurem que, just abans d'acabar la funció, habilitem la UART tot i que en cap moment la hem deshabilitat. Per què fem açò? Doncs per què una vegada s'ha omplert el *buffer* de recepció de la UART, deshabilitem el mòdul per a que el seu contingut siga coherent, pel que una vegada hem revisat el contingut, si volem tornar a rebre dades, l'hem d'activar. El motiu d'anar connectant i desconnectant la UART no és altre que aprofitar els temps morts entre refresc i refresc de la pantalla per a capturar els caràcters que ens envia el GPS i, d'eixa forma, disminuir el temps que tarda en trobar la funció *captura\_gprmc()* una frase GPRMC lliure d'errors.

Destacar que, de no haver utilitzat les interrupcions de la UART per a capturar les dades del GPS, no hauríem aconseguit registrar en la targeta micro-SD les dades del receptor GPS i de la IMU a la velocitat desitjada per l'usuari, ja que en el pitjor dels casos, el temps entre ompliment i ompliment del *buffer* de la UART pot arribar a ser d'uns quants segons.

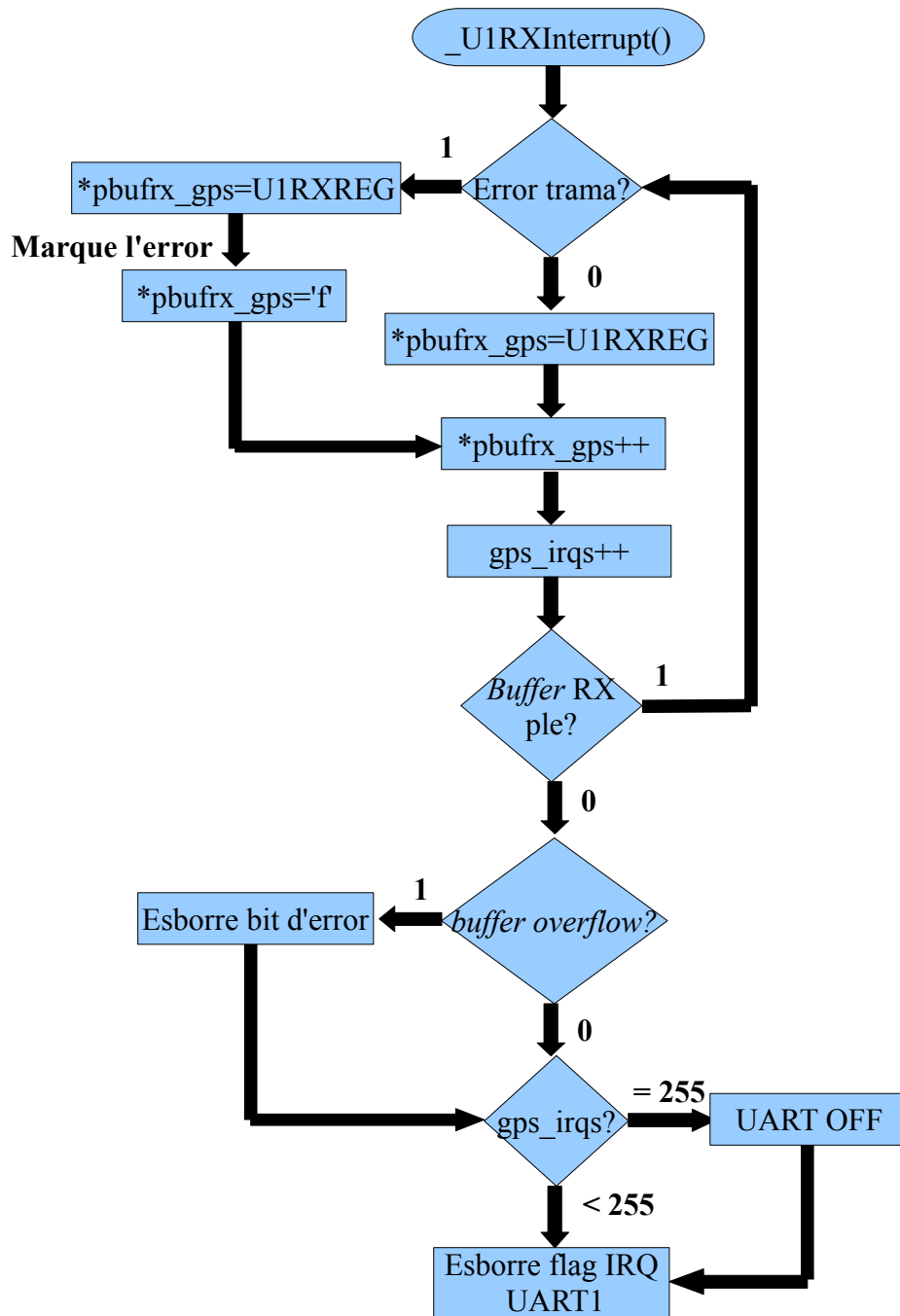


Figura 129: Diagrama de flux de la rutina d'atenció a la interrupció de la UART 1.

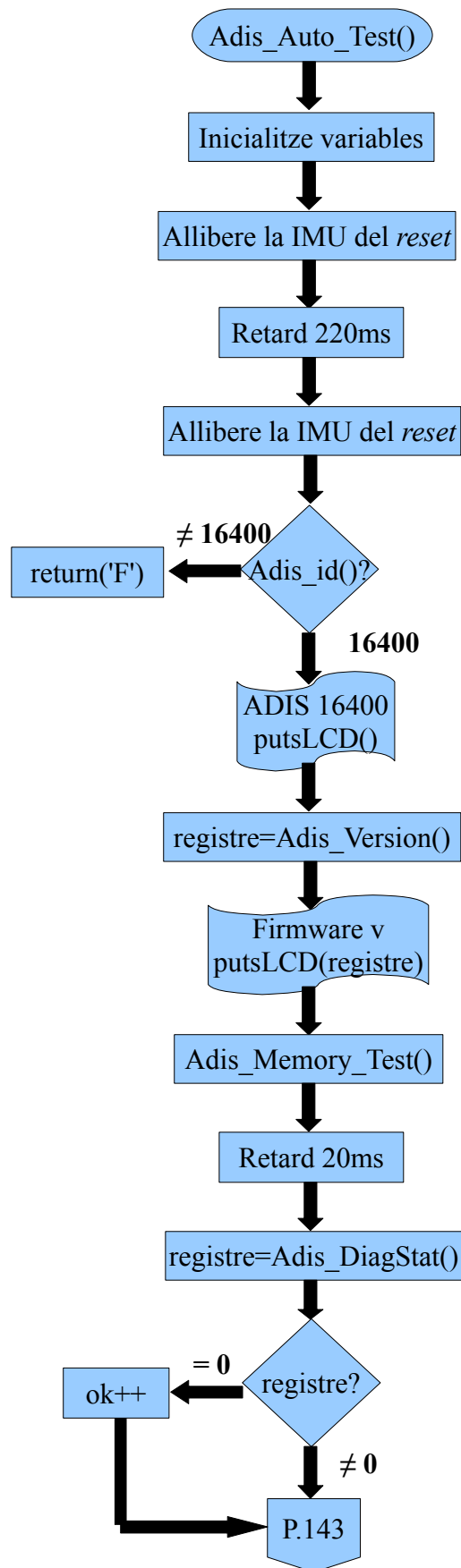
Degut als errors de silicona del PIC utilitzat, per tal d'evitar possibles problemes, hem de llegir els bits d'error de la UART dins de la rutina d'atenció a la interrupció. A més, en el nostre cas, com tenim errors de trama esporàdicament, aprofitem el moment en el que llegim el flag corresponent per a marcar l'error en el *buffer* i, d'esta manera, quan la funció *capturar\_gprmc()* l'examine, sàpia que la frase està corrupta i que no deu perdre més temps examinant-la.

### 9.5- Fitxer *adis16x00.c*

Dins d'aquest fitxer es troben les funcions encarregades de gestionar la comunicació amb el sensor d'inèrcia, comprovar la seva integritat, enviar per pantalla les lectures obtingudes i guardar-les en la targeta Micro-SD junt amb la frase GPRMC.

### 9.5.1- Funció Adis\_Auto\_Test()

Tan prompte com s'han configurat els diferents perifèrics interns del microcontrolador, el primer que fa és inicialitzar la IMU i comprovar que aquesta funcione correctament.



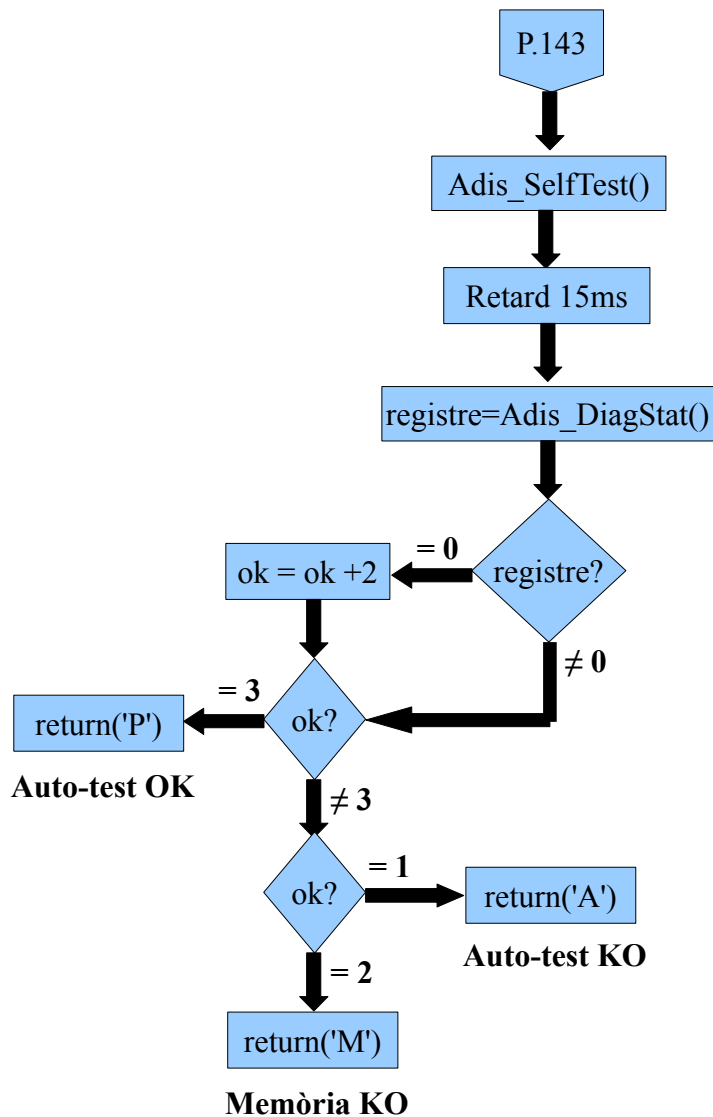
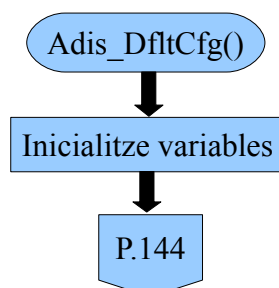


Figura 130: Diagrama de flux simplificat de *Adis\_Auto\_Test()*

A la vista de la figura 130 queda clar que les comprovacions que es realitzen sobre la IMU són molt bàsiques i no arriben a aportar excessiva informació sobre el resultat del test, però com la IMU sempre ha funcionat correctament, no hem tingut la necessitat d'elaborar més aquesta funció.

### 9.5.2- Funció *Adis\_DfltCfg()*

Aquesta funció podria ser considerada una continuació de l'anterior ja que s'encarrega de comprovar els registres de lectura/escriptura del sensor inercial i de modificar-los en cas de no coincidir.



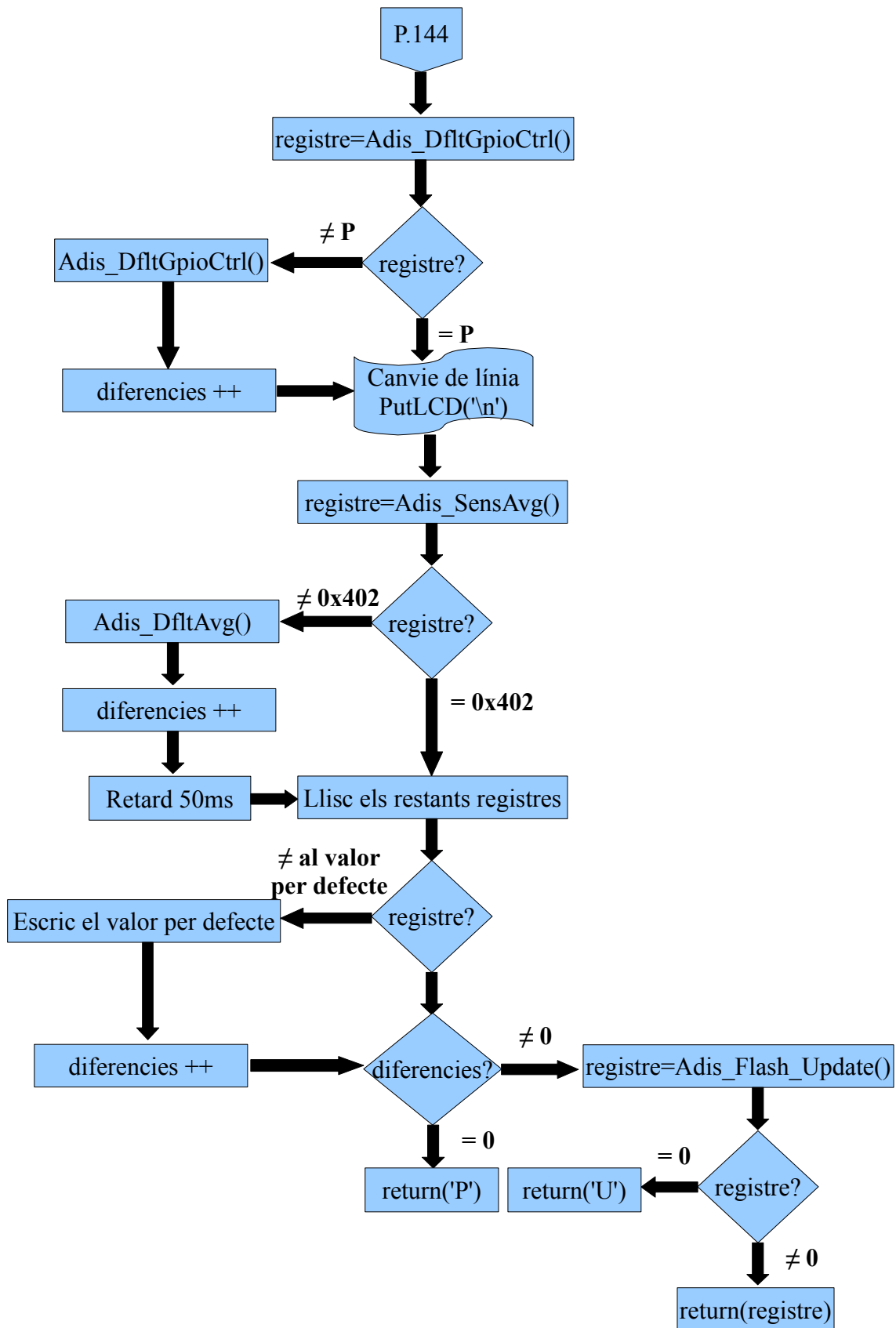


Figura 131: Diagrama de flux simplificat de *Adis\_DfltCfg()*

Tot i que el diagrama de flux d'aquesta funció podria fer-nos pensar que executem diferents funcions per a conèixer el contingut dels registres i canviar el seu valor, en realitat el que estem fent és utilitzar macros basades en la funció *Read\_Adis()* i *Write\_Adis()* per a realitzar estes operacions.



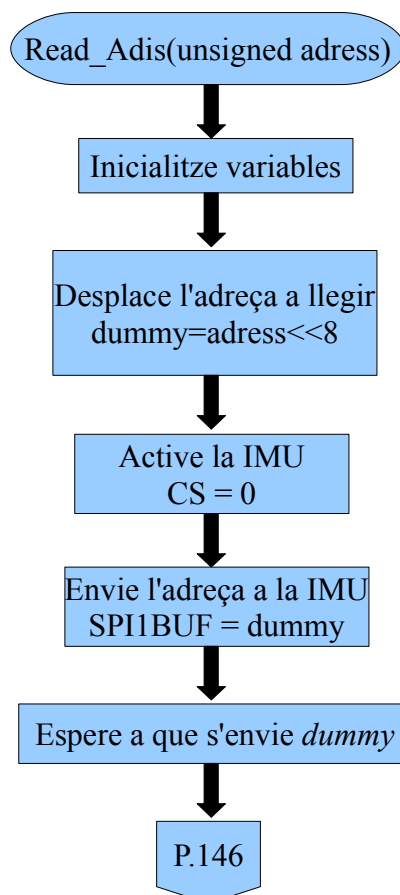
Cal dir que mentre depuràvem aquesta funció vam arribar a pensar que el sensor havia eixit defectuós de fabrica, pues al llegir alguns registres no obteníem el valor esperat i, en el cas del registre GPIO\_CTRL, inclús no canviava de valor després de ser escrit. A l'observar estes anomalies tornarem a contactar amb *Mark Looney* i, parlant amb ell, ens va aclarir que, en el cas de GPIO\_CTRL, la part alta del registre reflectia l'estat dels terminals DIOx quan estos estaven configurats com a entrades, mentre que el fet d'obtenir valors diferents en els registres d'*off-set* podria ser perfectament un efecte secundari d'haver configurat incorrectament la comunicació SPI amb la IMU, cosa més que possible ja que, en eixa situació, l'ADIS16400 podria haver interpretat qualsevol cosa quan intentàvem accedir a ell.

### 9.5.3- Funcions de lectura

La IMU ADIS16400 pot ser llegida de dues maneres: mitjançant la lectura simple d'un registre o bé en mode ràfega. La principal diferència d'aquest mètodes, sempre que no ens fixem en el procediment de lectura, és que la lectura simple ens permet accedir a qualsevol registre mentre que la lectura en mode ràfega sols ens permet accedir als registres d'eixida dels sensors. Esta peculiaritat fa que, per norma general, utilitzem el mode ràfega per a llegir els sensors i utilitzem la lectura simple per a llegir l'estat intern del sensor.

#### 9.5.3.1- Funció *Read\_Adis(unsigned char adress)*

La funció *Read\_Adis()* és l'encarregada de realitzar una lectura simple de la IMU. Per norma general l'utilitzem per a llegir els registres de configuració del dispositiu, encara que no directament sinó que baix la aparença d'una macro per a, d'eixa manera, no tenir que recordar a cada vegada l'adreça del registre al que volem accedir.



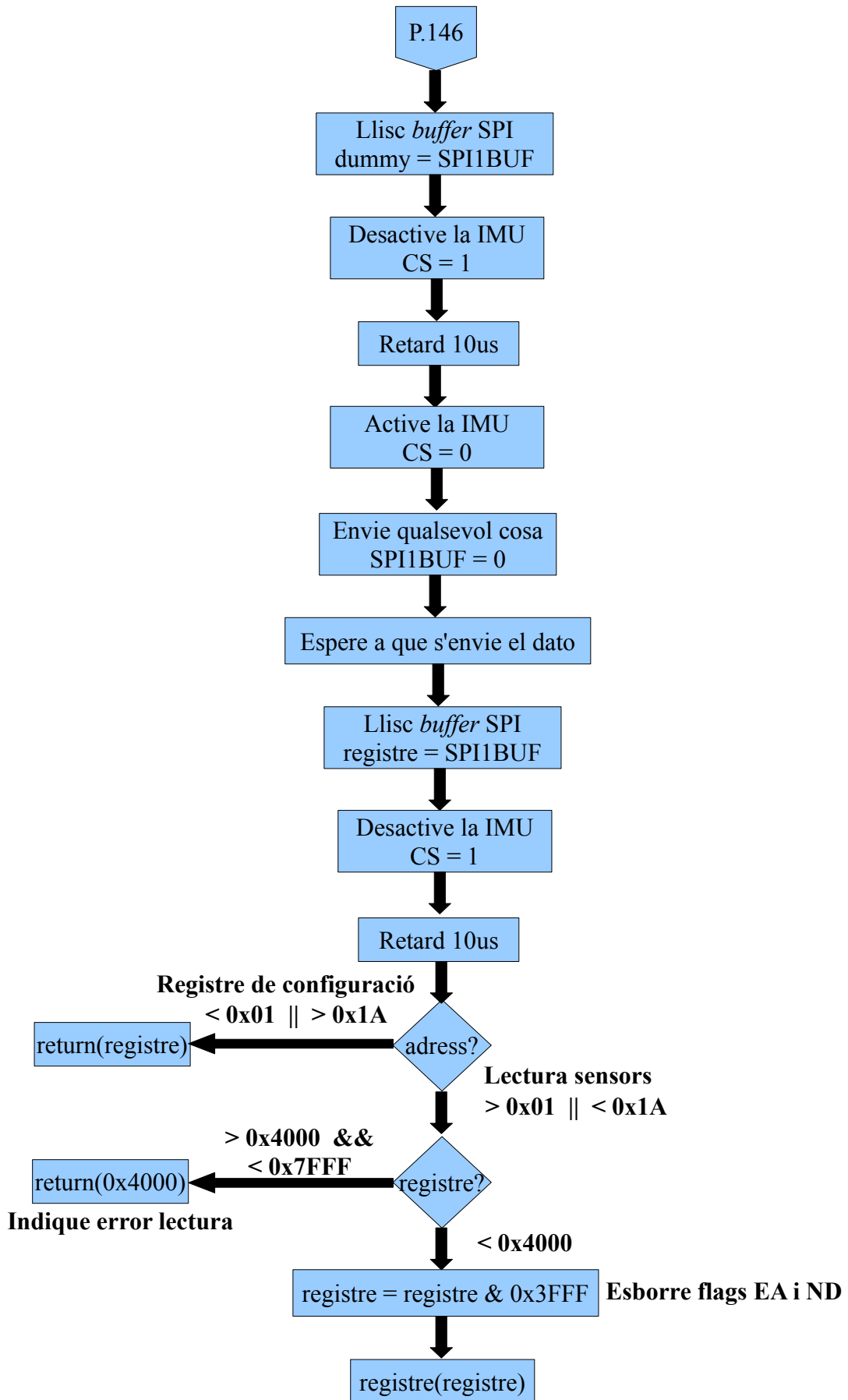


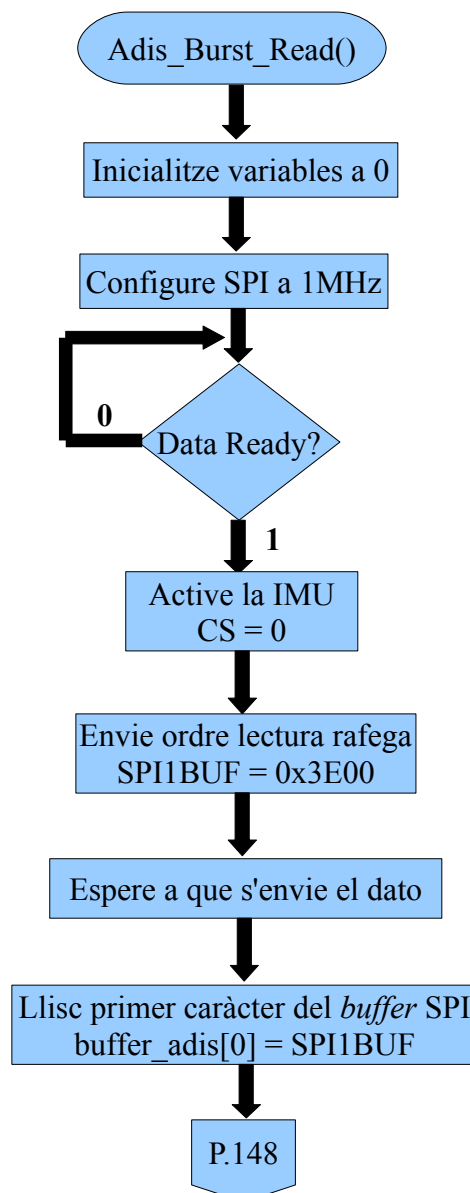
Figura 132: Diagrama de flux de *Read\_Adis(unsigned char adress)*

Com per norma general no utilitzem aquesta funció per a llegir l'eixida dels sensors de l'ADIS16400, hem optat per no esperar al polse que ens envia la IMU quan disposa de noves lectures. Això vol dir que, en cas de voler utilitzar aquesta funció per a llegir en dos instants diferents un dels sensors interns, hauríem d'esperar l'activació de DIO1 abans de tornar-lo a llegir.

Una altra cosa a tenir en compte és el temps d'espera entre lectura i lectura. Sí mirem la fulla de característiques de l'ADIS16400 veurem que diu que aquest temps ha de ser d'uns 9 micro segons, però la veritat és que eixe temps ha de ser lleugerament major quan es treballa a la màxima velocitat de rellotge que permet aquest sensor inercial, doncs a una  $F_{SCLK}$  de 2MHz i amb un retard de 9 us entre accés i accés, es produeixen esporàdicament errors de lectura, per això entre lectura i lectura esperem 10 micro segons.

### 9.5.3.2- Funció *Adis\_Burst\_Read()*

Esta funció ens permet llegir de colp tots els registres de dades del sensor. Degut a que no hi ha que esperar entre lectura de registre i lectura de registre, este mètode ens permet llegir d'una manera més eficient i ràpida l'ADIS16400. A més, com totes les mostres s'han obtingut en el mateix instant es garanteix la coherència de les mateixes.



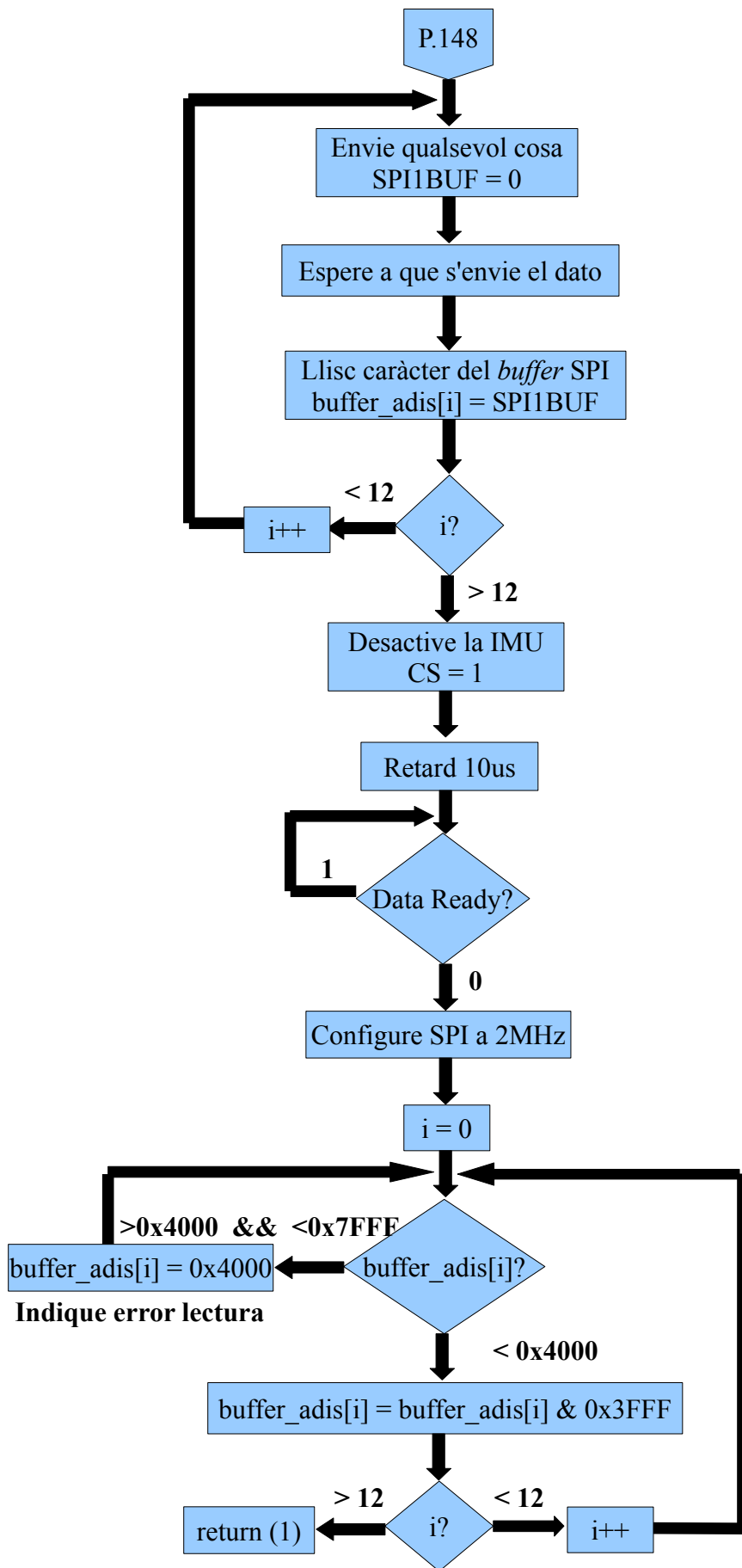


Figura 133: Diagrama de flux de la funció *Adis\_Burst\_Read()*

### 9.5.4- Funcions d'escriptura

En principi, per a escriure en els registres de l'ADIS16400, hi ha prou amb una única funció, però per comoditat, nosaltres n'utilitzem dos: una per a escriure 8 bits i una altra per a escriure'n 16, és a dir, per a escriure tot el registre de colp.

#### 9.5.4.1- Funció *Write\_Adis(unsigned char adress, int data)*

La funció *Write\_Adis()* ens permet escriure 8 bits en la part alta o en la baixa de qualsevol registre del sensor d'inèrcia que no siga de sols lectura.

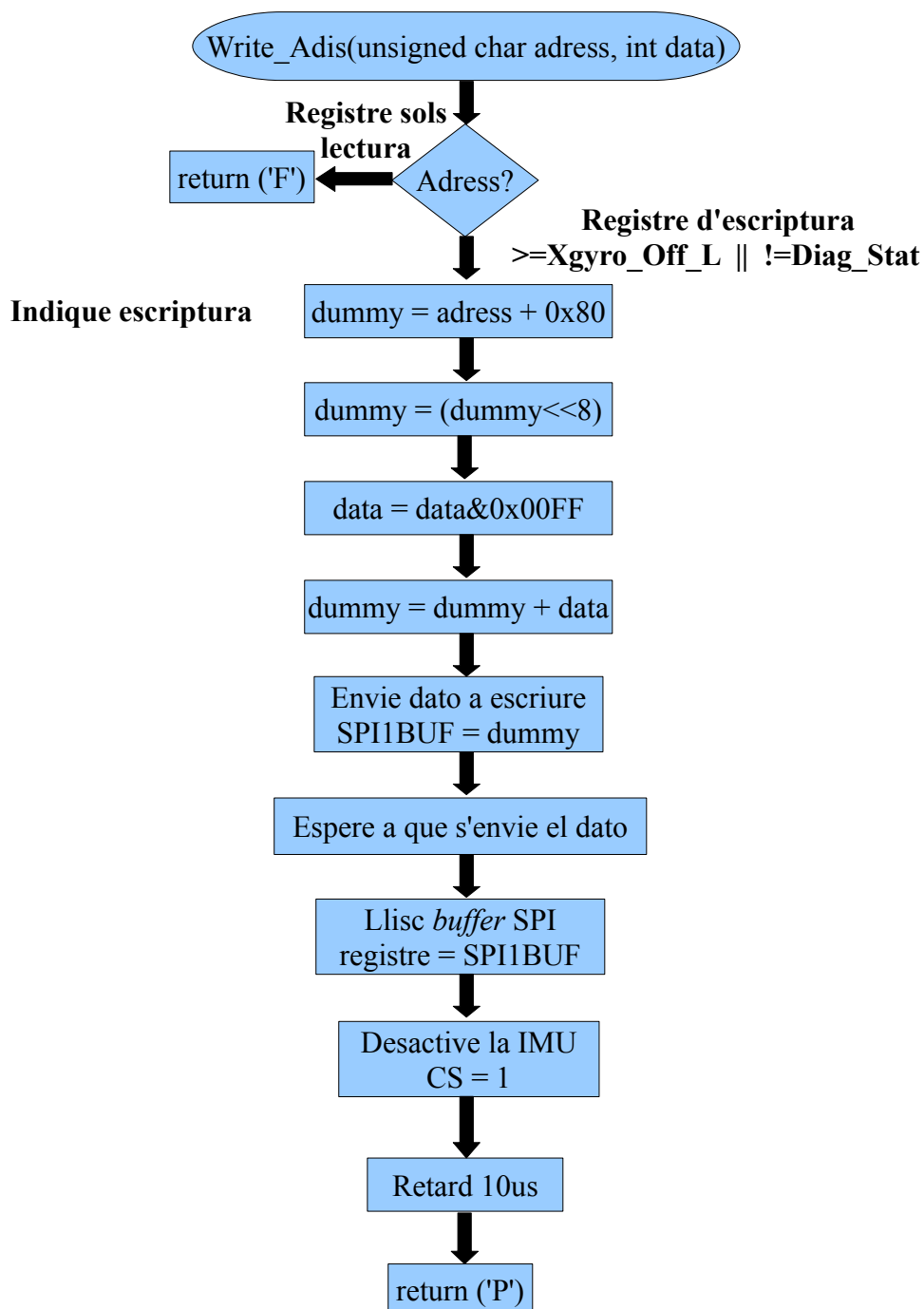


Figura 134: Diagrama de flux de *Write\_Adis(unsigned char adress, int data)*

Possiblement pareixerà absurd utilitzar una variable de 16 bits per a guardar el valor de 8 bits que volem escriure en la IMU, però en realitat no ho és perquè operar matemàticament amb variables de diferent longitud de paraula pot provocar un comportament inesperat del sistema. De fet, és el que ens va passar a nosaltres quan estàvem provant la funció *Write\_Adis\_16bit()*, pues al passar el valor de 8 bits a escriure a la funció *Write\_Adis()*, l'escriptura no és realitzava correctament perquè, just abans de sumar-la al valor amb el valor que es carregaria el registre SPI1BUF, havíem realitzat operacions lògiques amb variables de diferent longitud. Sí no arriba a ser per un comentari de Toni, el tècnic de laboratori, no hauríem caigut en el detall perquè, sorprenentment, el compilador C18 sí produïa un missatge d'avertència al realitzar una operació d'aquest tipus, però el C30 no.

#### 9.5.4.2- Funció Write\_Adis\_16bit (unsigned char adress, int data)

Gràcies a esta funció podem escriure els 16 bits que ocupa un registre del sensor d'inèrcia amb una sola invocació i sense preocupar-nos de res més que facilitar l'adreça i el valor a escriure.

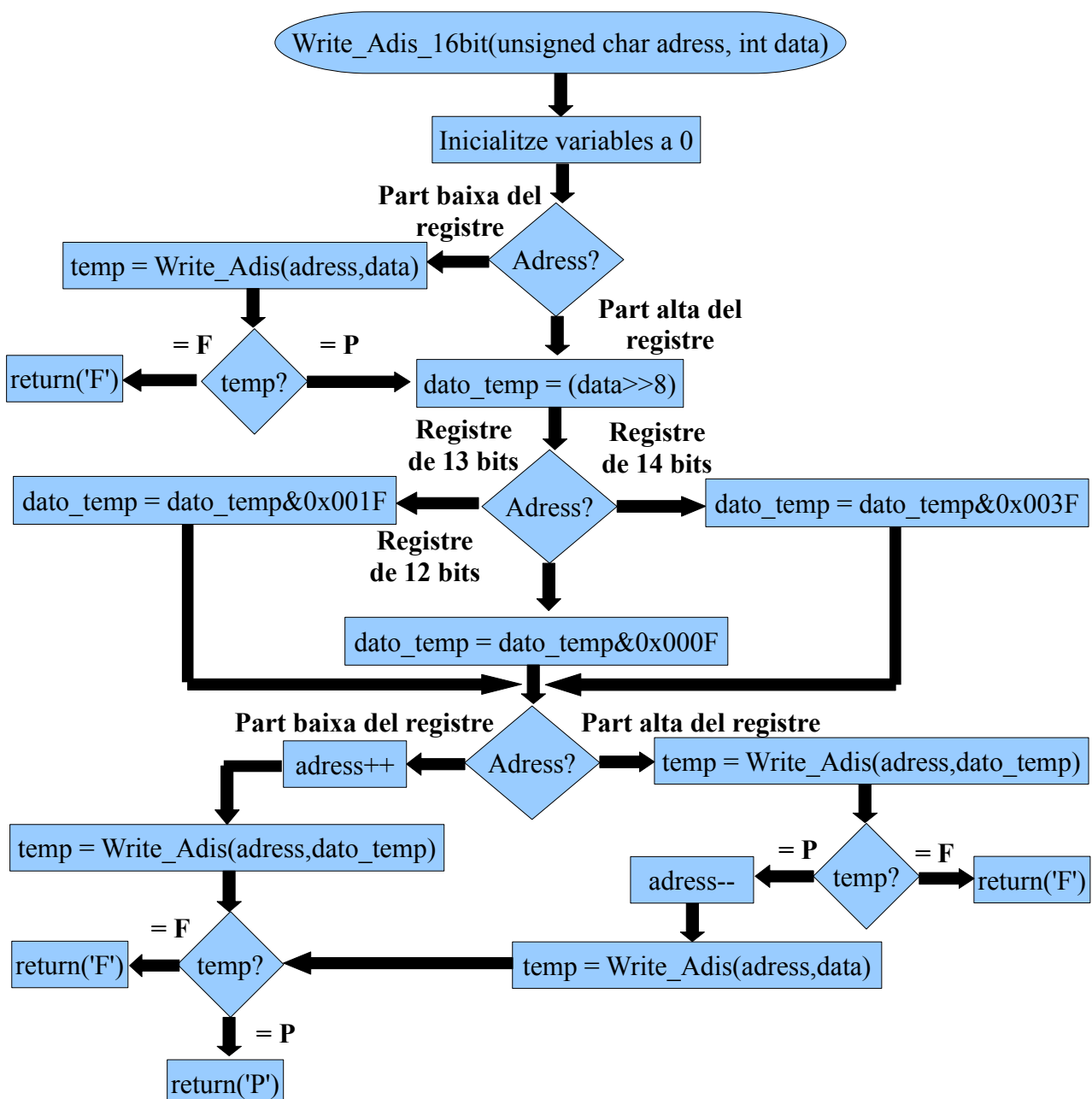
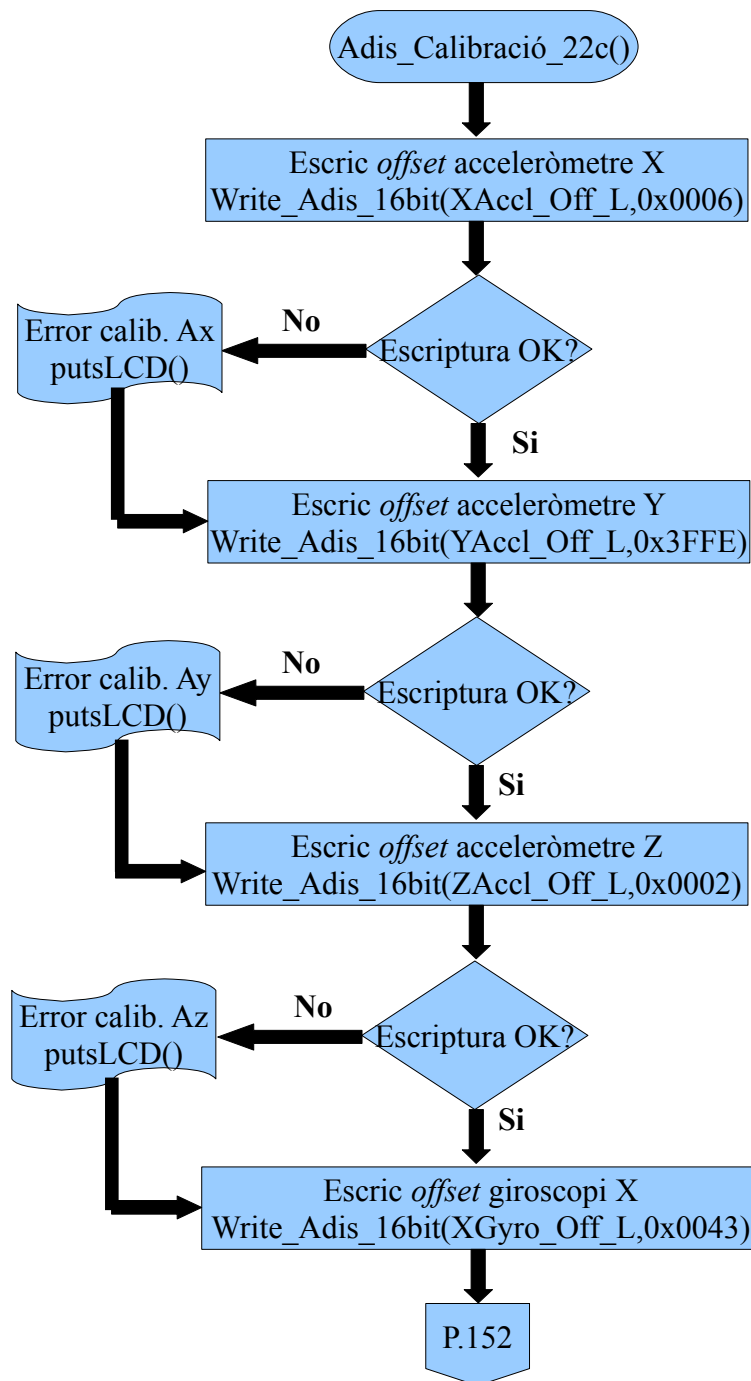


Figura 135: Diagrama de flux de *Write\_Adis\_16bit(unsigned char adress, int data)*

Com ja veurem en les seccions posteriors, aquesta funció ens vindrà molt bé per a escriure els valors obtinguts durant el calibratge dels sensors, doncs no haurem de preocupar-nos per separar el valor resultant en dos parts ni fixar-nos si anem a escriure la part alta o la part baixa del registre d'*off-set* corresponent.

### 9.5.5- Funció *Adis\_Calibracio\_22c()*

Aprofitant que els valors del procés de calibratge solen canviar molt poc entre una vegada i una altra vegada, vam decidir escriure aquesta funció per a no tenir que calibrar el sistema en cada inici del mateix.



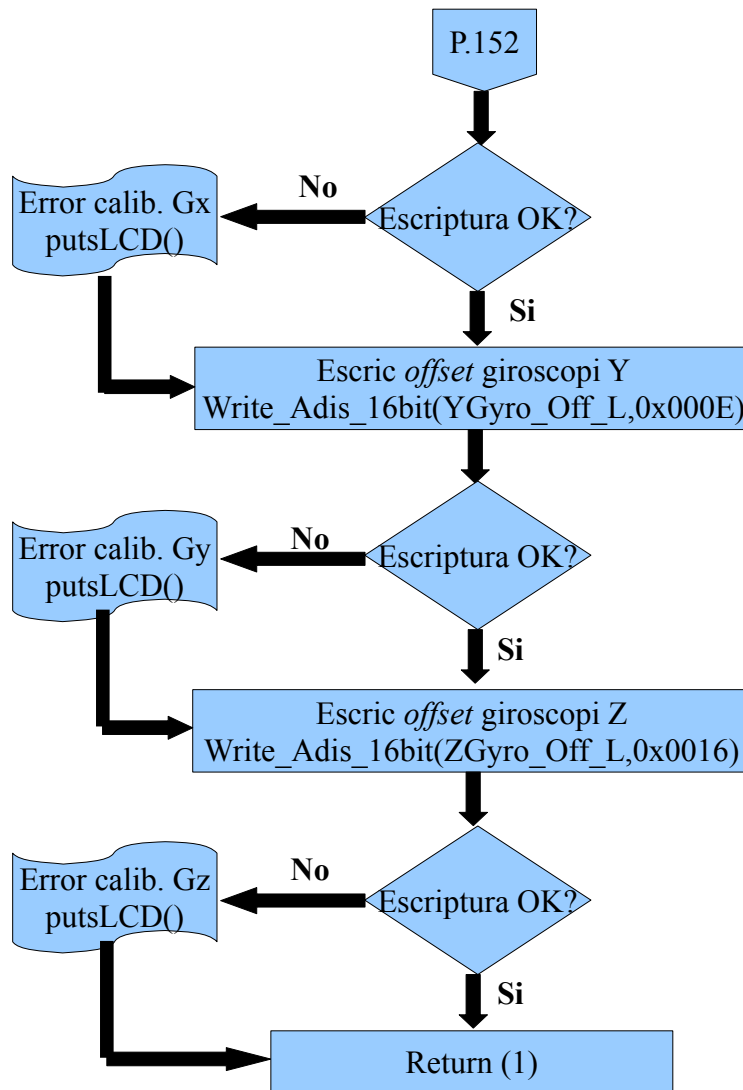
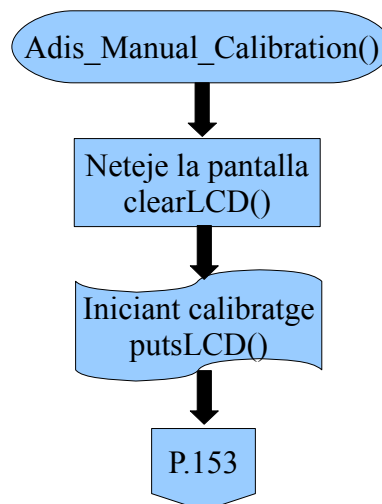


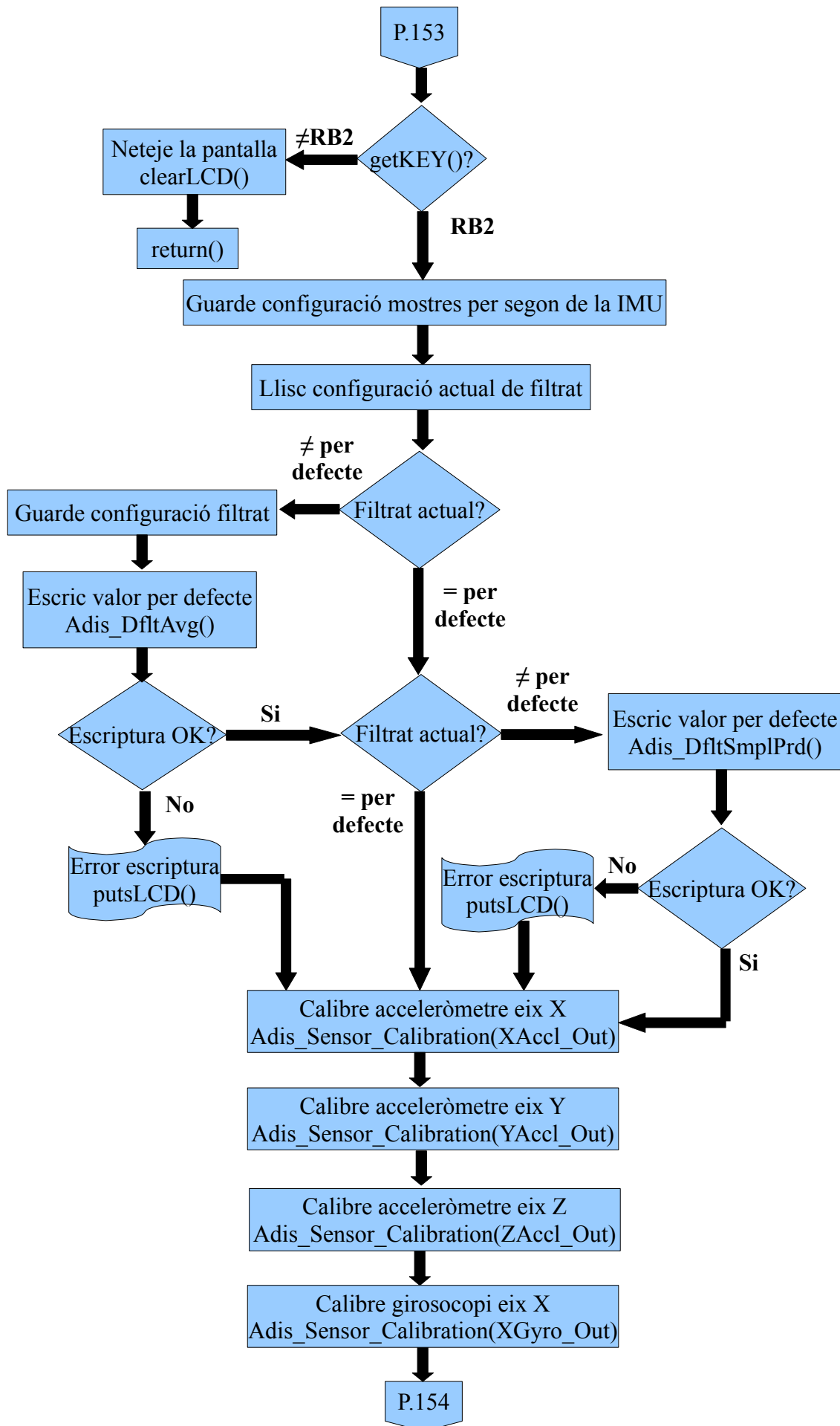
Figura 136: Diagrama de flux de *Adis\_calibracio\_22c()*

### 9.5.6- Funció *Adis\_Manual\_Calibration()*

Esta funció, al contrari que pot fer pensar el seu nom, s'encarrega de preparar i iniciar el procés de calibratge dels acceleròmetres i giroscopis de l'ADIS16400.







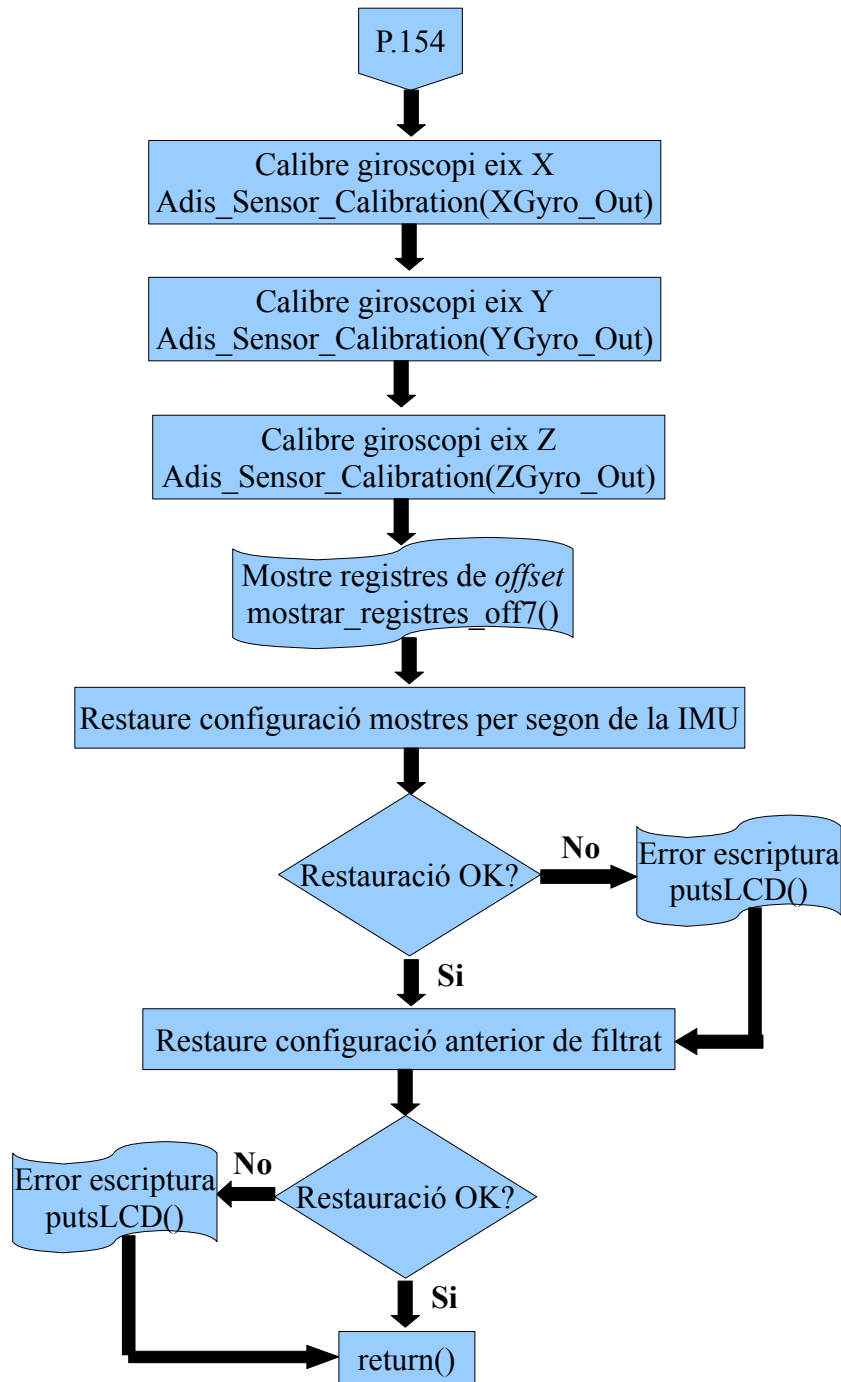


Figura 137: Diagrama de flux de *Adis\_Manual\_Calibration()*

### 9.5.7- Funció *Adis\_Sensor\_Calibration(unsigned char sensor)*

La funció *Adis\_Sensor\_Calibration()* permet a l'usuari realitzar el procés de calibratge del sensor especificat, encara que per a que el resultat siga l'òptim la IMU deurà d'estar prèviament configurada per a prendre 819 mostres per segon amb 2 etapes de filtrat ja que aquesta funció no canvia ni comprova en cap moment la configuració del sensor inercial.

Degut als problemes d'estabilitat que vam experimentar al utilitzar la funció *malloc()* i a que amb aquesta rutina podem calibrar tant els acceleròmetres com els giroscopis, el diagrama de flux resultant és, tal i com podem apreciar en la figura 138, un tant caòtic i enrevessat.

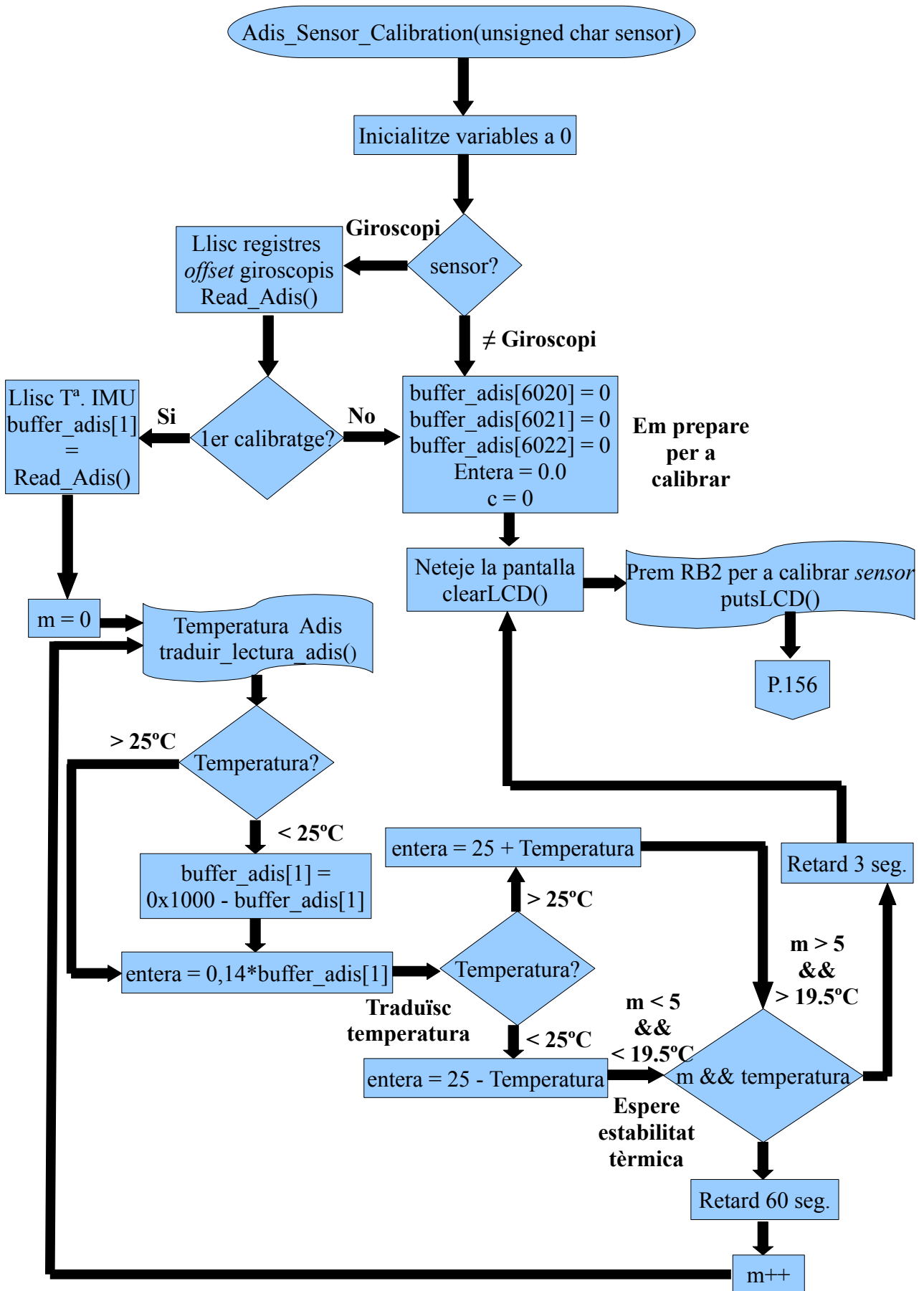


Figura 138a: Diagrama de flux simplificat de la primera etapa de *Adis\_Sensor\_Calibration()*

Com podem veure en la primera part del diagrama de flux de *Adis\_Sensor\_Calibration()*, el primer que fem és comprovar de quin sensor es tracta, ja que si el sensor que volem calibrar és un giroscopi, hem d'esperar a que la temperatura interna de la IMU siga estable per a, d'eixa manera, reduir els efectes de la temperatura sobre l'eixida dels giroscopis durant el procés de calibratge. Cal dir que, tot i que nosaltres esperem a que la IMU estiga a 19.5°C, l'estabilitat tèrmica del sensor no és eixa, sinó que és uns 4-5 graus major, però com no vam notar una diferència excessivament significativa entre calibrar a una o una altra temperatura, establim la temperatura mínima en 19.5°C per a no allargar més el procés de calibratge.

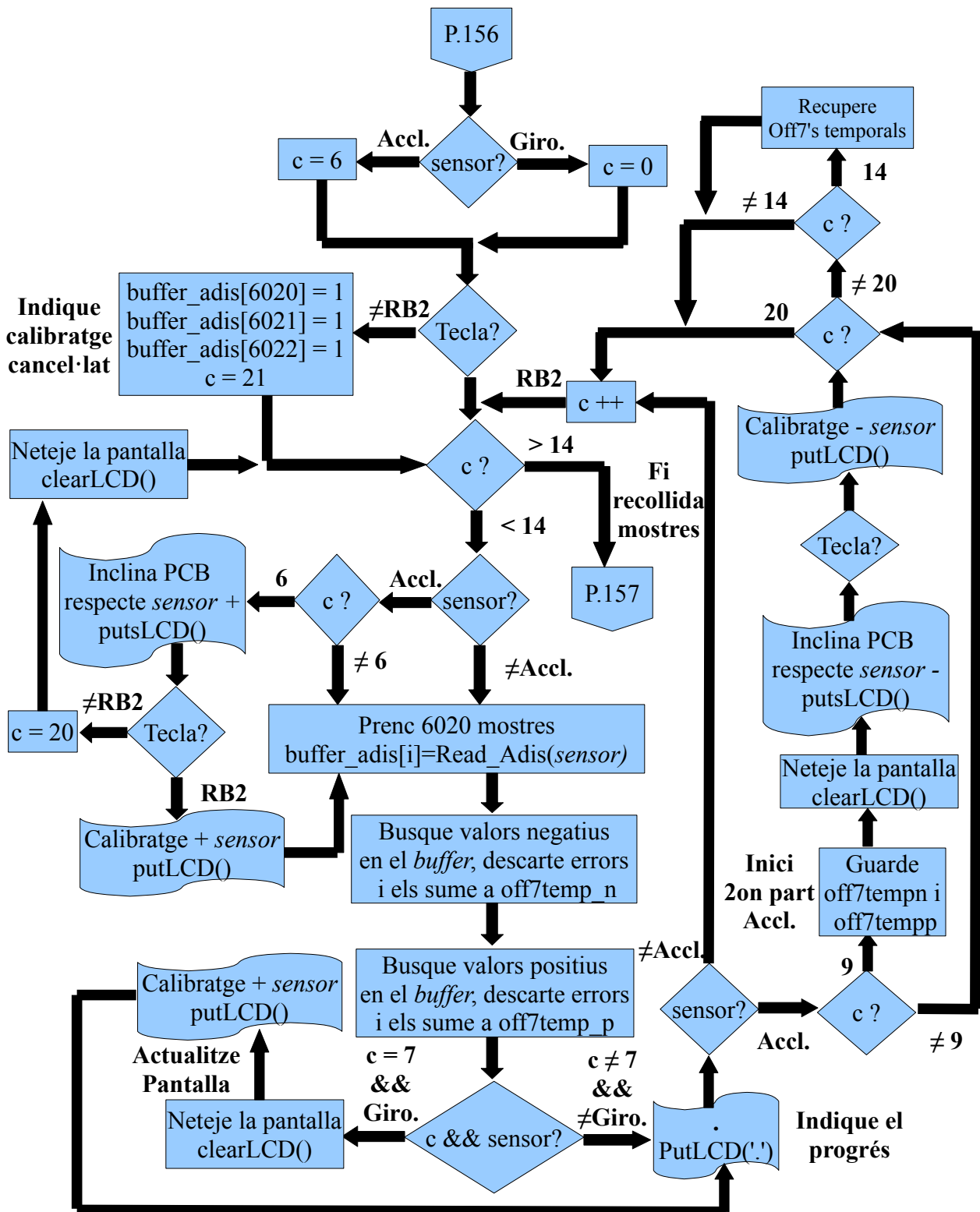


Figura 138b: Diagrama de flux simplificat de la segon etapa de *Adis\_Sensor\_Calibration()*

La part de la funció *Adis\_Sensor\_Calibration()* representada en el diagrama de flux de la figura 138b és l'encarregada de guiar a l'usuari durant el procés de calibratge, adaptar el comportament de la funció al sensor que serà calibrat i finalment obtenir la mitja dels valors capturats. Això vol dir que realitzarà més o menys iteracions del bucle de captura depenent del sensor que està sent calibrat i, que en cas de ser un acceleròmetre, demanarà a l'usuari que incline la placa d'acord amb el sentit de la força que anem a capturar.

Com el sistema no té prou memòria per a capturar la quantitat de mostres exigides pel procés i com quan vam utilitzar *malloc()* el sistema és reiniciava cada dos per tres, vam optar per crear un *buffer* de memòria fixe. El principal problema de fer açò és que eixa memòria ja no es pot alliberar pel que sí volem afegir noves funcions, haurem de modificar amb quasi total seguretat esta funció i qualsevol altra funció que utilitze les últimes posicions del *buffer* com a variables temporals.

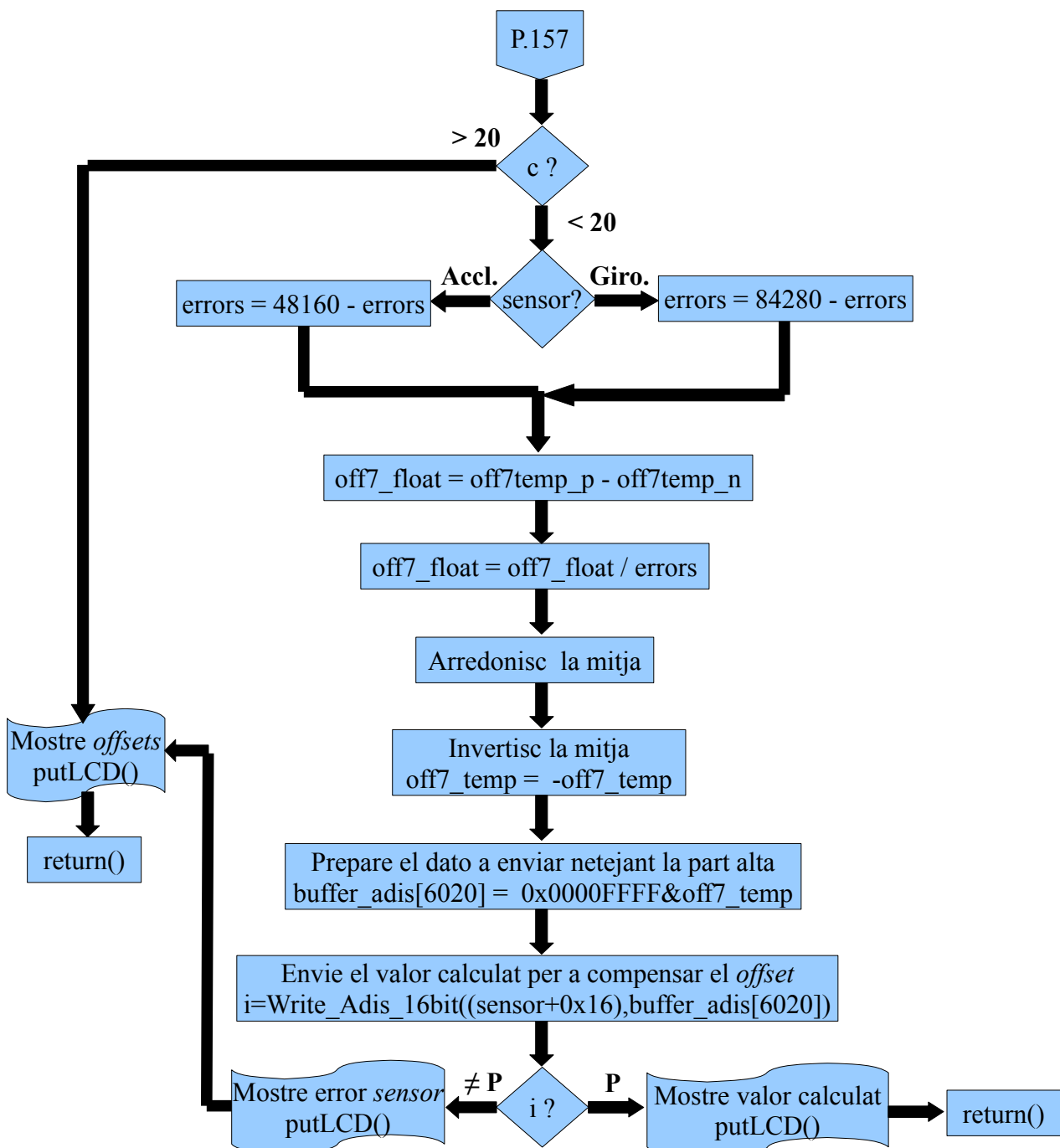
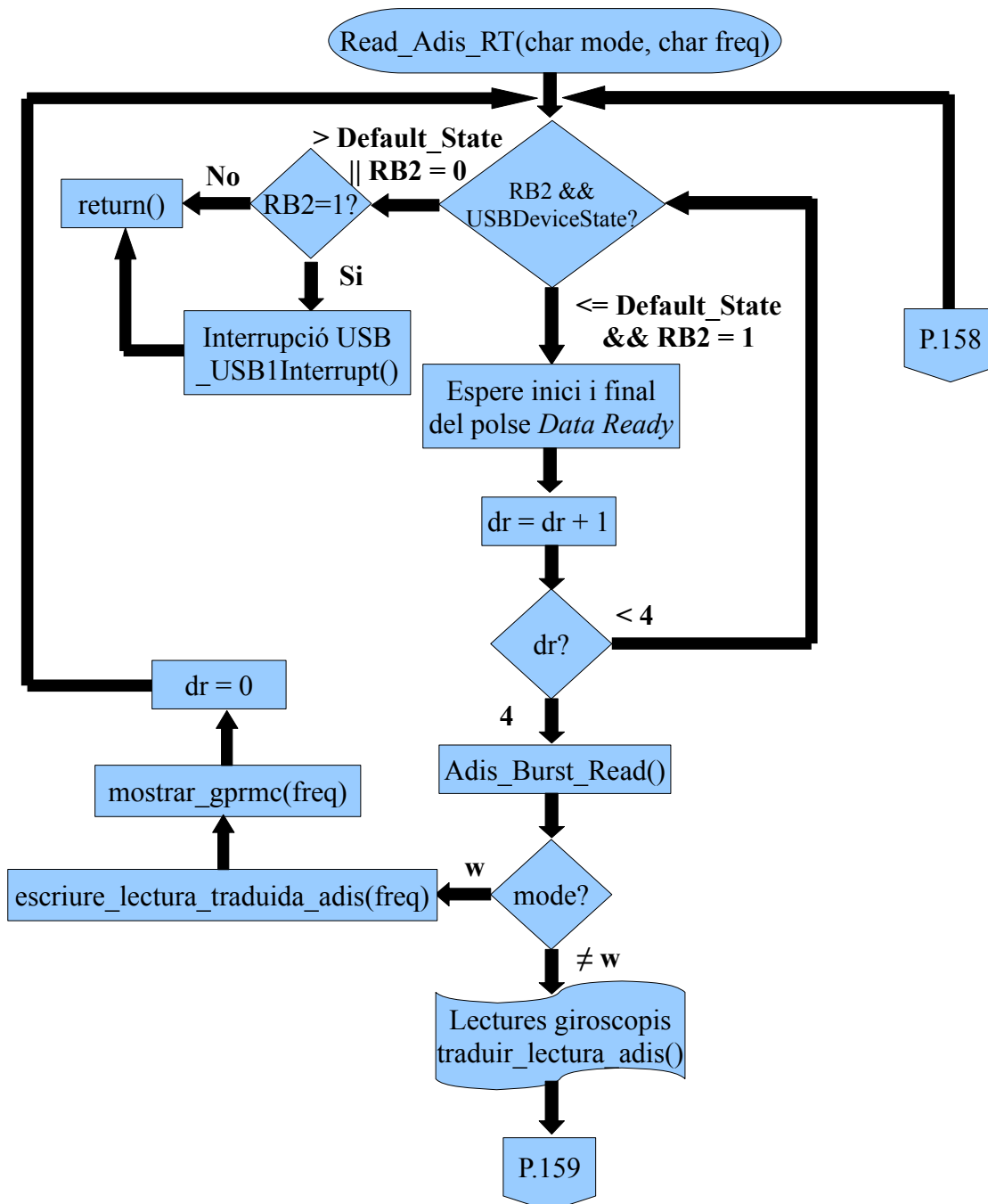


Figura 138c: Diagrama de flux simplificat de la tercera etapa de *Adis\_Sensor\_Calibration()*

Una vegada s'han obtingut la quantitat de mostres necessàries la funció calcula la mitja i, aprofitant que este valor es guarda en una variable *float*, l'arrodoneix per a augmentar la precisió del procés. Com per a poder guardar la suma dels valors no hi ha prou amb variables de tipus *int*, s'ha d'utilitzar una variable de tipus *long* i eliminar la part alta de la mateixa, pues la transformació tal qual d'un tipus de variable a un altre podria donar resultats inesperats.

### 9.5.8- Funció *Read\_Adis\_RT(char mode, char freq)*

Esta rutina llig la IMU en mode ràfega a la velocitat especificada per l'usuari i la mostra per pantalla, a no ser que se també se li indique que les dades capturades han de ser guardades en la targeta micro-SD, situació en la que cedirà el control a *escriure\_lectura\_traduida\_adis()* i en acabar a *mostrar\_gprmc()*.



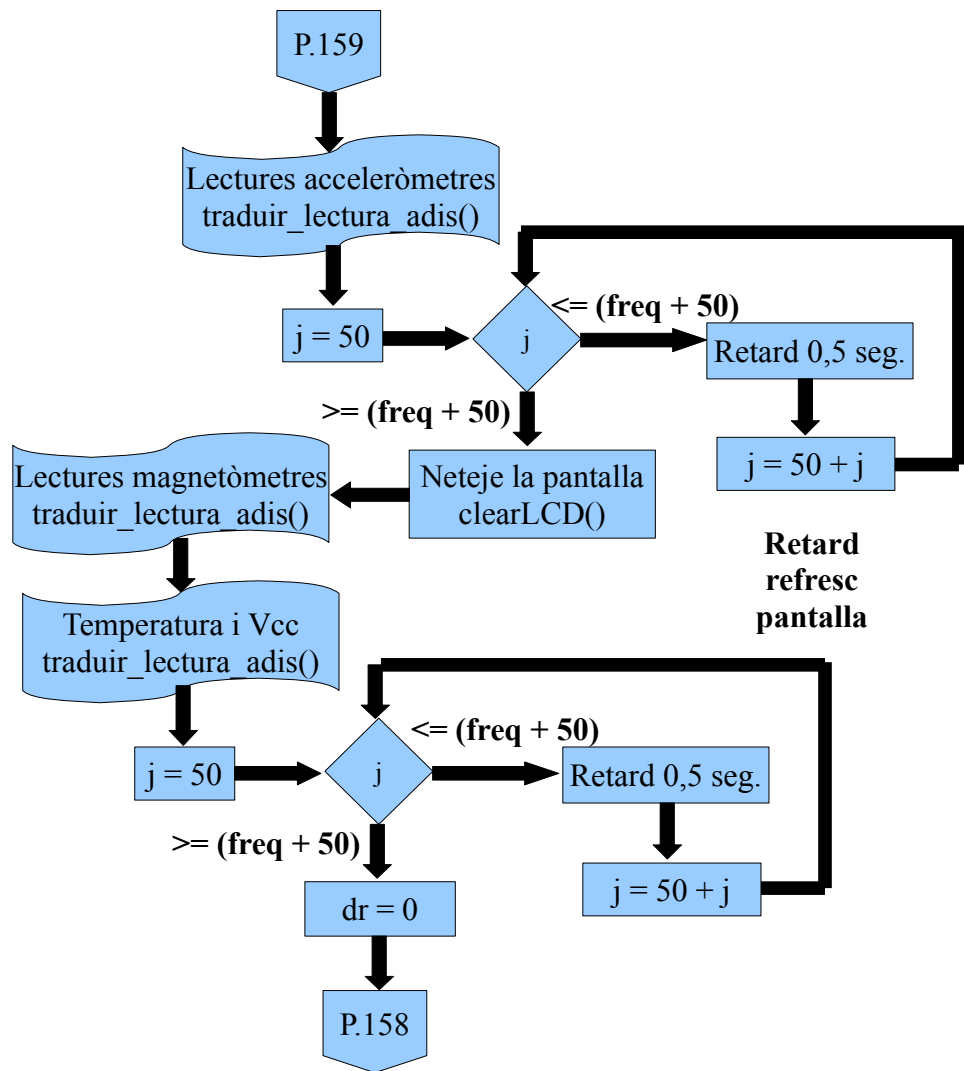
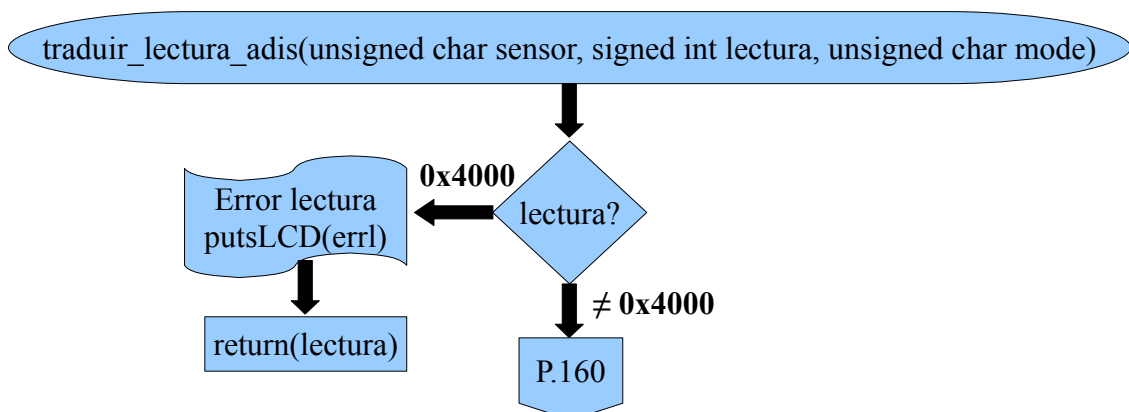


Figura 139: Diagrama de flux simplificat de *Read\_Adis\_RT()*

**9.5.9- Funció traduir\_lectura\_adis(unsigned char sensor, signed int lectura, unsigned char mode)**

Com el nom indica, esta funció tradueix una lectura associada a un determinat sensor i, depenent del paràmetre mode, la mostra per pantalla o no.



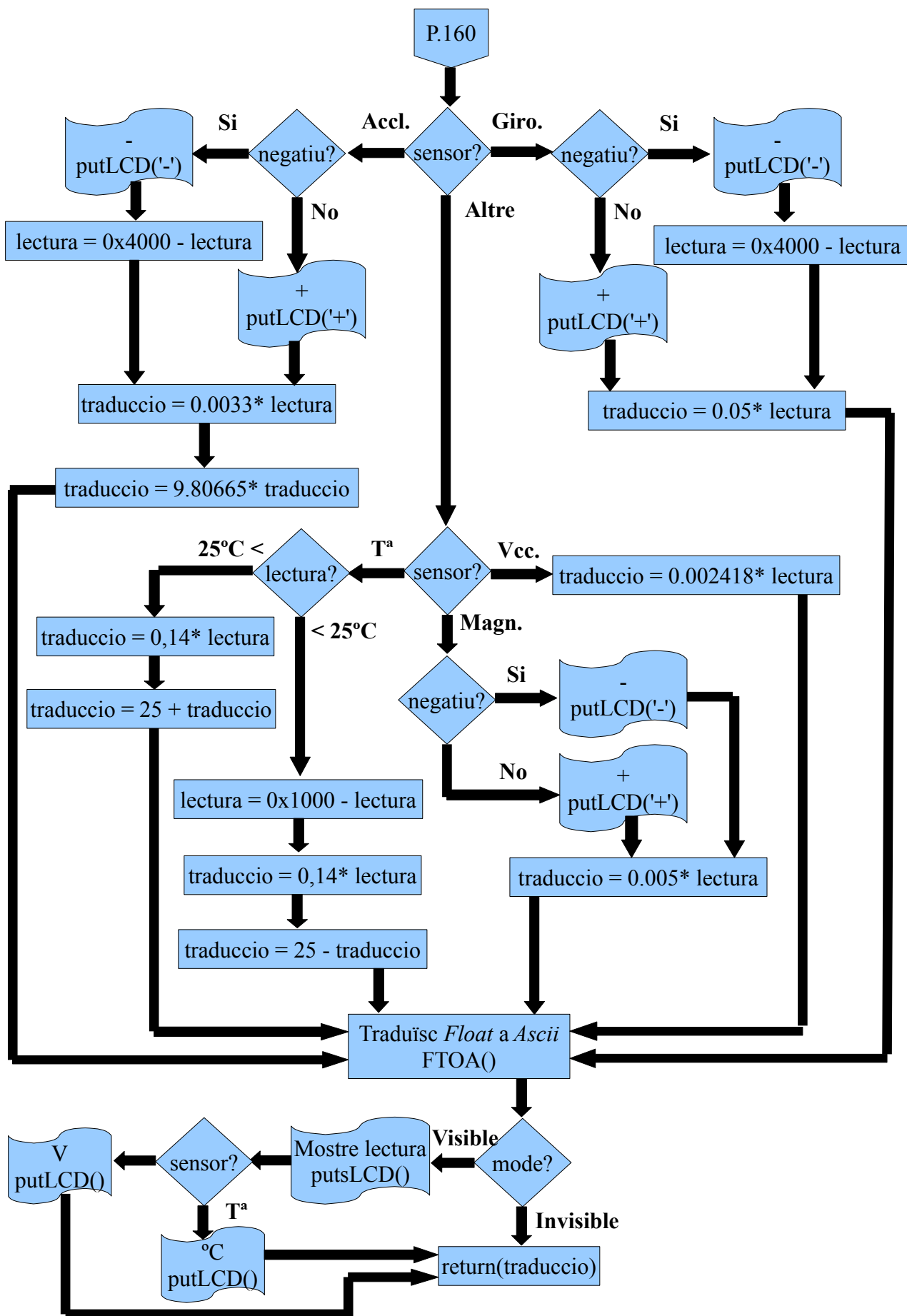
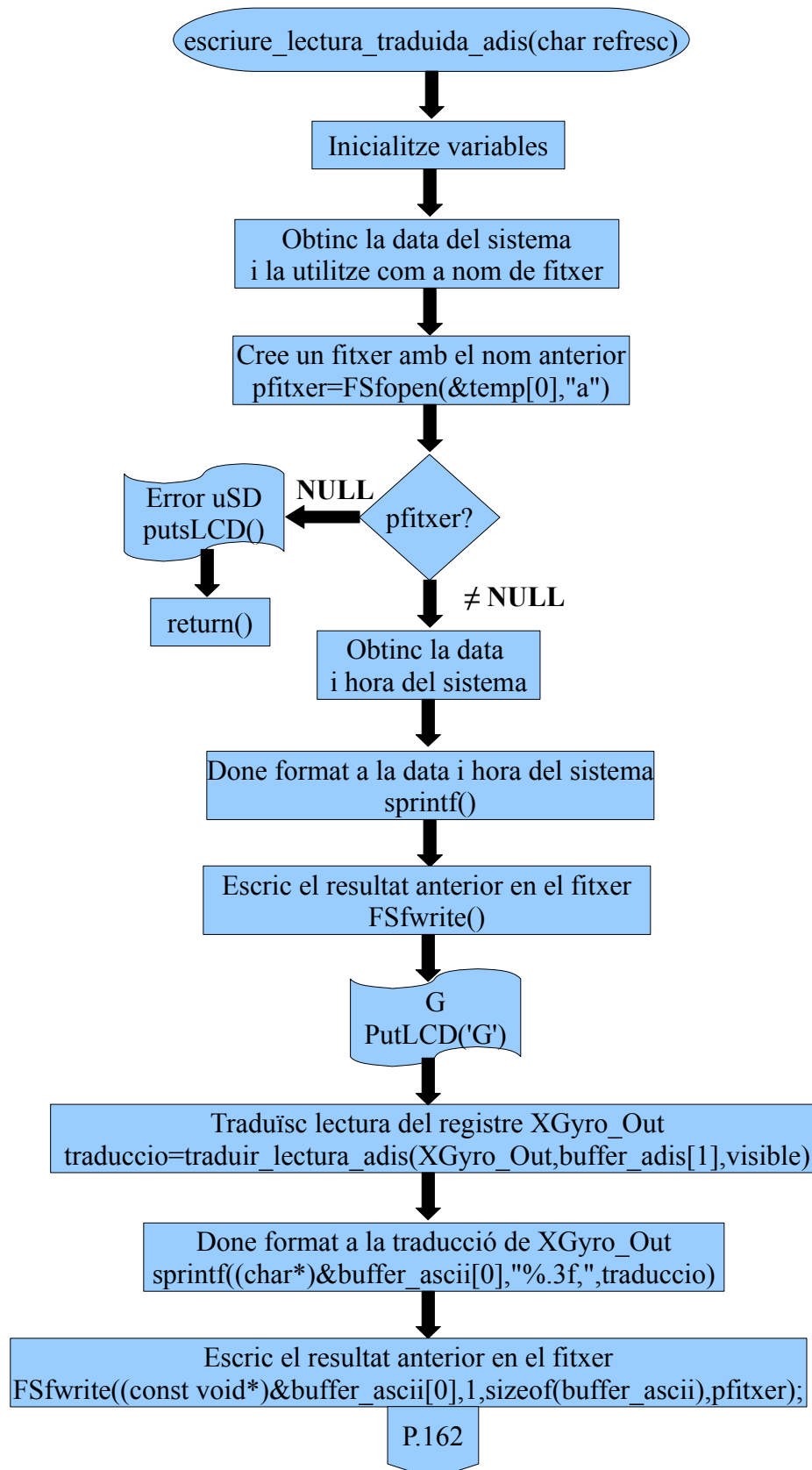


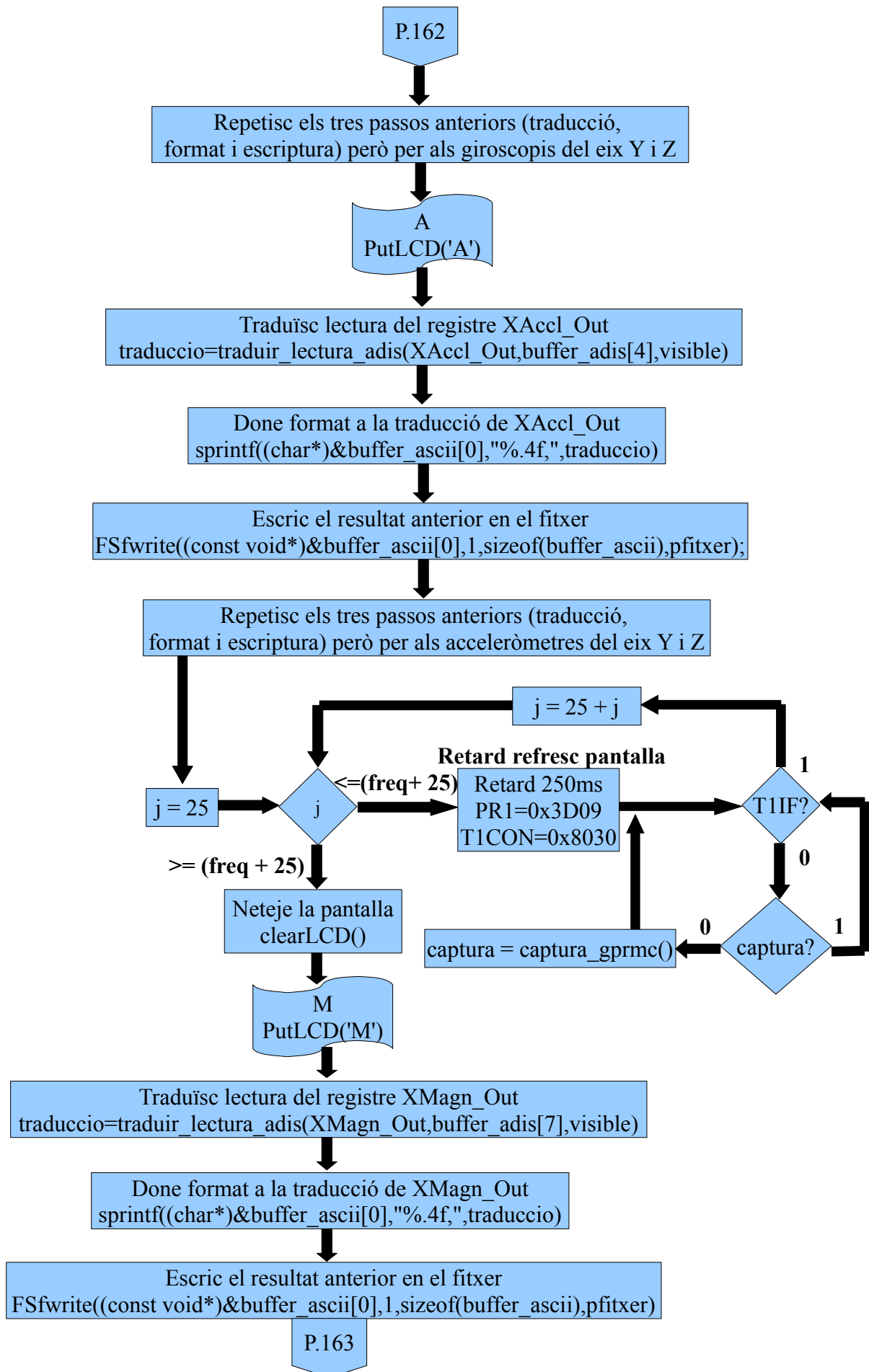
Figura 140: Diagrama de flux simplificat de *traduir\_lectura\_adis()*



### 9.5.10- Funció escriure\_lectura\_traduida\_adis(char refresc)

Tot i que el nom d'aquesta funció sols fa referència a escriure les lectures traduïdes per l'ADIS16400, esta funció no sols mostra estes lectures per pantalla, sinó que també escriu les frases capturades del receptor de GPS.





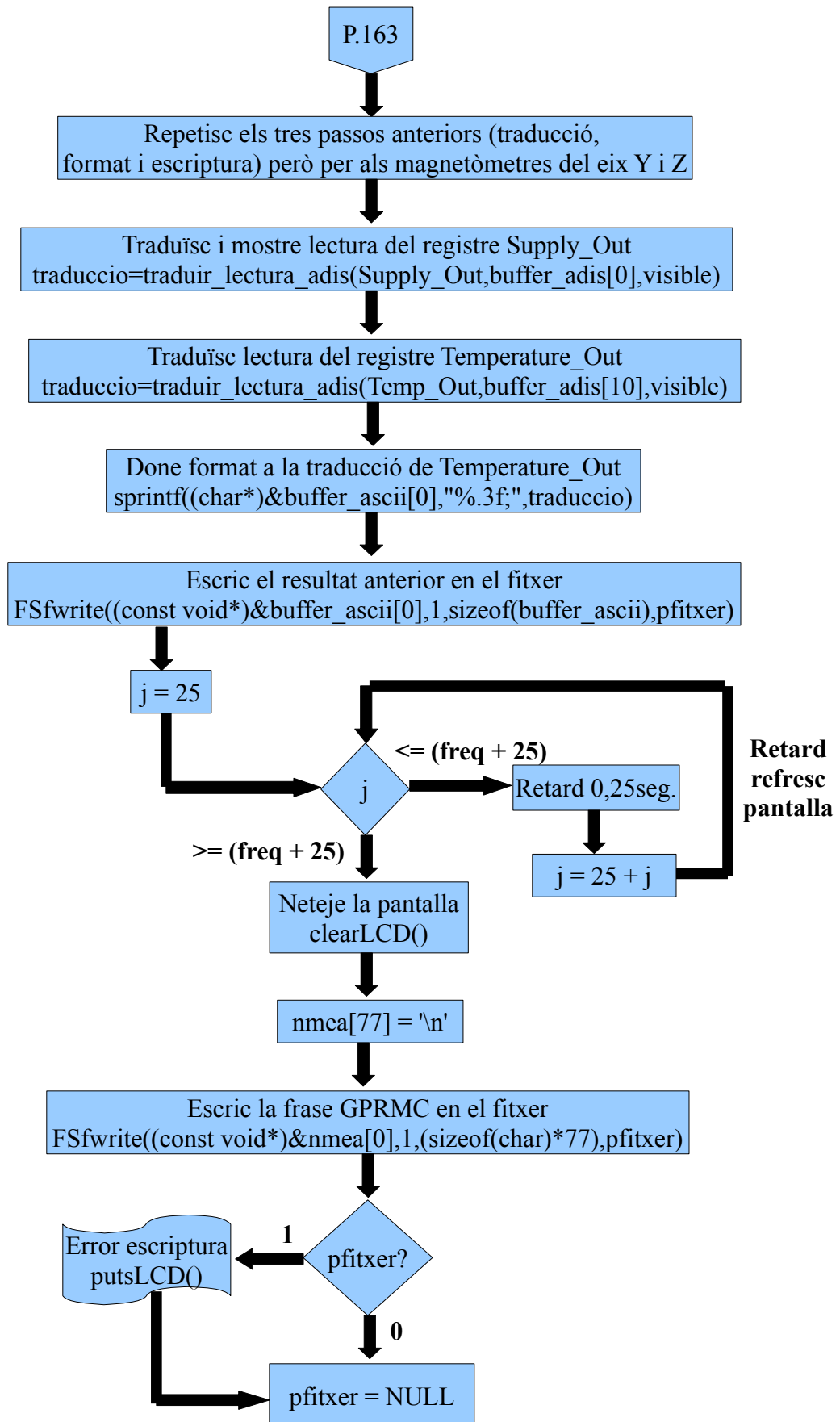


Figura 141: Diagrama de flux simplificat de *escriure\_lectura\_traduida\_adis()*

A l'igual que passava amb la funció *mostrar\_gprmc()*, *escriure\_lectura\_traduida\_adis()* també necessita que, abans de ser cridada, el *buffer* de recepció continga el resultat d'una lectura en mode *burst*, pues de no ser així el resultat de cridar esta funció serà arbitrari.

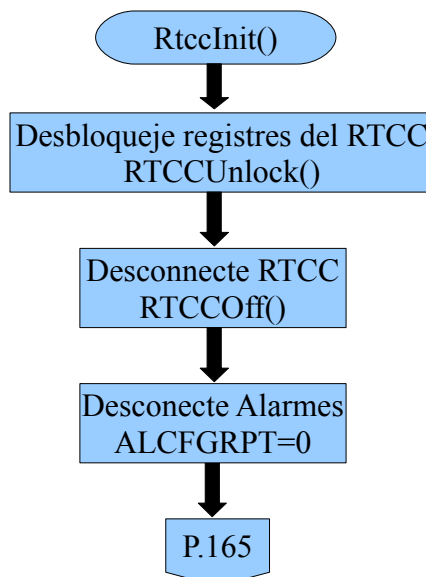
Per últim, cridar l'atenció sobre un detall del diagrama de flux d'aquesta funció. Sí parem atenció al primer retard que fem abans d'esborrar la pantalla i mostrar les lectures dels restants sensors de la IMU, veurem que no ens hem limitat a fer com altres vegades i hem especificat un temps de retard, sinó que hem optat per detallar esta part. Tot i que la principal raó de fer açò no és altra que mostrar el moment en el que aprofitem el temps entre refresc de pantalla i refresc per a omplir el *buffer* de la UART, hi ha una altra raó: detallar el procés de configuració dels *timers* d'aquesta família de PICs. El motiu de fer açò és que aquest procés no s'especifica correctament en la fulla de característiques del PIC, pues en este document, quan s'anomenen els passos bàsics de configuració s'obliden del més important: posar a zero el registre TMRx. Sí no es fa, el temporitzador acaba de contar abans del previst ja que al tenir un valor diferent a zero, necessita contar menys fins arribar al valor especificat per PRx. El pitjor de tot és que els efectes d'aquest error de procediment no tenen perquè ser apreciables a simple vista, raó per la que ens podem considerar afortunats pues els efectes secundaris d'aquesta inicialització incorrecta no es van fer evidents fins que el projecte estava prou avançat, és a dir, fins que no investigarem les possibles raons per les que els calibratges duraven menys del esperat.

## 9.6- Fitxer *rtcc.c*

L'arxiu *rtcc.c* inclou les funcions necessàries per a configurar i utilitzar el mòdul de rellotge de temps real d'un PIC. Algunes d'aquestes funcions estan basades en els exemples que podem trobar en la llibreria d'aplicacions de *Microchip*, pel que per a tenir més detall sobre el funcionament de les mateixes el millor és examinar la documentació proporcionada pel fabricant.

### 9.6.1- Funció *RtccInit()*

La finalitat d'aquesta llibreria és iniciar i configurar el mòdul RTCC per a que pugui ser utilitzat posteriorment. Destacar que sí s'ha configurat la llibreria *MDD File System* per a utilitzar el mòdul del rellotge en temps real, és imprescindible inicialitzar el mòdul RTCC abans que esta llibreria ja que de no fer-ho, la inicialització del sistema d'arxius fallarà.



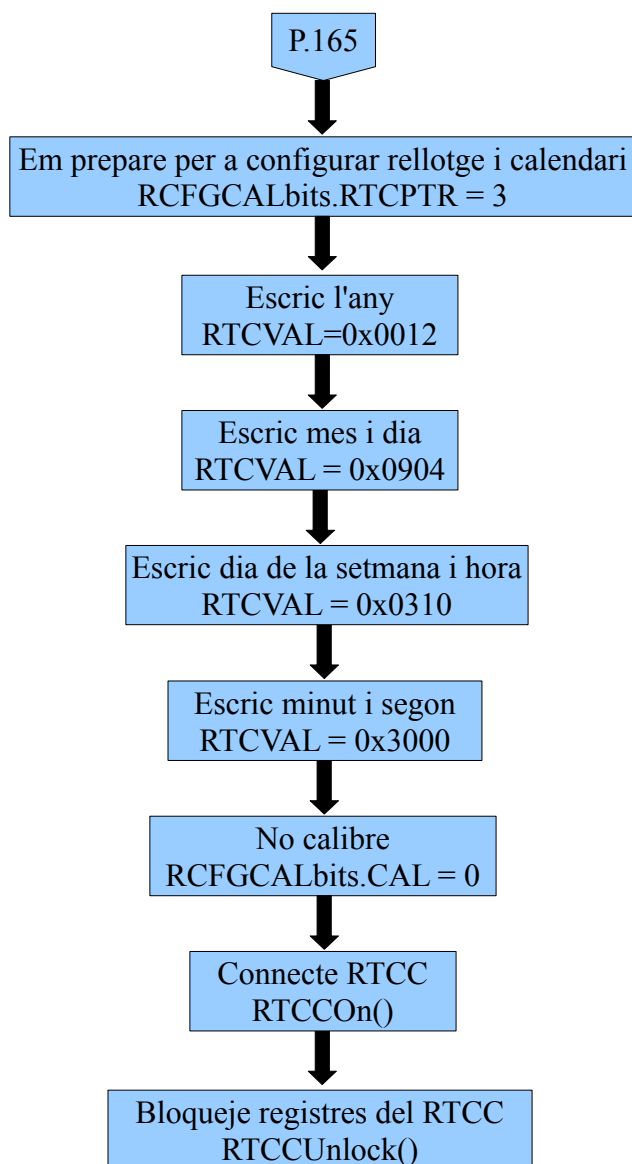
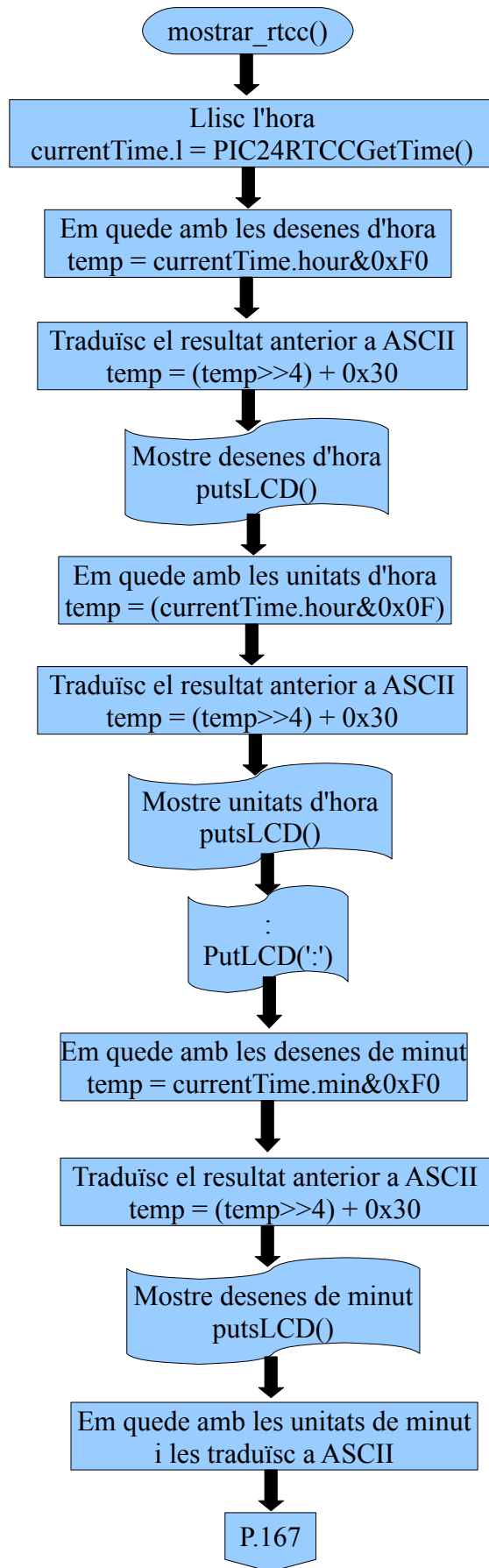


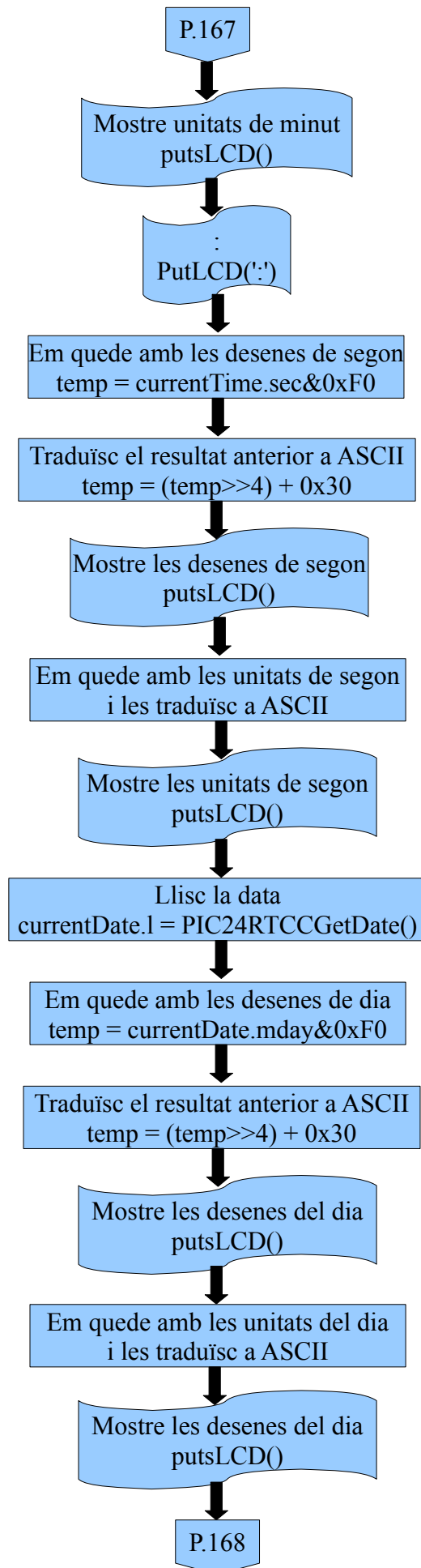
Figura 142: Exemple de configuració del mòdul RTCC utilitzant com a data 04/09/12 i hora 10:30

Sí ens fixem en el diagrama de flux de la figura 142, observarem que aparentment el procés d'inicialització del mòdul RTCC és prou lineal i simple, però la realitat és que és un poc més complicat ja que el mòdul permet calibrar la desviació del oscil·lador secundari utilitzat per a millorar la precisió del mòdul. Com la nostra prioritat era fer funcionar el mòdul, no ens hem preocupat per aquest procés i ens l'hem botat, però de cara a una versió final del sistema seria més que recomanable escriure la part del programa que realitzara aquesta funció. Una altra cosa que també podria resultar interessant estudiar és l'ús alarma del mòdul, pues encara que nosaltres no l'hem utilitzat, més avant podria vindre bé per a programar alguna tasca de manteniment del sistema.

### 9.6.2- ***Funció Mostrar\_rtcc()***

Esta funció ens permet visualitzar la data i hora del sistema en pantalla com sí es tractes d'un rellotge (car), tot i que nosaltres la vam crear inicialment per a saber sí configuràvem correctament el mòdul RTCC, al final l'hem utilitzat per a que l'usuari pugui conèixer l'hora i data actual i, d'esta manera, sàpiga sí l'ha de canviar o no.





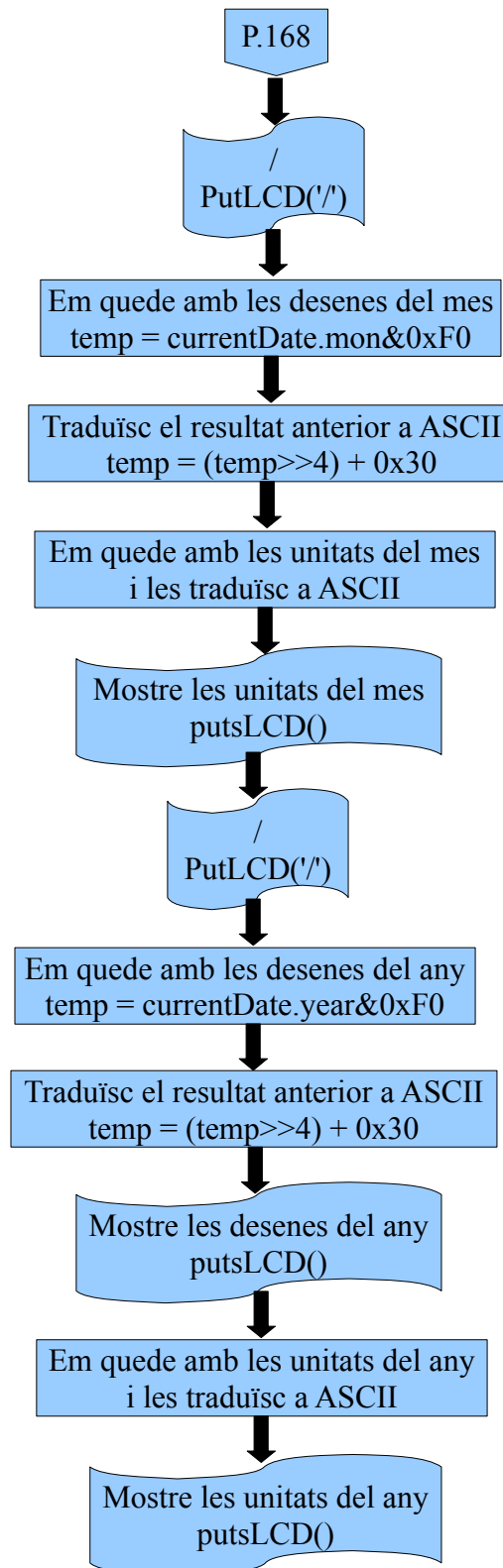


Figura 143: Diagrama de flux simplificat de la funció *mostrar\_rtcc()*

Com podem apreciar en la figura 143, el procediment utilitzat per a mostrar la data i la hora per pantalla és repetitiu, però directe, pues sols cal emprar les funcions *PIC24RTCCGetDate()* i *PIC24RTCCGetTime()* per a que, quan operem amb els *structs currentDate* i *currentTime*, aquests continguin valors coherents que al ser traduïts de BCD a ASCII, ens indiquen l'hora i data actuals.



### 9.6.3- Funcions *PIC24RTCCGetTime()* i *PIC24RTCCGetDate()*

Estes dos funcions formen part de la llibreria d'aplicacions de *Microchip* i permeten obtenir la data i l'hora respectivament. El seu funcionament és quasi calcat i sols es diferencien en el valor que s'escriu al registre RCFGAL: RCFGALbits.RTCPTR0=1 i RCFGALbits.RTCPTR1=1 per a llegir la data, mentre que per a llegir el rellotge s'ha d'escriure RCFGALbits.RTCPTR0=1 i RCFGALbits.RTCPTR1=0.

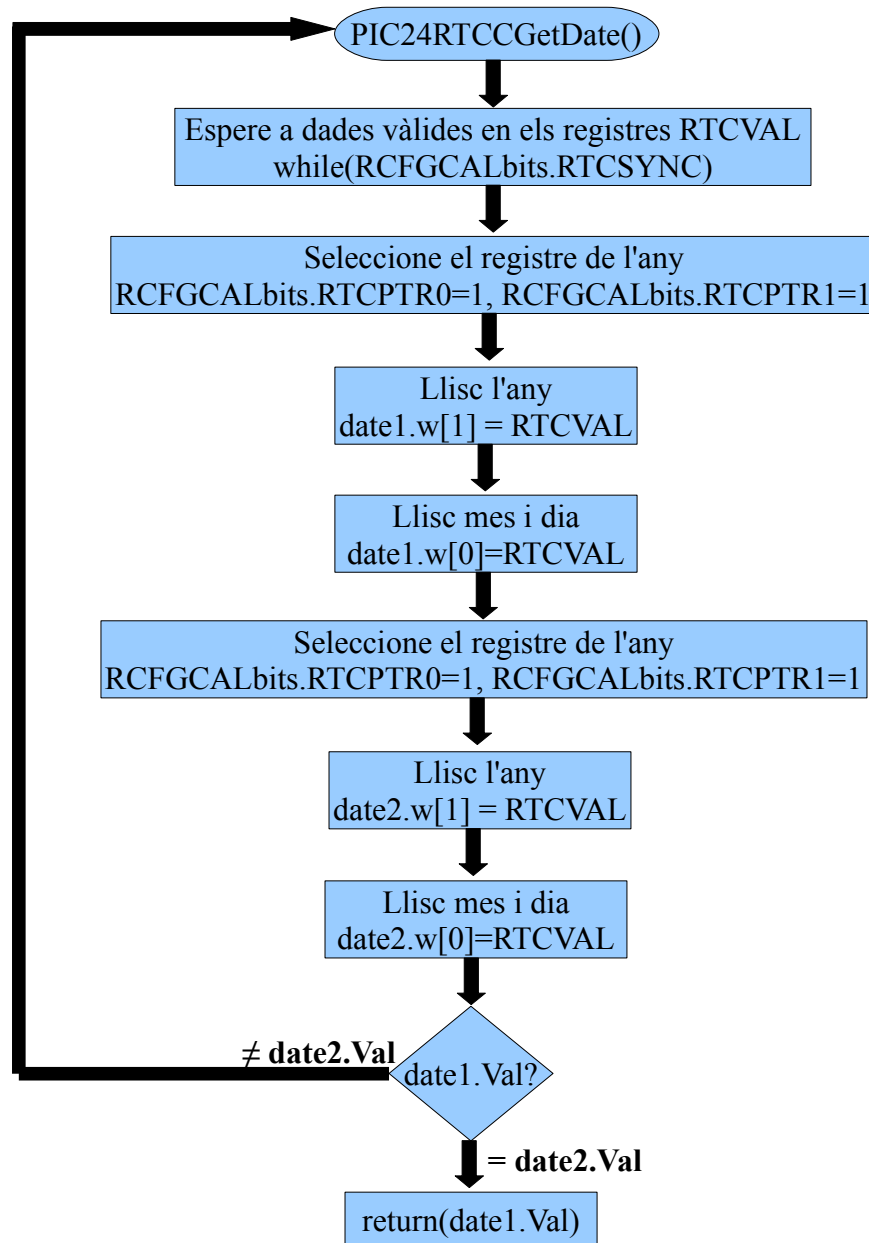


Figura 144: Diagrama de flux de *PIC24RTCCGetDate()*

Sí ens fixem en el diagrama de la figura 144, veurem que dins del procés de lectura, realitzem dues lectures. Açò es fa per a estar segurs que el valor llegit és correcte ja que podria donar-se la coincidència que, just en el moment que llegim el registre RTCVAL, este vaja a ser actualitzat, pel que la lectura ja no seria real.

#### 9.6.4- Funcions PIC24RTCCSetTime (int weekDay\_hours, int minutes\_seconds) i PIC24RTCCSetDate (int xx\_year, int month\_day)

El cas d'estes dos funcions és similar a l'anterior, sols que estes funcions s'utilitzen per a ajustar la data i hora del sistema.

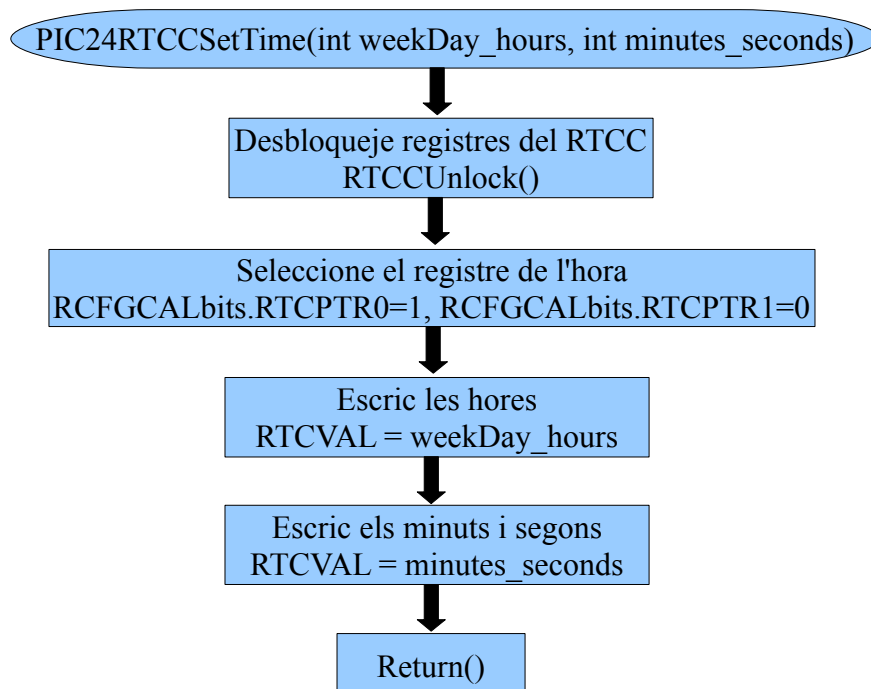


Figura 145: Diagrama de flux de *PIC24RTCCSetTime()*

Aquestes quatre funcions, a pesar que no les anomenen en cap moment en els diagrames de flux de les funcions del fitxer *menus.c*, són les funcions utilitzades per *configurar\_calendari()* i *configurar\_rellotge()* per a ajustar la data i hora del sistema després d'haver adaptat les dades introduïdes per l'usuari al format utilitzat per *SetTime()* i *SetDate()*.

### 9.7- Fitxer funcions\_externes.c

Fins ara hem estat parlant de funcions que hem escrit nosaltres des de zero, però no totes les funcions que hem utilitzat les hem creat nosaltres, sinó que algunes les hem agafat prestades d'Internet o d'algun llibre. Totes estes funcions que no encaixen en cap altre lloc les hem copiat en aquest fitxer. Dins del mateix podem trobar funcions relacionades amb quatre tasques específiques:

- **Gestió sistema de fitxers FAT:** les llibreries de *Microchip* encarregades de gestionar el sistema d'arxius FAT no proporcionen informació sobre el suport físic que conté aquest sistema d'arxius, pel que nosaltres vam copiar un parell de funcions d'un dels exemples d'aplicació de *Microchip* per suplir eixa carència. Teòricament estes funcions deuriem ajudar-nos a conèixer la capacitat de la targeta SD que estem utilitzant i l'espai lliure, però en la pràctica, nosaltres no hem aconseguit fer-les funcionar.
- **Gestió del subsistema USB:** *Microchip* proporciona en la seva llibreria d'aplicacions el material necessari per a poder utilitzar els seus microcontroladors en aplicacions que facen ús d'aquest bus, però estes llibreries, tot i ser totalment funcionals, necessiten una personalització per part de l'usuari ja que, per exemple, *Microchip* no pot saber com vol

que reaccione el microcontrolador en cas que es produïska un error de comunicació USB. Com nosaltres tampoc sabíem exactament com configurar esta part i sols volíem provar que podríem utilitzar el port USB en cas de necessitat, copiarem esta part del codi font de l'exemple d'aplicació *Mass Storage - SD Card data logger* desenvolupat per *Microchip*.

- **Gestió dels pulsadors:** els nostres pulsadors, com és d'esperar, produeixen rebots, pel que, per a evitar els efectes dels mateixos en el sistema, s'han d'eliminar. Podríem haver escrit una funció que s'encarregara d'açò, però aprofitant que en el llibre del que vam obtenir les funcions encarregades de gestionar la pantalla hi havia una funció que feia açò mateixa, decidirem no perdre el temps creant una funció des de zero i utilitzarem esta però amb un petit canvi: un esdeveniment USB pot acabar la funció. De no ser per esta modificació, el nostre sistema no s'enumeraria correctament al connectar-se a un dispositiu *host*, pel que este no podria accedir als recursos del nostre sistema.
- **FTOA():** sense dubte, una de les funcions més útil i importants per al funcionament del sistema. Sense ella no podríem veure per pantalla les traduccions de les lectures dels sensors ni estes podrien ser escrites en fitxer de text, pues esta funció el que fa és traduir una variable *float* a un caràcter *ascii* (FloatTOAscii). Però la importància d'aquesta funció va més enllà d'açò, pues gràcies a la mateixa, hem pogut depurar amb més facilitat el programa ja que la funció *ITOA()*<sup>58</sup> proporcionada amb el compilador C30 no admet variables *long int*, pel que no pot ser utilitzada per a mostrar, per exemple, les mitges calculades durant el calibratge. Com esta funció no forma part de les llibreries subministrades amb el compilador C30, la vam tenir que buscar per Internet. En la cerca vam trobar varies implementacions de la funció en el fòrum del fabricant, però ens vam quedar amb una que permetia configurar al nostre gust precisió.

## 9.8- Llibreries d'aplicació de Microchip

Les llibreries d'aplicació de *Microchip* ens han permès utilitzar d'una forma prou fàcil i ràpida el port USB i la targeta Micro-SD. El funcionament d'aquestes llibreries és prou intuïtiu i per a fer-les funcionar hi ha prou amb estudiar un poc per damunt els fitxers \*.h. Per aquest motiu no entrarem en detalls de com utilitzar-les i ens limitarem a comentar quina ha sigut l'experiència d'ús i quins canvis hem realitzat sobre les mateixes per a fer-les funcionar en el nostre sistema, pel que sí el lector vol saber més sobre les mateixes, li recomanem la lectura de la documentació inclosa amb les mateixes.

### 9.8.1- Llibreria Memory Disk Drive File System

La finalitat d'aquesta llibreria és permetre al microcontrolador accedir al contingut d'una targeta SD, una targeta Compact Flash o una memòria USB que utilitze el sistema d'arxius FAT16/FAT32 per a organitzar el seu contingut. Un dels avantatges de la biblioteca és que també gestiona la comunicació amb el dispositiu de memòria extern, pel que no és necessari conèixer els detalls de funcionament del dispositiu extern ni que la nostra aplicació implemente el protocol de comunicació amb el mateix. El preu a pagar és l'eficiència a l'hora de gestionar els arxius.

Esta llibreria està formada pels següents arxius:

- **FSIO.c:** és el fitxer que conté totes les funcions necessàries per a gestionar un sistema d'arxius FAT16/FAT 32.

<sup>58</sup> ITOA: *IntegerTOAscii*, funció que converteix una variable entera a un caràcter ASCII

- **SD-SPI.c:** aquest fitxer és l'encarregat de gestionar la comunicació entre el PIC i la targeta de memòria SD/MMC.
- **InternalFlash.c:** permet utilitzar la memòria *flash* de programa lliure com un disc *flash*.
- **CF-PMP.c / CF-Bit transaction:** gestionen la comunicació amb una targeta *Compact Flash* connectada al sistema mitjançant el port paral·lel o una emulació *software* del mateix.

Tots aquests fitxers tenen el seu corresponent arxiu de capçalera en la carpeta `\Include\MDD File System` pel que, una vegada hagem triat el dispositiu al que accedirem, els tindrem que copiar també a la carpeta del nostre projecte o bé, especificar on es troben aquests arxius. A més, també haurem de copiar o localitzar un parell d'arxius més per a resoldre les dependències de la llibreria: el fitxer `FSDefs.h` i l'arxiu `FSConfig.h`. El primer el podem trobar en la citada carpeta `include` mentre que el segon el tindrem que agafar d'alguna de les *demos* existents en la llibreria que l'utilitze i editar-lo per ajustar-lo a les nostres necessitats.

### 9.8.1.1- Edició de l'arxiu `FSConfig.h`

El pas d'editar el fitxer `FSConfig.h` mereix una menció especial perquè no consisteix exclusivament en habilitar o deshabilitar funcionalitats d'aquesta llibreria sinó que, a més, cal afegir certa informació del nostre sistema al mateixa. Açò es deu a que estes llibreries s'han desenvolupat pensant en les plaques de demostració de *Microchip*, pel que, si volem que funcionen en el nostre sistema, haurem de proporcionar als fitxers de la llibreria esta informació, ja siga editant el fitxer `HardwareProfile.h` o bé, prescindint d'aquest i afegint la informació corresponent al fitxer `FSConfig.h`. Com el fitxer `HardwareProfile.h` depèn de la placa de demostració i pot ser un tant confús, per a simplificar, eliminarem la línia que incloïa aquest fitxer al projecte i afegirem les següents línies al inici del fitxer:

```
#define CLOCK_FREQ 32000000
#define GetSystemClock() CLOCK_FREQ
#define GetInstructionClock() 16000000
#define GetPeripheralClock() GetInstructionClock()
#define USE_SD_INTERFACE_WITH_SPI
#define USE_PIC24F
#define USE_16BIT
```

Com podem veure, estes línies indiquen a les llibreries quina serà la capa física que s'utilitzarà per a accedir al dispositiu de memòria no volàtil i quina és la velocitat de rellotge del sistema, la qual s'utilitzarà, entre altres coses, per a configurar la velocitat màxima del bus SPI.

En acabar d'afegir les línies vam procedir a configurar el comportament de la llibreria afegint comentaris en les característiques que no ens interessaven i llevant-los quan ens interessava la funcionalitat, quedant configurada finalment d'aquesta manera:

```
#define ALLOW_GET_DISK_PROPERTIES 1
#define FS_MAX_FILES_OPEN 1
#define MEDIA_SECTOR_SIZE 512
#define ALLOW_FILESEARCH
#define ALLOW_WRITES
#define ALLOW_DIRS
#define ALLOW_FSFPRTF
#define SUPPORT_FAT32
#define USEREALTIMECLOCK
```

Els canvis més significatius realitzats respecte al fitxer original és la possibilitat d'obtenir les propietats del disc i la d'utilitzar el mòdul del rellotge en temps real per a establir una relació temporal exacta amb els canvis realitzats en el sistema d'arxius. A més, al no necessitar obrir més d'un fitxer a la vegada, decidirem ajustar el valor de *FS\_MAX\_FILES\_OPEN* a 1.

### 9.8.1.2- Edició de l'arxiu *SD-SPI.h*

L'altre fitxer que vam tenir que editar per a fer funcionar la llibreria *MDD FS* en el nostre projecte va ser el que defineix la capa *hardware* de la llibreria: el fitxer *SD-SPI.h*. Així doncs, en aquest fitxer vam afegir les línies necessàries per a fer-li saber a la llibreria quins terminals del microcontrolador estan connectats al lector de targetes *Micro-SD* i quins són els registres del PIC sobre els que ha d'actuar per a controlar els corresponents terminals d'entrada/eixida i el mòdul SPI.

En concret, les línies afegides a la secció de *defines/structs* són les següents:

```
#define SD_CS                PORTBbits.RB13
#define SD_CS_TRIS           TRISBbits.TRISB13

//estos 4 defines sobren ja que en una uSD no hi ha ni CartDetect ni
//WriteEnable
//#define SD_CD              PORTFbits.RF0
//#define SD_CD_TRIS         TRISFbits.TRISF0
//#define SD_WE              PORTFbits.RF1
//#define SD_WE_TRIS         TRISFbits.TRISF1

#define MEDIA_SOFT_DETECT

#define SPICON1              SPI2CON1
#define SPISTAT              SPI2STAT
#define SPIBUF               SPI2BUF
#define SPISTAT_RBF         SPI2STATbits.SPIRBF
#define SPICON1bits         SPI2CON1bits
#define SPISTATbits         SPI2STATbits
#define SPIENABLE           SPI2STATbits.SPIEN

#define SPICLOCK             TRISFbits.TRISF4
#define SPIIN                TRISBbits.TRISB14
#define SPIOUT               TRISFbits.TRISF5
```

A l'examinar els canvis hauria de cridar-nos l'atenció el fet de no haver eliminat els *define* que estan comentats. La raó per la que els hem deixat és perquè volem destacar el fet que, tot i que esta llibreria està feta per a ser utilitzada amb targetes SD, també pot ser utilitzada per a accedir targetes Micro-SD encara que estes no tinguen tots els terminals de control presents en les seves germanes majors. De fet, esta llibreria, per a suplir l'absència del terminal *Card Detect* en les targetes Micro-SD utilitza un mecanisme software per a detectar-les i, per això, s'ha d'afegir la definició *MEDIA\_SOFT\_DETECT*.

### 9.8.1.3- Funcions utilitzades de la llibreria *MDD File System*

De totes les funcions comentades en l'apèndix B de la nota d'aplicacions AN1045 de *Microchip* sobre la llibreria *MDD File System*, nosaltres hem utilitzat les següents:

- **FSfopen():** és la funció encarregada d'obrir el fitxer, ja siga en mode lectura, escriptura o per a afegir contingut al mateix.

- **FSfwrite():** esta funció copia N bytes al fitxer que s'ha obert anteriorment amb *FSfopen()*. Nosaltres l'utilitzem junt amb *Sprintf()* per a escriure text amb format en l'arxiu.
- **FSfclose():** s'utilitza per a tancar el fitxer obert amb *FSfopen()* una vegada hem acabat d'escriure en el mateix.
- **FindFirst():** es tracta d'una funció encarregada de buscar arxius. La idea era utilitzar-la junt a la funció *FSGetDiskProperties()*, la qual no forma part de la llibreria MDD, per a calcular l'espai lliure de la targeta Micro-SD, però lamentablement no ha funcionat.

### 9.8.2- Llibreries USB

El marc de treball USB proporcionat per *Microchip* permet desenvolupar quasi qualsevol tipus d'aplicació USB utilitzant com a base els exemples genèrics proporcionats en la llibreria. Gràcies a açò no és necessari conèixer el funcionament d'aquest bus per a poder començar a desenvolupar una aplicació que l'utilitze, ja que sí es tracta d'una aplicació genèrica com puga ser la implementació d'un dispositiu d'interfície humana (HID) o la d'un dispositiu d'emmagatzemament massiu, el més segur és que l'aplicació funcione amb tan sols adjuntar els arxius corresponents i compilar-los després de realitzar les pertinents configuracions. En la pràctica, almenys en la nostra, la cosa no va ser tan simple com quan vam començar a utilitzar les llibreries *MDD File System* ja que després d'editar i configurar la llibreria, el sistema aparentava quedar-se bloquejat després de ser desconnectat del *host* USB.

Dins de la llibreria USB podem trobar els següents fitxers bàsics:

- **usb\_device.c:** és el fitxer que conté totes les funcions, macros i definicions necessàries per a fer funcionar el sistema com un dispositiu USB.
- **usb\_hal\_pic24\*.c:** aquest fitxer permet abstraure la capa hardware del mòdul USB del microcontrolador de manera que permet utilitzar amb més comoditat la *pila* USB.
- **usb\_host.c:** aquest fitxer és necessari sí volem que el nostre sistema actue com un dispositiu USB amfitrió o *host*, doncs inclou tot el necessari per a implementar esta funció.
- **usb\_otg.c:** conté la implementació de la modalitat *USB-On-The-Go* la qual permet al sistema intercanviar-se amb un altre dispositiu USB la funció d'amfitrió.

A part dels corresponents fitxers de capçalera, dins de la llibreria USB, també podem trobar el codi font necessari per a donar-li al sistema alguna de les aplicacions típiques que es basen en aquest bus. De tot el codi de referència inclòs en la llibreria, el que ens és més útil és el que es troba en la carpeta MSD Device Driver: *usb\_function\_msd.c*. Aquest fitxer, junt als fitxers *usb\_device.c*, *usb\_hal\_pic24f.c* i el *usb\_descriptors.c*, el qual podem obtenir de qualsevol projecte de demostració inclòs en la llibreria, ens permetrà transformar el nostre disseny en un pràctic lector de targetes Micro-SD. Això sí, per a fer-lo funcionar com a tal no hi ha prou amb afegir els fitxers editats, també s'ha de tenir en compte que, sí no hem configurat el PIC des de el principi per a que utilitze el PLL de 96 Mhz, haurem d'habilitar-lo i tornar a calcular qualsevol paràmetre que depenga de la freqüència d'oscil·lació interna, com per exemple els temporitzadors o els mòduls SPI, ja que al utilitzar aquest oscil·lador esta freqüència canvia també, pel que sí des de un principi tenim pensat utilitzar en un futur el port USB, el millor és habilitar aquest oscil·lador per a evitar tenir que reconfigurar els perifèrics com ens va passar a nosaltres.

### 9.8.2.1- Edició de l'arxiu *usb\_device.h*

A l'igual que vam fer amb la llibreria MDD File System, en la llibreria USB també hem eliminat la dependència de la mateixa amb l'arxiu *HardwareProfile.h*, pel que al no tenir un fitxer dedicat a informar a la llibreria de quins recursos disposa el sistema, es fa necessari afegir en algun dels arxius de capçalera esta informació. En aquest cas, el fitxer que hem modificat és l'arxiu *usb\_device.h* i les línies que hem afegit són les següents:

```
#define self_power
#define USE_USB_BUS_SENSE_IO
#define USB_BUS_SENSE    U1OTGSTATbits.SESVD
```

Sí el sistema que hem dissenyat no s'alimenta utilitzant el bus USB perquè té alimentació pròpia, és imprescindible afegir les dos últimes línies per a que la llibreria funcione correctament. Açò es deu a que, quan es defineix l'opció *self\_power*, la llibreria determina sí el sistema està connectat o no a un *host* USB comprovant l'estat del registre o terminal especificat per la definició *USB\_SENSE*.

### 9.8.2.2- Edició de l'arxiu *usb\_device.c*

Fins ara, per tal de fer funcionar les llibreries en el nostre projecte, ens havíem limitat a afegir opcions de configuració als fitxers de capçalera, però després d'editar el fitxer *usb\_device.h*, al no funcionar la part USB, ens vam vore obligats a canviar de dinàmica tan prompte com vam descartar la part *hardware* com a possible causant dels bloquejos que experimentava el sistema cada vegada que era desconnectat del bus USB.

El principal problema que ens vam trobar al passar a analitzar la part *software* va ser la documentació de la llibreria. Tot i que hi ha més d'una nota d'aplicació que parla de com implementar amb un PIC un dispositiu d'emmagatzemament massiu utilitzant les llibreries USB, cap dels que hem vist explica com fer-ho utilitzant una placa de disseny propi. De fet, la documentació assumeix que estem utilitzant una de les plaques de demostració del fabricant pel que no arriba a donar detalls de quines coses hem de tenir en compte en cas d'utilitzar les llibreries en un "ecosistema" diferent a l'original.

Al no tenir detalls de com adaptar les llibreries al nostre disseny, quan vam veure que podíem accedir a la targeta SD des d'un PC però el sistema deixava de respondre als polsadors quan el desconnectàvem del PC, vam pensar que el problema seria de tipus elèctric ja que donava la sensació que el microcontrolador no detectava quan havia sigut desconnectat del port USB. Així doncs, agafarem un voltímetre i mesurarem la tensió en el terminal d'alimentació del bus, al veure que la tensió obtinguda podria ser interpretada com un '0', modificarem el divisor resistiu per a que caiguera menys tensió en ell, però el resultat no va ser satisfactori. Va ser en eixe moment quan vam caure en el detall que en cap moment ens havíem preocupat per tenir un mecanisme que ens permetera detectar una connexió/desconnexió del bus USB. Revisant la fulla de característiques del microcontrolador vam veure que *Microchip* tampoc estava fent res especial per a detectar-ho, així que ens vam posar a buscar en la xarxa dissenys del circuit USB per a veure sí es podia fer tal qual ho havíem fet nosaltres o hi havia que fer alguna cosa més. Pel que vam veure, el mecanisme de detecció es podia fer o bé utilitzant el terminal VBUS com havíem fet nosaltres, o bé utilitzant un terminal genèric d'entrada/eixida compatible amb tensions d'entrada de 5V, però hi havia que fer-li ho saber a la biblioteca assignant a la definició *USB\_BUS\_SENSE* el nom del terminal o registre que s'encarregaria de detectar la tensió d'alimentació del bus USB. Al no haver afegit eixa línia al fitxer de capçalera vam pensar que la causa del problema era eixa, però no, ni afegint el nom del registre ni utilitzant un terminal d'entrada vam notar cap canvi, pel que al haver descartat la part elèctrica com a possible causa dels bloquejos, ens posarem a analitzar la part *software*.

Com era d'esperar, el primer pas que vam donar va ser esbrinar en quin lloc es quedava penjat el microcontrolador. Per a fer-ho, afegirem estratègicament en el programa punts d'interruptió i a cada interruptió anàvem mostrant per pantalla en quin punt del programa estàvem. Gràcies a açò vam veure que en realitat el que estava passant és que el microcontrolador no tornava a entrar en la funció *menu\_inicial()* perquè la màquina d'estats de la pila USB es quedava en l'estat *DEFAULT\_STATE*. Açò, segons el diagrama de flux de la figura 116, no hauria de ser un problema, però en aquell moment estàvem utilitzant una porció de codi font d'un dels exemples d'aplicació de *Microchip* que necessitava que l'estat fóra major a aquest per a executar qualsevol altra tasca que no estiguera relacionada amb el port USB. Canviar aquesta condició tampoc va servir de molt pel que el següent pas va ser assegurar-nos que el problema no tingués relació amb la màquina d'estats.

L'anàlisi de la màquina d'estat no ens va portar a cap conclusió clara sobre el que estava passant ja que, per molt que vam estar jugant amb ella, l'únic que aconseguíem era empitjorar les coses. Davant esta situació, per tal de no perdre més temps, ens vam posar a analitzar amb més calma el fitxer *usb\_device.c* en busca d'algun detall que no haguérem tingut en compte... i el vam trobar: aquest fitxer no sols necessita que se li indique quin mètode va utilitzar-se per a detectar una connexió/desconnexió del bus, sinó que a més, necessita que se li diga sí esta detecció va a ser *hardware* o bé mitjançant una rutina *software* que periòdicament va a estar comprovant l'estat del bus. Per a variar, afegir la línia `#define USE_USB_BUS_SENSE_IO` al fitxer de capçalera no va produir l'efecte esperat pel que no ens va quedar altra que continuar analitzant el codi font en busca d'alguna cosa que no encaixara.

Durant el nou anàlisi del codi font de la llibreria ens vam trobar amb accions que no acabaven d'encaixar amb el que nosaltres esperàvem, però les proves efectuades van demostrar que el codi havia d'estar escrit tal i com ho havia fet *Microchip*, així que vam continuar analitzant fins que ens vam trobar amb una cosa que no esperàvem: el procés d'inicialització del mòdul estava aparentment incomplet, doncs en cap moment vam veure que es connectara el mòdul USB i la seva interrupció. Al detectar aquest “detall” vam afegir a la funció *USBDeviceInit()* les següents dos línies:

```
U1CONbits.USBEN=1;           //Connecte el mòdul USB
USBEnableInterrupts();       //i habilite la seva IRQ
```

Afortunadament per a nosaltres, aquest canvi sí va ser decisiu i, després de compilar una altra vegada el programa, el sistema va deixar de bloquejar-se al ser desconnectat del PC.

Una vegada implementada correctament la funció de dispositiu d'emmagatzemament massiu ens quedava fer un canvi obligatori al fitxer *usb\_device.c*: impedir que el sistema entrara en suspensió quan el *host* li ho demanara. Açò es deu a que els PICs d'aquesta família tenen un error *hardware* que impedeix al mòdul USB tornar a activar-se per programa una vegada ha sigut suspès, pel que, en aplicacions com la nostra en la que no és necessària l'alimentació del bus USB per a funcionar, el que s'ha de fer és comentar o eliminar la línia de la funció *USBDeviceTasks()* en la que s'avisava a *USBSuspend()* per a iniciar la suspensió del mòdul.

### **9.8.2.3- Funcions utilitzades de les llibreries USB**

Una vegada tenim la llibreria adaptada a les nostres necessitats i funcionat, el seu ús és prou senzill i transparent ja que sols cal cridar un parell de funcions per a que aquesta llibreria s'encarregue de gestionar qualsevol procés relacionat amb l'accés a la targeta Micro-SD des de el port USB:

- **USBInit():** és l'encarregada d'inicialitzar tota la pila USB del sistema. Imprescindible cridar-la per a que el mòdul USB funcione.



- **USBAttach()**: el següent pas després de cridar a *USBInit()* és cridar a aquesta funció. Si no es fa, el sistema ignora els esdeveniments USB ja que la màquina d'estats esta en un nivell inferior al necessari per a emprendre una acció.

## 9.9- Firmware: coses a millorar i a implementar.

Actualment, una de les coses que sempre es pot millorar en un sistema digital micro-controlat és l'apartat del programa que executa el processador del sistema. El nostre disseny no és una excepció i el programa principal té moltes coses que poden ser millorades.

La millora més important que es pot fer és la gestió i ús de la memòria. Tal i com s'ha quedat el desenvolupament del *firmware* al acabar el projecte es pot dir que este apartat és deficient ja que la RAM que li queda al sistema després de compilar el codi font és d'un 8%. Això es deu a que utilitzem un 80% de la RAM per a emmagatzemar les captures necessàries per a completar el procés de calibratge i, com no s'assigna de forma dinàmica, sinó que sempre està reservada eixa quantitat, estem desaprofitant RAM que podria venir bé per a afegir noves funcions. Així doncs, una de les coses que s'hauria de fer és depurar el codi font per a evitar que, al utilitzar la funció *mallo()*, el sistema es torne inestable. Una altra opció seria reduir considerablement el *buffer* assignat al sensor d'inèrcia tenint en compte que quan més és reduït és aquest, més fàcil serà que es desborden les variables temporals utilitzades per a guardar el resultat de l'*off-set* calculat en cada iteració del bucle de la funció *Adis\_Sensor\_Calibration()*.

Program Memory [Origin = 0x200, Length = 0x2a9f6]

| section                            | address | length (PC units) | length (bytes) (dec) |
|------------------------------------|---------|-------------------|----------------------|
| .text                              | 0x200   | 0x1f00            | 0x2e80 (11904)       |
| .const                             | 0x2100  | 0x47a             | 0x6b7 (1719)         |
| .text                              | 0x257a  | 0xc026            | 0x12039 (73785)      |
| .dinit                             | 0xe5a0  | 0x232             | 0x34b (843)          |
| .text                              | 0xe7d2  | 0x380             | 0x540 (1344)         |
| Total program memory used (bytes): |         |                   | 0x15dfb (89595) 34%  |

Data Memory [Origin = 0x800, Length = 0x4000]

| section                         | address | alignment gaps | total length (dec) |
|---------------------------------|---------|----------------|--------------------|
| .nbss                           | 0x800   | 0              | 0x278 (632)        |
| _00e98c4050420489               | 0xa78   | 0              | 0x200 (512)        |
| _00e98d9050420489               | 0xc78   | 0              | 0x200 (512)        |
| .nbss                           | 0xe78   | 0              | 0x2d0 (720)        |
| .nbss                           | 0x1148  | 0              | 0x3c (60)          |
| .ndata                          | 0x1184  | 0              | 0x20 (32)          |
| .nbss                           | 0x11a4  | 0              | 0x1a (26)          |
| .dconst                         | 0x11be  | 0              | 0x2a (42)          |
| .bss                            | 0x11e8  | 0              | 0x6 (6)            |
| .data                           | 0x11ee  | 0              | 0x2 (2)            |
| _018b94605042048a               | 0x1200  | 0              | 0x20 (32)          |
| .bss                            | 0x1220  | 0              | 0x2f10 (12048)     |
| .data                           | 0x4130  | 0              | 0x200 (512)        |
| .data                           | 0x4330  | 0              | 0x26 (38)          |
| Total data memory used (bytes): |         |                | 0x3b46 (15174) 92% |

Dynamic Memory Usage

| region | address | maximum length (dec) |
|--------|---------|----------------------|
| heap   | 0       | 0 (0)                |
| stack  | 0x4356  | 0x4aa (1194)         |

Maximum dynamic memory (bytes): 0x4aa (1194)

Figura 146: Detall de l'espai ocupat pel *firmware* en la memòria *flash* del microcontrolador i la memòria RAM lliure.

Una altra de les millores que es podria fer és optimitzar el programa ja que hi han operacions que s'han duplicat. També seria interessant fer ús dels punters i de revisar amb calma el codi en busca de d'operacions que no és necessari fer per tal de fer el sistema més eficient, doncs com podem veure en la figura 146, amb les optimitzacions del compilador deshabilitades, el *firmware* ocupa al voltant d'una tercera part de la memòria total de programa.

També resultaria interessant documentar-se tranquil·lament sobre els modes de funcionament de baix consum del microcontrolador i del sensor d'inèrcia i implementar-los en el *firmware* per tal d'augmentar la duració de la bateria. Al mateix temps, també hauríem d'implementar un sistema d'estimació de bateria restant per a, d'eixa manera, al detectar un nivell de bateria crític, tancar el fitxer que conté les lectures dels sensors evitant la corrupció del mateix. A més, aquest sistema d'estimació i control de bateria també podria utilitzar-se per a gestionar millor els perifèrics ja que, per exemple, no arribem a desconnectar el mòdul USB en cap moment.

En el que respecta a la part relacionada amb l'ADIS16400, seria interessant aprofitar d'alguna manera el sistema d'alarmes que incorpora i veure d'implementar algun mètode per a realitzar un calibratge ràpid dels magnetòmetres. D'esta manera es podrien utilitzar per tal de compensar les desviacions de les lectures produïdes per l'efecte *random walk* dels giroscopis. A més, també s'hauria d'implementar algun mecanisme per a corregir periòdicament les desviacions en les lectures produïdes pels errors de *bias* dels sensors, sempre tenint en compte que, aquestes variacions, depenen de la temperatura. Una altra opció seria donar a l'usuari l'opció de poder elegir quins sensors guardar i/o mostrar en pantalla per a simplificar el procés de lectura i interpretació de dades.

L'apartat de la targeta de memòria també podria ser millorat afegint la possibilitat de gestionar de forma bàsica els arxius que conté. De la mateixa manera, el *firmware* també hauria de ser capaç de detectar l'espai lliure en la targeta i encarregar-se d'eliminar els arxius més vells en cas de detectar que la targeta s'està quedant sense espai lliure.

En cas de continuar utilitzant-se una pantalla LCD de 20x2, hauríem de millorar el sistema de menús per tal de simplificar el procés d'afegir noves opcions i menús al mateix, encara que, sincerament, el millor seria canviar de tipus de pantalla o bé directament prescindir d'ella i implementar en una de les UARTs disponibles un terminal al que poder connectar-se amb, per exemple, un ordinador i poder controlar i accedir a tots els recursos del sistema, o bé fer el mateix però utilitzant el bus USB, cosa que pot ser seria més lògic ja que hui en dia aquest port gaudeix d'una popularitat enorme, pel que ja posats, podríem aprofitar per a utilitzar-lo com un altre mètode més de comunicació amb l'exterior, doncs igual que el podríem connectar a un PC per a accedir als recursos del sistema, també podríem fer que utilitzarà un altre dispositiu USB, com per exemple una targeta *WiFi* USB, per a transmetre dades a l'exterior.

Finalment, sí continuarem treballant amb el mateix receptor GPS, seria recomanable configurar-lo correctament per a augmentar l'eficiència del sistema i, independentment de sí canviem de receptor GPS o no, resultaria interessant implementar utilitzar el camp de la frase GPRMC corresponent a la data per a configurar el calendari del mòdul RTCC ja que a nosaltres se'ns va oblidar fer-ho. De la mateixa manera, també es podria escriure una funció que s'encarregara de guardar en un fitxer de text el registre de posició pues, ara mateixa, el programa sols contempla la possibilitat de fer-ho quan s'està utilitzant el sistema com a *datalogger*.

## 10- Implementació física i pràctica del sistema

Al principi del projecte, com sols ens faltava el sensor d'inèrcia per a poder començar el projecte i fins Setembre no tindríem accés al mateix, vam aprofitar que disposàvem d'una placa de demostració amb un PIC18F per a anar avançant la part del *Software*. Més avant, quan vam començar a treballar amb el sensor d'inèrcia, les limitacions dels microcontroladors de 8 bits es van fer tan evidents que vam tenir que dissenyar un nou sistema basat en un microcontrolador de 16 bits i adaptar a la nova plataforma el que ja teníem fet.

### 10.1- Implementació inicial amb un PIC18F46J50

Les primeres proves de concepte d'aquest projecte es van fer utilitzant la placa de demostració la figura 79. Esta placa dissenyada per *Tiertex Studios* conté els elements bàsics per a poder utilitzar tots els recursos del microcontrolador PIC18F46J50:

- Connector mini-usb per a desenvolupar aplicacions basades en el bus USB.
- Regulador de 3.3 volts per a alimentar el sistema des del bus USB o des d'una font de 5 volts.
- 3 polsadors: 1 per al *reset* del sistema i 2 connectats al port B per a ús general.
- 2 LEDs connectats al port E.
- Oscil·lador del sistema de 12MHz i secundari de 32,768KHz.
- 40 dels 44 terminals del PIC disponibles per a ser utilitzats per l'usuari.

Això vol dir que, per a començar a fer proves, sols vam tenir que connectar el GPS i la pantalla LCD als terminals corresponents del PIC<sup>59</sup>, programar el microcontrolador amb un *Kit Pic* i veure el resultat.

El primer element que vam provar van ser els recursos bàsics del microcontrolador: temporitzadors i ports d'entrada/eixida. Després de comprovar que els LEDs parpallejaven al ritme imposat pel temporitzador 1 i que el procés s'interrompia al prémer un polsador, vam passar a provar la pantalla amb el típic *Hola món*.

Aquesta segon prova no va ser tan ràpida com la primera ja que, primer vam tenir que adaptar les rutines de manipulació del LCD que vam obtenir del llibre *Learning to Fly the PIC: Programming 16 Bit Microcontrollers in C* per a que funcionaren amb el nostre PIC i la nostra pantalla. En acabar, com no apareixia res per pantalla i aparentment el problema no eren les rutines de manipulació del LCD, ens va tocar comprovar sí el problema era una mala connexió o bé el *display* estava trencat. Afortunadament el problema el vam causar nosaltres al malinterpretar la funció del terminal *Enable* del LCD i connectar-lo al terminal de *Chip Select* del port paral·lel del PIC, quan en realitat devia d'anar connectat al terminal d'escriptura PMWR.

Una vegada solucionat el problema amb la pantalla ens vam posar a treballar amb el receptor de GPS. Ací, la idea era capturar 512 caràcters de la UART a la que estava connectat el GPS i mostrar-los en pantalla per a veure sí la comunicació estava configurada correctament. Però va ser més complicat de fer que de dir, pues després d'escriure la rutina que realitzaria aquesta funció, el PIC no mostrava res per pantalla i es quedava bloquejat. Després d'esbrinar que el bloqueig es produïa perquè el microcontrolador es quedava esperant que a la UART li arribara un caràcter que mai arribava, ens vam posar a comprovar connexions i, al veure que tot estava en ordre i que en cap moment cap dels LEDs associats al receptor parpallejava, vam sospitar d'un possible defecte de fabricació. Així doncs, alimentarem el receptor sense connectar-lo a la placa de demostració i, amb

---

59 L'esquemàtic de les connexions d'aquesta implementació el podem trobar en l'apèndix .

l'ajut d'un oscil·loscopi, vam mirar si l'eixida del mateix oscil·lava. Com no va ser el cas, confirmarem les nostres sospites i, com el receptor era nou, vam tenir que gestionar amb RS la devolució i substitució per un altre idèntic, cosa que no va ser tan fluida com ens agradaria ja que RS, per defecte, quan els tornes un componen defectuós, no assumeix que vols que te'l canviïn per un altre igual. Com desconeixíem aquest detall, al veure que passaven els dies i no ens arribava un altre receptor, ens vam tornar a posar en contacte amb ells per a veure que passava... pel que amb esta ximplesa estiguérem quasi una setmana sense poder progressar amb el receptor de GPS.

Però els problemes amb el GPS no es van acabar ací. Una vegada ens va arribar l'unitat de substitució, vam provar la funció que havíem escrit per a mostrar en pantalla el *buffer* de recepció de la UART i esta, com era d'esperar, no funcionava correctament. Després de revisar i corregir els errors més evidents de programació, la cosa continuava sense funcionar, encara que esta vegada el problema no era un error de programació, sinó falta d'informació sobre el comportament del receptor de GPS. Resulta que nosaltres suposàvem que tan prompte com se li aplicava l'alimentació, el receptor començava a enviar per la UART frases NMEA, pel que tant prompte que com s'omplia el *buffer* de 128 caràcters que havíem assignat a la UART, desconnectàvem les interrupcions i iniciàvem la cerca del primer caràcter de la frase per tal de fer una presentació coherent de dades. Pues bé, resulta que el receptor, una vegada iniciat, envia informació bàsica sobre ell i, una vegada ha acabat, comença a enviar per la UART les 5 frases NMEA que té configurades per defecte. Aquesta falta d'informació també ens va retardar uns quants dies, però almenys vam obtenir una cosa de profit: el model i fabricant del *chipset* del receptor.

En el moment que vam aconseguir mostrar el *buffer* per pantalla, vam escriure una rutina que s'encarregara de buscar en el *buffer* la frase NMEA que conté la informació bàsica de navegació: la GPRMC. Durant la depuració d'aquesta funció va aparèixer un problema que no hem pogut solucionar: la recepció de frases NMEA corruptes.

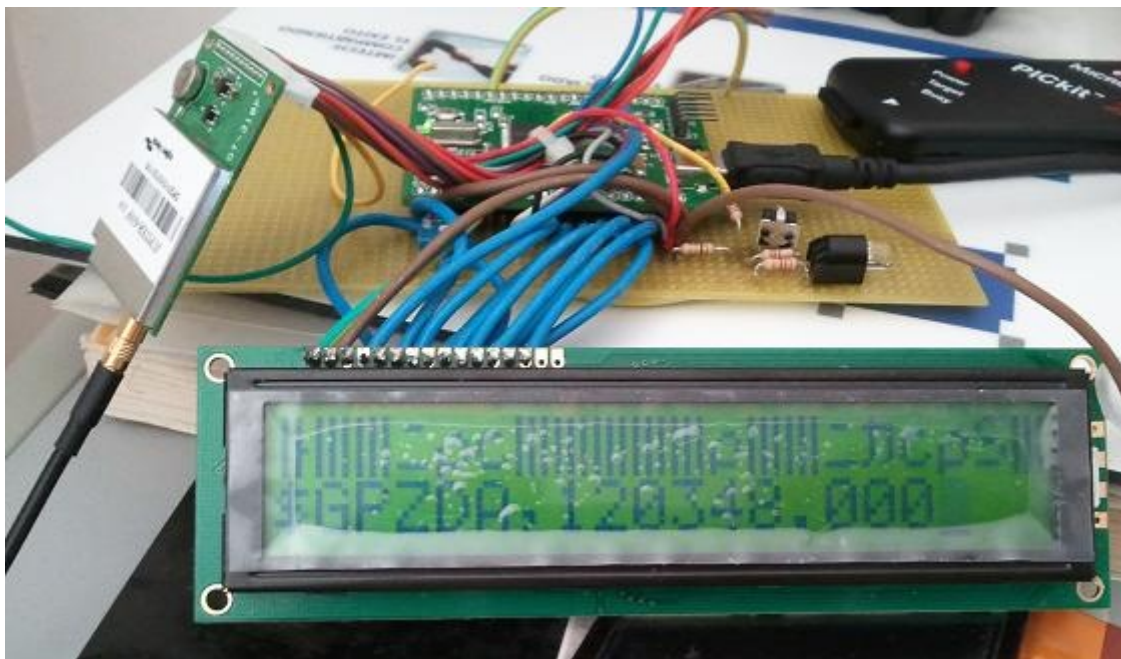


Figura 146: Recepció corrupta de frases NMEA.

En un principi tot indicava que este problema era de tipus *software* ja que moltes vegades el programa es quedava bloquejat o mostrava quasi tota la frase correctament. I en part va ser d'eixa manera perquè el programa perdia el rumb al desglossar els camps de la frase, però no per un error de programació, sinó perquè les dimensions dels camps de les frases NMEA especificades en la fulla de característiques del receptor no coincideixen amb el que envia aquest. A l'adonar-nos d'esta diferència pensàvem que ja havíem solucionat el problema, però no: al cap d'una estona de mostrar

correctament la frase es va tornar a repetir l'incident i no va quedar altra que començar a descartar possibles fonts de problemes. La primera va ser netejar part del codi. La segon, com el receptor estava connectat a la placa amb cables tal i com s'aprecia en la figura 146, consistí en afegir un filtre EMI de ferrita al cable. Al no tenir cap efecte aquesta última mesura, decidirem analitzar més detingudament que passava dins del PIC per sí ens aclaria alguna cosa, però l'únic que vam aconseguir va ser confirmar que la recepció falla degut a errors de trama, és a dir, de sincronisme. A més, durant tot el temps que vam estar fent proves amb el receptor, en cap moment va arribar a fixar la nostra posició, així que, davant la possibilitat que el problema fos una altra vegada un receptor GPS defectuós, decidirem apartar temporalment aquesta part i dedicar la nostra atenció al sensor d'inèrcia.

Sí controlar el GPS va ser una aventura, fer funcionar el sensor d'inèrcia va ser tota una odissea. Per què? Doncs perquè per a assegurar-nos que accedíem correctament al ADIS16400, decidirem llegir un registre el valor del qual coneixíem: el registre d'identificació. Desgraciadament, la fulla de característiques conté una errada que ens va tocar prou els nassos: sols especifica el valor del registre ID per a la IMU 16405 tot i que la fulla de característiques és també la del ADIS16400. L'omissió d'aquest valor pot fer-nos pensar que no s'està llegint correctament el sensor sí el primer registre que intentem llegir és aquest i, a més, no es cau en el detall que el valor hexadecimal especificat per a eixe registre en la fulla de característiques correspon al valor decimal del model del sensor.

I això és el que ens va passar: no ens vam adonar d'aquesta analogia i, com no rebíem periòdicament la mateixa serie de valors, vam interpretar que, o bé estàvem fent alguna cosa mal, o bé la comunicació es corrompia en algun moment.

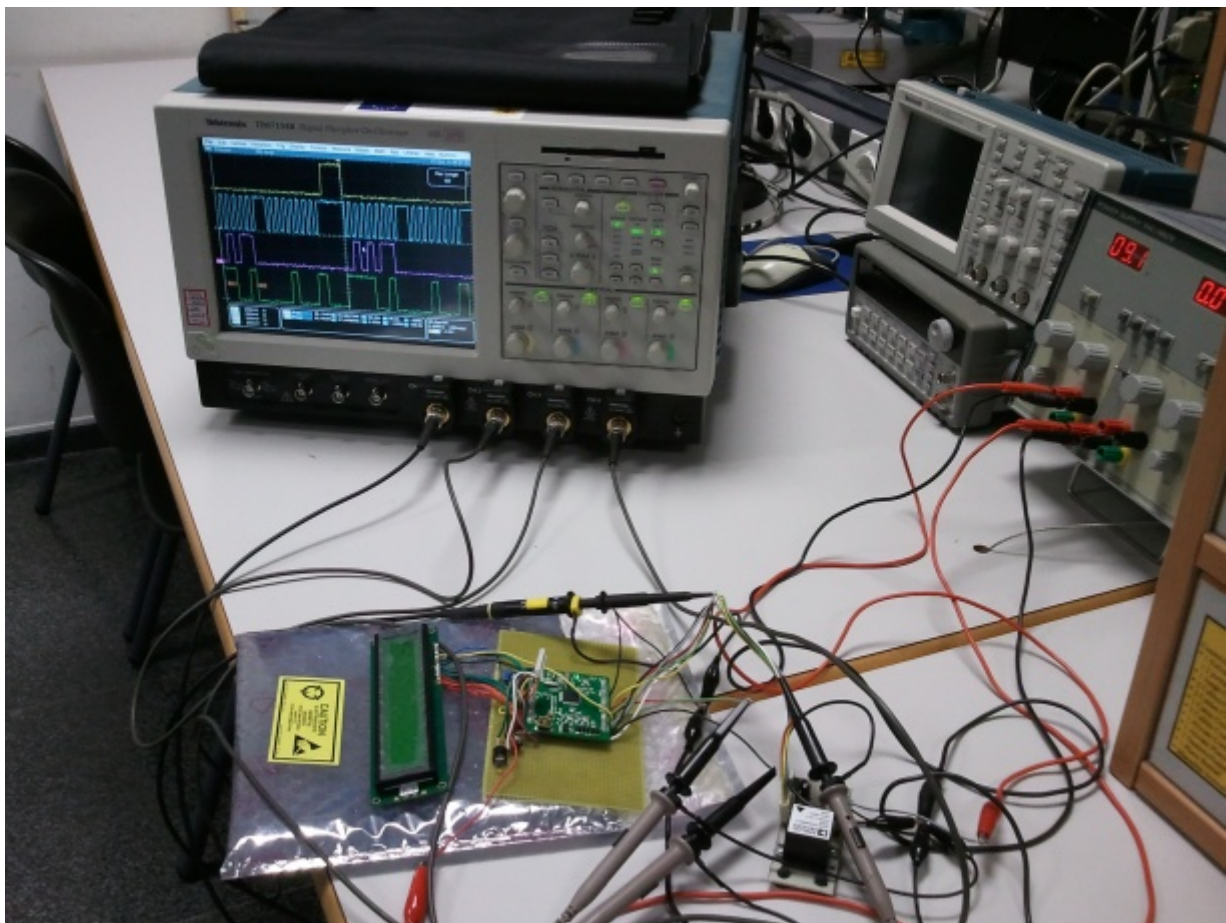


Figura 147: Muntatge per a poder capturar les quatre senyals involucrades amb la comunicació amb el ADIS16400.

Després de perdre varies vesprades fent proves amb els modes de comunicació del protocol SPI i revisant la rutina encarregada de llegir el sensor, al veure que no aconseguíem cap progrés

significatiu, decidirem contactar amb el suport tècnic d'*Analog Devices*<sup>60</sup>, primer utilitzant un formulari *online* i després, al dir-nos que no donaven suport a universitats, directament amb l'enginyer encarregat d'aquests sensors: *Mark Looney*.

A l'explicar-li el que passava a Mark, aquest ens va demanar que li facilitarem més informació i, a ser possible, una imatge amb les senyals involucrades en la comunicació per tal de poder diagnosticar el problema amb més facilitat. Una vegada li facilitarem el que ens demanava, ens va confirmar que hi havia corrupció de dades en la comunicació amb la IMU i ens va demanar que explicarem això de *estem rebent una serie de valors que es repeteixen amb el temps*. Al dir-li que volíem dir que sempre estàvem rebent els caràcters 0xF211, 0x4010, 0x4080, 0x0011 i 0x0000 ja ens va donar per segur que teníem un problema de corrupció de dades en la comunicació perquè un d'eixos valors, concretament el 0x4010, era el valor vàlid per a l'ADIS16400, pues en la fulla de característiques del sensor inèrcia hi havia una errada. Quan ens va dir açò ens va deixar bocabadats i no ens vam poder resistir a suggerir-li que la corregiren el més prompte possible, perquè més d'un anava a tenir problemes per eixe error... i la seva resposta va ser: *Què et jugues?*

Tan prompte com ens van confirmar que el problema de comunicació era degut a la corrupció de dades, el primer que vam fer va ser assegurar-nos que totes les senyals involucrades en la comunicació complien les especificacions temporals i d'amplitud. Al comprovar que tot pareixia estar dins dels límits acceptables, se'ns va ocórrer canviar la velocitat del rellotge inicial de 188KHz per la de 750KHz.... i premi: adéu a la corrupció de dades.

Una vegada solucionat el problema de la corrupció de dades, després de veure com se les gastaven els d'*Analog Devices* amb les errades en la fulla de característiques, procedirem a comprovar que el contingut dels restants registres coincidien amb el que s'indicava en la *datasheet*. I menys mal que ho vam fer perquè de no haver-ho fet, ni ens hauríem assabentat que molts dels registres no contenien el valor que els corresponia, ni *Mark Looney* ens hagués explicat que, quan els terminals DIOx són configurats com a entrades, llegir la part alta del registre GPIO\_CTRL equival a llegir aquests terminals.

Siga com siga, al detectar aquest "error" per tal de saber sí es tractava d'una altra errada de la fulla de característiques o, en el pitjor dels casos, era algun defecte de fabricació del sensor, vam tornar a parlar amb *Mark Looney* i este, després de demanar-nos nombre de serie i lot del sensor, ens va confirmar que no s'havia detectat cap incident amb IMUs del mateix lot i que segurament, com ja li havíem insinuat, esta des-configuració era un efecte secundari d'haver estat fent proves de comunicació amb l'ADIS16400 utilitzant els diferents modes SPI. I eixe degué de ser el motiu, perquè després d'escriure de nou els valors per defecte de cada registre ja no tornàrem a experimentar el problema.

Com aparentment ja teníem tot en orde, el següent pas fou mostrar per pantalla els valors d'eixida dels acceleròmetres i giroscopis. El procés no va ser excessivament complicat, però com els valors d'eixida oscil·laven prou a pesar que el sistema estava en repòs, el resultat va ser un poc decepcionant. Al no saber fins quin punt era normal una variació d'uns quants LSBs en l'eixida dels giroscopis, ens tornarem a posar en contacte amb *Mark Looney* i este ens va comentar que, després del calibratge, la variació normal era menor de mig grau/segon i que, de no ser d'esta manera, hauríem de sospitar de soroll en el sistema. Nosaltres sols vam llegir la part del soroll, pel que vam afegir un filtre de ferrita sobre el cablejat de comunicació amb el sensor i al veure que la cosa millorava, vam donar per suposat que teníem un problema de soroll i que la millor manera de solucionar-lo seria fent una placa on muntar tots els components del sistema.

Al arribar a aquesta conclusió, vam començar a buscar informació de com podríem detectar o mesurar, per exemple, ones i tot el que vam trobar apuntava a que la millor forma de fer-ho era,

---

60 En l'apèndix 2 s'han adjuntat les converses per eMail que es van mantindre amb personal d'*Analog Devices*.

primer registrar durant un període de temps de varies hores el canvi d'altura de les ones i, després, realitzar la transformada de *Fourier* dels valors obtinguts. Açò ens va fer veure que el microcontrolador actual anava a quedar-se curt en tots els aspectes, ja que sols per a poder registrar les ones durant eixa quantitat de temps deuríem prescindir de la pantalla per a poder tenir prou terminals per a instal·lar un lector de targetes SD i, a més, sacrificar el 60% de la RAM del sistema per a poder utilitzar les llibreries FAT de *Microchip*.

Aquestes exigències, sumades al soroll del sistema i a l'enrevessat mètode de lectura/escriptura del sensor d'inèrcia, van motivar l'abandó d'aquesta plataforma i la posterior migració a un sistema de 16 bits.

## 10.2- Implementació final amb un PIC24FJ256GB106

La implementació d'aquest projecte amb un microcontrolador de 16 bits no va ser tan problemàtica com la implementació amb el PIC de 8 bits, però no per això ha estat lliure de problemes. Això es deu a que les problemes han sigut de diferent natura. De fet, gràcies a la primer muntatge, les dificultats que hem tingut en aquesta implementació han sigut quasi exclusivament de programació, pues partíem de rutines funcionals per a la manipulació bàsica dels perifèrics externs.



Figura 148: Prototip de la Implementació amb un PIC24FJ256GB106 mostrant la nostra localització en l'instant de la fotografia.

De la mateixa manera que en el primer muntatge vam començar posant a prova la pantalla, en aquest vam fer el mateix. La diferència està en que, com era d'esperar, en aquesta configuració la pantalla va funcionar a la primera.

El segon pas també va ser mostrar en pantalla el *buffer* de recepció de la UART. En aquest cas el funcionament no va ser instantani perquè aquest PIC té una UART amb un funcionament diferent al

de la UART d'un PIC18F. Les diferències van ser tan significatives que vam tenir que reescriure tota la rutina d'atenció a la interrupció de la UART i, a més, llegir-nos les anomalies *de silicona* d'aquesta família de PICs perquè els errors de trama i de *overflow* del *buffer* eren molt més freqüents que en el PIC de 8 bits. Després d'invertir moltes hores intentant solucionar el problema i no aconseguir una milloria destacable, vam tornar a deixar de costat el desenvolupament de la part del GPS i ens vam centrar en el sensor d'inèrcia fins que aconseguirem el que desitjàvem d'ell.

La part encarregada de gestionar l'ADIS16400 també va funcionar a la primera, encara que a mesura que anàvem progressant en la seva implementació anàvem trobant-nos amb detalls que no havíem tingut en compte o en situacions en les que no havíem estat en l'anterior disseny, pel que vam tenir que corregir el que ja teníem fet i afegir noves funcions.

Així pues, la primera cosa que vam fer va ser mostrar en pantalla les lectures dels acceleròmetres i dels giroscopis. Al comprovar que les variacions en l'eixida els giroscopis continuaven sent excessives, tornàrem a rellegir el correu que ens havia enviat *Mark* per a veure sí havíem obviat algun detall i caiguérem en el detall del calibratge. Com en la fulla de característiques del sensor no es donaven detalls del procés, ens posarem a buscar per la xarxa i un dels documents que trobarem fou la nota d'aplicació AN-1041 del fabricant. En aquesta nota, sense arribar a donar una explicació detallada, donava la suficient informació com per a poder calibrar tant els giroscopis com els acceleròmetres. Això sí, el procediment, tot i que no ens va parèixer excessivament complicat, ens va plantejar una pregunta: com ens les arreglem per prendre 100 segons de mostres sí sols disposem de memòria suficient per a capturar uns 8 segons? La lògica va suggerir que combinàrem l'assignació dinàmica de memòria<sup>61</sup> amb la captura de les lectures del sensor dins d'un bucle de 12 iteracions, però la pràctica ens va demostrar que el mètode era excessivament inestable com per a utilitzar-lo i millor reservar una gran quantitat de RAM com a *buffer* de lectura del sensor.

Durant les proves que vam fer a la rutina de calibratge vam observar que el procés durava quasi la mitat del que s'esperava. Al investigar les causes d'aquest fenomen vam descobrir que aquest efecte es produïa perquè en la funció de lectura del ADIS no esperàvem a que el polse de *Data Ready* del sensor d'inèrcia passara a nivell baix, pel que, com el procés de lectura durava menys temps del que dura eixe polse, al tornar a llegir la IMU, la senyal *Data Ready* encara continuava en nivell alt i llegiem una altra vegada el mateix valor.

Una vegada solucionat el problema de la doble lectura, com ja teníem clar que el microcontrolador escollit anava a quedar-se curt per al que volíem fer en aquest projecte, vam decidir fer ús del lector de targetes micro-SD de la nostra placa per a emmagatzemar els registres del sensor.

L'única complicació que vam tenir a l'hora de fer funcionar el lector va ser de tipus *software* tot i que, quan vam dissenyar el circuit elèctric d'esta part, ens vam quedar amb el dubte de sí funcionaria o no, pues en alguns dels circuits que ens vam trobar mentre buscàvem informació sobre les targetes SD deien que eren necessàries resistències *pull-up* en les línies de transmissió per tal de millorar la compatibilitat amb les targetes. Nosaltres, com no li acabàvem de veure sentit a afegir estes resistències, vam optar per no posar-les i, pels resultats obtinguts, es confirma que estes no són imprescindibles, almenys amb les targetes micro-SD.

Com llegir el contingut de la targeta SD es feia pesat perquè a cada vegada que volíem bolcar el seu contingut, havíem de treure-la del lector del prototip i utilitzar un adaptador micro-SD a SD per a llegir-la en un PC, vam pensar que seria bona idea utilitzar el port USB i les llibreries d'emmagatzematge massiu de *Microchip* per a fer més còmode el procés.

Fer funcionar aquest port ens va costar prou més que el lector de targetes ja que el sistema es quedava bloquejat al desconnectar-lo del bus USB. En un principi tot pareixia indicar que això passava perquè el microcontrolador no se'n adonava que havia sigut desconnectat del host, així que

---

61 En el llenguatge de programació C aquesta tècnica sol implementar-se mitjançant la funció *malloc()*.



analitzarem el circuit per tal de veure si efectivament eixe era el problema. Al mesurar la tensió en el terminal  $V_{BUS}$  i buscar en la fulla de característiques del PIC el valor de tensió mínim per a considerar una entrada com un nivell alt<sup>62</sup>, vam caure en el compte que 2.5 Volts podia quedar al limit del que pot considerar-se un '1', pel que per a assegurar-nos que el problema no era eixe, canviarem el valor del divisor resistiu de manera que la tensió en  $V_{BUS}$  s'aproximara als 3.3 Volts.

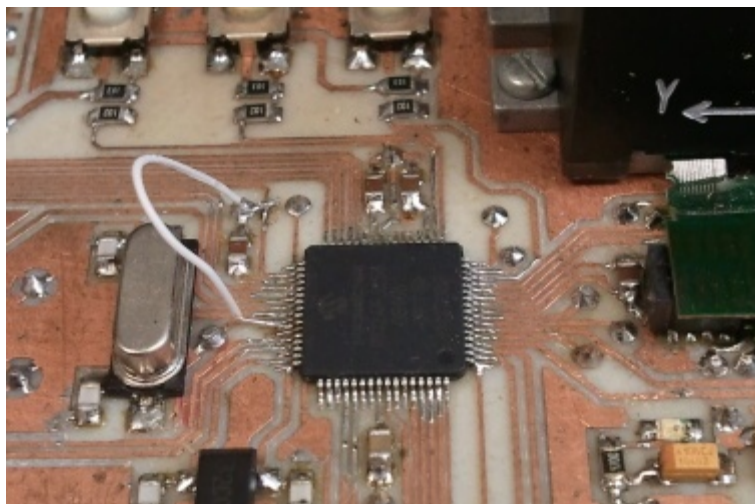


Figura 149: Connexió temporal de VBUS a un terminal d'entrada/eixida.

La modificació del divisor de tensió no va produir cap canvi aparent així que ens posarem a buscar per la xarxa exemples de circuits per tal de comparar-los amb el nostre. Quasi tots els circuits que ens vam trobar utilitzaven un terminal genèric d'entrada/eixida per a detectar la connexió/desconnexió del bus, així que escollirem un terminal dels que havien quedat lliures i el connectarem a VUSB i re-configurarem la llibreria de *Microchip* per a que utilitzara aquest *pin*, però no sols no va servir de res, sinó que a més, ens va marejar un poc més pues, de sobte, teníem 5 volts en cada extrem del divisor de tensió. Al buscar la causa d'açò ens vam donar compte que el terminal que havíem utilitzat per a la nova connexió era justament l'eixida de l'alarma del RTCC i, encara que nosaltres no utilitzàvem aquesta funció, la teníem habilitada per programa. Solucionar aquest conflicte tampoc va servir de molt així que com, des de el punt de vista del circuit, tot estava en ordre ens vam centrar en la part *software*. En eixa part vam veure varies coses que no ens pareixen lògiques, les canviarem i adéu als bloquejos produïts per les desconexions del bus USB. Al posar-nos a treballar de nou amb el receptor de GPS ens vam emportar una agradable sorpresa: havia aconseguit fixar la nostra posició un any després de tenir-lo en el nostre poder. Així doncs, al veure que el receptor no tenia perquè estar defectuós com havíem arribat a pensar a finals de l'any anterior, vam tornar a dedicar un poc de temps a analitzar el problema, però sense sort, pel que decidirem millorar i fer més robusta la part encarregada de buscar en el *buffer* la frase GPRMC.

La millora va ser evident, però de vegades els errors de sincronisme són tan freqüents que poden passar més de 4 segons per a obtenir una trama GPRMC no corrupta. Cal dir que aquest retràs es podria reduir considerablement si haguérem pogut configurar el receptor de GPS per a que ens enviara sols la trama GPRMC amb una freqüència inferior a un segon, el valor per defecte del receptor. Desafortunadament no ens va donar temps a dissenyar una placa que adaptara els nivells de tensió del receptor amb els del port serie d'un PC, pel que esta part no té un funcionament òptim.

<sup>62</sup> Els valors mínims que defineixen els valors lògics '0' i '1' venen especificats en la taula 29-7 de la *datasheet*.

## 11- Prova experimental del sistema

Quan vam iniciar aquest treball de final de carrera, ens vam proposar dissenyar i implementar un sistema de control d'estabilitat que, a més, ens facilitara la informació de localització del sistema. Al buscar una possible aplicació per a un sistema d'aquestes característiques se'ns va ocórrer que resultaria interessant utilitzar-lo en l'oceà per tal de monitoritzar "l'estabilitat" del mateix. Així doncs, si instal·larem el sistema en l'interior d'una boia hauríem de ser capaços, no sols de deduir l'estat del mar en les coordenades proporcionades pel receptor GPS, sinó que a més hauríem de poder calcular i representar els moviments experimentats per la boia, tant en l'apartat bidimensional com en el tridimensional, d'una manera prou precisa.

Per a fer-nos una idea de com de viable era aquest procediment vam fer un petit experiment: registrar el trajecte en cotxe des de la casa de l'autor des de sa casa fins el seu lloc de treball: l'hospital de Gandia.



Figura 150: Representació parcial del trajecte realitzat un dia de treball des de l'entrada a Gandia fins a aparcar i arribar a l'hospital.

A la vista de la figura 150 queda clar que la precisió de la localització proporcionada pel receptor de GPS utilitzat ronda els 3 metres ja que, com es pot apreciar en esta figura, una vegada vam aparcar el cotxe i vam fer el trajecte a peu cap a l'Hospital, aquest va ser capaç inclús de detectar els canvis de vorera que vam fer a propòsit o el moviment que vam fer al entrar per la porta auxiliar del taller dels mecànics del servici de manteniment del Hospital Francisc de Borja.

```
$GPRMC,055846.003,A,3857.96721,N,00011.62025,W,0000.00,203.41,090812,,,A*42
$GPRMC,055850.003,A,3857.96717,N,00011.62018,W,0000.91,066.10,090812,,,A*43
$GPRMC,055851.003,A,3857.96717,N,00011.62015,W,0001.26,075.14,090812,,,A*44
$GPRMC,055853.003,A,3857.96718,N,00011.62012,W,0001.71,076.09,090812,,,A*43
$GPRMC,055855.003,A,3857.96720,N,00011.61934,W,0002.12,095.22,090812,,,A*42
$GPRMC,055857.003,A,3857.96703,N,00011.61783,W,0001.81,104.82,090812,,,A*49
$GPRMC,055859.003,A,3857.96693,N,00011.61609,W,0002.10,092.52,090812,,,A*44
```

Figura 151: Fragment de les trames capturades una vegada hem baixat del cotxe

En l'apartat referent a la velocitat proporcionada per la frase GPRMC el resultat és prou satisfactori,

tot i que, com s'aprecia en la figura 151, hi han bots entre lectura i lectura. Aquests bots es deuen a la freqüència utilitzada per a capturar les dades proporcionades per receptor, doncs al ser relativament lenta hi han canvis apreciables entre la velocitat d'una lectura i la següent. A més, també s'ha de tenir en compte el fet que, durant el trajecte, no vam estar observant el comptakilòmetres del cotxe pel que les deduccions que podem treure d'aquest paràmetre són sols estimacions aproximades. Pot ser, sí l'haguérem llegit a més velocitat hauríem pogut determinar fins quin punt és fiable aquesta informació, però a pesar d'estes carències es pot dir que el resultat obtingut és prou satisfactori, doncs al revisar les frases registrades en el trajecte representat per la figura 150, hem comprovat que les velocitats capturades encaixen dins del que s'espera pues, sí traduïm les velocitats mostrades per les frases de la figura 151, les quals corresponen a una part del trajecte a peu, a kilòmetres hora, obtenim velocitats que oscil·len entre els 3 i 4 km/h, és a dir, la velocitat a la que sol caminar algú que no té pressa per arribar a la feina.

```
$GPRMC,053511.001,A,3851.68577,N,00000.72416,W,0064.88,317.58,090812,,A*44
$GPRMC,053513.001,A,3851.71182,N,00000.75596,W,0064.74,316.09,090812,,A*48
$GPRMC,053515.001,A,3851.73744,N,00000.78821,W,0064.68,315.15,090812,,A*4F
$GPRMC,053517.001,A,3851.76230,N,00000.82126,W,0064.29,313.85,090812,,A*4F
$GPRMC,053519.001,A,3851.78569,N,00000.85335,W,0061.64,312.54,090812,,A*42
$GPRMC,053521.001,A,3851.80665,N,00000.88339,W,0056.68,311.39,090812,,A*40
$GPRMC,053523.001,A,3851.82463,N,00000.91044,W,0050.76,310.24,090812,,A*41
$GPRMC,053525.001,A,3851.84110,N,00000.93600,W,0047.74,309.56,090812,,A*4D
$GPRMC,053527.001,A,3851.85725,N,00000.96163,W,0047.17,308.52,090812,,A*49
$GPRMC,053529.001,A,3851.87331,N,00000.98816,W,0047.76,308.16,090812,,A*46
$GPRMC,053531.001,A,3851.88945,N,00001.01536,W,0048.72,307.20,090812,,A*46
$GPRMC,053533.001,A,3851.90579,N,00001.04337,W,0049.56,306.57,090812,,A*4A
$GPRMC,053535.001,A,3851.92217,N,00001.07183,W,0050.16,305.97,090812,,A*4C
$GPRMC,053537.001,A,3851.93837,N,00001.10106,W,0050.57,305.33,090812,,A*47
```

Figura 152: Fragment de les frases registrades en una zona de la N332 en la que hi ha situat un radar de tràfic.

El mateix passa sí traduïm les velocitats mostrades en la figura 152, les velocitats obtingudes encaixen amb la velocitat a la que circulaven per eixe punt, doncs en eixa zona vam circular a propòsit a uns 120 km/h per a després baixar de colp a 80-90 km/h i, d'esta manera, comprovar com havia reaccionat l'ADIS16400 a aquest canvi brusc de velocitat.

Després de veure els bons resultats de rendiment del GPS se'ns plantegen algunes idees que no se'ns van plantejar quan vam decidir incloure'l en el projecte. La més important d'aquestes idees és que la utilitat d'aquest element pot anar més enllà de la que li hem donat nosaltres, doncs a la vista de les velocitats que ha registrat, queda clar que no sols pot utilitzar-se per a localitzar un objecte, sinó que pot ajudar-nos a guanyar exactitud o corregir les mesures obtingudes per un altre sensor. Després de tot, és un dispositiu que no sols ens facilita una velocitat de desplaçament o una posició, també ens pot donar altra informació, com ara mateixa l'altitud sobre el nivell del mar, la qual més avant ens podria ser útil, pel que li hauríem de treure molt més partit del que li hem tret.

A la vista dels resultats hauríem de configurar el receptor de GPS per a adaptar-lo a la nostra aplicació. En un principi, com estaria situat en el mig de l'oceà i, per norma general, el desplaçament que podria experimentar en un parell de segons no seria gran, hi hauria prou amb configurar-lo per a que ens enviara les frases GPRMC i GPGGA en format binari a una freqüència no inferior al Hertz. Al enviar les dades en format binari, reduiríem la carrega d'ús del microcontrolador ja que aquest no deuria de perdre el temps desglossant trames i traduint els caràcters ASCII en valors que puga entendre el programa. Esta conversió podria parèixer mala idea perquè ens costaria més interpretar les lectures del receptor, però hem de tenir en compte que no hi haurà una persona al costat del sistema la major part del temps, pel que és absurd esforçar-se en fer comprensible esta informació el 100% del temps, sense contar que l'accés al sistema segurament es farà de forma remota pel que el sistema que es connecte a ell podria encarregar-se perfectament de fer eixa traducció.

Una vegada començarem a rebre dades del receptor GPS, el primer seria desglossar-les, després marcar-les temporalment i, passat un temps, hauríem de processar estes dades junt a les obtingudes amb el sensor d'inèrcia per tal de corregir estes últimes i garantir que aquestes lectures són fiables.

Tot i que aquest últim pas podria fer-nos pensar que estem intentant implementar un sistema de navegació inercial, la realitat és que la tecnologia utilitzada per a fabricar el sensor inercial que utilitzem ens obliga a fer-ho sí planegem utilitzar el sensor constantment. La raó de tenir que compensar aquestes lectures no és altra que els efectes de *Random Walk* i *Bias Stability* dels dispositius MEMS. Com sabem, aquests efectes produeixen una desviació de l'eixida del sensor que, en el cas del *Random Walk*, va augmentant amb el pas del temps. Per al sensor que hem utilitzat nosaltres tenim que:

|                                    | Giroscopi                                 | Acceleròmetre                        |
|------------------------------------|---|--------------------------------------|
| <i>Bias Temperature Coeficient</i> | $\pm 0.025^\circ/\text{s}/^\circ\text{C}$ | $\pm 1.35\text{mG}/^\circ\text{C}$   |
| <i>Angular Random Walk</i>         | $2^\circ/\sqrt{\text{hora}}$              | -                                    |
| <i>Velocity Random Walk</i>        | -   | $0.2 \text{ m/s}/\sqrt{\text{hora}}$ |

Taula 18: Especificacions dels efectes *Bias Stability* i *Random Walk* dels sensors de l'ADIS16400

Els valors d'aquestes variacions poden parèixer poc importants però al llarg del temps la desviació produïda pot ser important. Per a veure-ho més clar sols cal fer uns càlculs ràpids, així que simplement anem a suposar que utilitzem el sistema durant 24 hores i que la temperatura interna del sensor val  $18.5^\circ\text{C}$  a l'inici del registre mentre que al final del dia esta temperatura és de  $25^\circ\text{C}$ . Açò vol dir que la desviació de l'orientació mesurada per un dels giroscopis serà de:

$$\begin{aligned} \text{Variació de l'eixida produïda per la temperatura} &= \pm(0.025^\circ/\text{s})*6.5^\circ\text{C} = \pm 0,16^\circ \\ \text{Angular Random Walk Final} &= 2^\circ/\sqrt{24} = 9.8^\circ \end{aligned}$$

mentres que en un acceleròmetre tindrem:

$$\begin{aligned} \text{Variació de l'eixida produïda per la temperatura} &= \pm(1.35\text{mG}/^\circ\text{C})*6.5^\circ\text{C} = \pm 8,8\text{mG} \\ \text{Velocity Random Walk Final} &= 0.2 \text{ m/s}/\sqrt{24} = 0,98 \text{ m/s} \end{aligned}$$

Com podem veure, aquestes quantitats són ja més importants i, sí no sabem que el sistema ha estat en repòs, poden fer nos pensar que l'orientació del sistema ha canviat  $10^\circ$  des de l'inici del dia i que ens hem arribat a desplaçar a una velocitat de 3.6 Km/h. Degut a que el error de *Random Walk* és acumulatiu en el temps, és important implementar un mecanisme per a corregir-lo. Eixe mecanisme hauria d'estar basat en l'ús d'un receptor GPS, com ja hem dit, i en un algoritme per programa que periòdicament s'encarregues d'utilitzar de forma combinada les dades GPS, el valor de la temperatura interna de la IMU i els valors proporcionats pel fabricant dels paràmetres *Random Walk* i *Bias Temperature Coeficient*.

Desgraciadament, el nostre sistema no es prou potent per a implementar un procés d'aquestes característiques ja que, per a corregir les dades dels sensors, s'han d'utilitzar integrals. Així doncs, l'únic que podem fer és utilitzar el valors d'aquests paràmetres per a realitzar una correcció bàsica i deixar la part que necessita més potència de càlcul a un sistema més potent.

El no haver pogut corregir aquestes desviacions no ens ha impedit realitzar algunes proves del sistema. De fet, no és necessari realitzar cap correcció en les lectures sinó anem a donar-li un ús prolongat al sistema, pel que podem utilitzar les lectures registrades anteriorment en el trajecte casa-feina. L'únic problema de fer açò és que, per a acabar d'entendre el valor de les mesures, és molt recomanable haver estat en la zona en la que han sigut registrades. A pesar d'açò intentarem fer una

descripció bàsica per a situar al lector en el context.

Les primeres mesures a analitzar tenen lloc en l'antiga carretera nacional 332 al seu pas per Ondara, concretament en el carrer de la figura 151.



Figura 151: zona de pendent moderada de l'interior d'Ondara

Una vegada instal·lat el sistema sobre el tauler de control del cotxe vam iniciar el trajecte cap a Gandia. D'aquest trajecte ens quedarem en la part en la que circulem per l'antiga carretera nacional 332 al seu pas per Ondara, concretament en el carrer de la figura 151, el qual es caracteritza per tenir una pendent suau i per haver sigut empedrat recentment. Aquestes característiques el fan ideal per a revisar les lectures del sensor ja que és una zona amb un lleuger però constant canvi d'inclinació i en la que l'empedrat afecta més del normal l'estabilitat del cotxe, sense contar que, els desperfectes en l'empedrat obliguen a fer canvis bruscs per a sortejar-los.



Figura 152: Placa prototip instal·lada en el tauler del control del cotxe

En esta zona la IMU va registrar les lectures de la taula 20:

| Gx    | Gy    | Gz     | Ax     | Ay     | Az      |
|-------|-------|--------|--------|--------|---------|
| 0.150 | 1.950 | 30.750 | 1.5022 | 2.8084 | 9.4703  |
| 0.950 | 1.250 | 11.150 | 1.0450 | 0.1306 | 10.1234 |
| 0.000 | 0.700 | 0.100  | 1.4695 | 0.4245 | 10.0907 |
| 1.050 | 3.200 | 0.250  | 0.9144 | 0.0653 | 9.7315  |
| 0.700 | 0.050 | 2.450  | 1.2409 | 0.1306 | 9.7642  |
| 2.850 | 0.200 | 7.100  | 0.3919 | 1.0450 | 10.2867 |
| 2.650 | 4.600 | 0.950  | 0.1306 | 0.6205 | 10.0907 |
| 0.450 | 5.250 | 0.900  | 0.3266 | 0.6205 | 6.9884  |
| 0.200 | 1.650 | 0.450  | 0.1959 | 0.3919 | 8.9804  |
| 0.500 | 1.250 | 0.100  | 0.2286 | 0.0980 | 10.0907 |
| 4.200 | 0.650 | 1.600  | 1.2083 | 0.4898 | 8.2293  |
| 1.500 | 0.550 | 0.850  | 1.4042 | 0.1306 | 9.7968  |
| 1.500 | 2.050 | 0.750  | 1.4695 | 0.2939 | 8.3600  |
| 1.900 | 0.050 | 4.050  | 1.8614 | 0.4898 | 9.3397  |
| 1.600 | 3.200 | 9.150  | 1.6328 | 1.1103 | 9.3723  |
| 3.050 | 0.400 | 10.400 | 1.1430 | 2.1553 | 10.2540 |

Taula 20: Lectures registrades en la zona de la figura 152

Com podem apreciar en esta taula, tan prompte com girem a l'esquerre per a baixar pel carrer de la Pau i el carrer Sant Doménech, el giroscopi detecta el canvi d'orientació brusca de 30°/s en l'eix Z i, tan prompte com el cotxe torna a estar adreçat, torna marcar valors al voltant de zero excepte quan estem intentat esquivar els desperfectes de l'empedrat del carrer. Els restants giroscopis, com no estem realitzant cap moviment ràpid en l'eix que registren, es limiten a mostrar els valors de les vibracions resultants de circular per un carrer empedrat. Una cosa similar es pot dir dels acceleròmetres: en els eixos X i Y els acceleròmetres registren les vibracions del cotxe i en l'eix Z, tot i que no vam calibrar aquest acceleròmetre per a compensar-lo dels efectes de la gravetat, registrem els bots del cotxe al circular pel carrer.

Per a conèixer l'exactitud d'aquestes mesures hauríem de tenir detalls exactes de les diferents irregularitats del carrer, doncs d'aquesta manera saber fins quin punt els valors registrats corresponen als valors reals.

A pesar d'aquest detall tot fa pensar que, sí esta informació fos registrada a més velocitat, és a dir, prenguérem més mostres per segon dels sensors, segurament podríem utilitzar-la per a modelar la superfície del empedrat del carrer, ja que com podem veure, els acceleròmetres arriben a mesurar els petits canvis de velocitat que es produeixen entre cada bot que experimenta el cotxe al passar per damunt de l'empedrat del carrer. Açò, teòricament, també deuria implicar que els acceleròmetres haurien de ser capaços de mesurar sense problemes variacions més lentes de l'acceleració pues els valors registrats en la taula 20 són en realitat variacions brusques de la velocitat que produïxen acceleracions de poc més de 1 m/s<sup>2</sup>. Així doncs, és de suposar que, mostrejant a suficient velocitat el sensor, hauríem de ser capaços de registrar els canvis de velocitat experimentats pel sistema una vegada instal·lat en la boia i, amb les lectures obtingudes en un període de temps, calcular l'altura de la superfície de l'oceà en cada instant.

Cal dir que, amb esta informació de l'acceleració, podríem obtenir més detalls sobre quin desplaçament està experimentant la boia, pues podríem combinar els valors dels tres acceleròmetres per a formar un vector que ens indicara l'angle que està formant la boia amb la superfície del mar.

Tot i que en l'exemple que hem utilitzat possiblement no s'acabe d'apreciar la utilitat dels giroscopis, la informació proporcionada per aquestos ens pot ajudar a saber l'orientació actual de la boia. Per a tenir una idea de la precisió oferida per aquest sensors van decidir mesurar el gir d'un element del

qual ja coneixíem de forma aproximada la seua velocitat de gir: un plat de tocadiscs. Com ara veurem, els resultats van ser prou satisfactoris ja que vam obtenir velocitats que diferien de la velocitat original menys d'un 5% tal i com mostra la taula 21.

| Gx    | Gy    | Gz      | Ax     | Ay     | Az      | T      |
|-------|-------|---------|--------|--------|---------|--------|
| 0.100 | 0.050 | 0.050   | 0.0980 | 0.1633 | 9.7315  | 25.560 |
| 2.050 | 1.750 | 203.100 | 0.5225 | 0.8491 | 9.7315  | 25.420 |
| 1.900 | 1.900 | 207.100 | 0.5878 | 0.4898 | 9.7642  | 25.560 |
| 2.200 | 1.750 | 206.100 | 0.2286 | 0.7184 | 10.1234 | 25.560 |
| Gx    | Gy    | Gz      | Ax     | Ay     | Az      | T      |
| 2.300 | 2.100 | 205.300 | 0.2286 | 0.7511 | 9.7642  | 25.560 |
| 2.900 | 2.250 | 279.300 | 0.8491 | 0.9797 | 9.6989  | 25.560 |
| 2.600 | 2.150 | 277.500 | 0.4572 | 1.1430 | 9.7642  | 25.560 |
| 2.900 | 2.450 | 278.700 | 0.8817 | 1.0777 | 9.7642  | 25.700 |

Taula 21: Registre de velocitat de gir d'un plat de tocadiscs. A dalt girant a 33 RPM, baix girant a 45RPM.

La prova del tocadiscs ens proporciona informació prou útil sobre el comportament del sensor ja que, com podem veure, els acceleròmetres estan mesurant valors que no corresponen als valors obtinguts en repòs. Aquest fenomen no deuria de sorprendre'ns ja que hem sotmès el sistema a un moviment circular uniforme i, com sabem, qualsevol cos sotmès a este tipus de moviment, experimenta una acceleració perpendicular a la trajectòria del moviment. Així doncs, l'acceleració registrada pels acceleròmetres correspon a l'acceleració centrífuga, la qual, en el cas d'un objecte situat en el centre del plat del tocadiscs val aproximadament:

$$a_{33RPM} = \omega^2 * r = (198 * 2\pi / 360)^2 * 0.1 = 1.2 \text{ m/s}^2$$

$$a_{45RPM} = \omega^2 * r = (270 * 2\pi / 360)^2 * 0.1 = 2.2 \text{ m/s}^2$$

En principi, els valors teòrics no coincideixen amb els pràctics, però tampoc és de sorprendre ja que la nostra placa té unes dimensions de 12x15 cm i el plat del tocadiscs utilitzat té un radi de d'uns 20 cm, pel que les dimensions de la placa i la situació del sensor en la mateixa ens van impedir centrar-la en el mateix, pel que, al quedar el sensor a pocs centímetres de l'eix de gir, el radi és menor i, per tant, l'acceleració registrada és també menor.



Figura 153: Placa prototip sobre el plat del tocadiscs.

L'altre resultat interessant d'aquesta prova és que, quan més s'aproxima la velocitat de gir al rang dinàmic seleccionat en la IMU, major és l'exactitud de la mesura. Açò es dedueix d'observar la resposta del sensor quan aquest gira a 33 i a 45 revolucions per minut, doncs:

$$\begin{aligned} \text{Error}_{33\text{RPM}} &= 207 - (33 * 360)/60 = 207^\circ/\text{s} - 198^\circ/\text{s} = 9^\circ/\text{s} \rightarrow \text{Error}_{33\text{RPM}} = 900/198 = 4.45\% \\ \text{Error}_{45\text{RPM}} &= 278 - (45 * 360)/60 = 278^\circ/\text{s} - 270^\circ/\text{s} = 8^\circ/\text{s} \rightarrow \text{Error}_{45\text{RPM}} = 800/270 = 2.96\% \end{aligned}$$

Com demostren els càlculs, a mesura que ens arrimem al rang dinàmic seleccionat, l'error disminueix considerablement, pel que seleccionar correctament aquest rang serà imprescindible per a millorar l'exactitud de la mesura.

Cal destacar, que tot i que aquestes mesures disten entre un 3 i un 5 % del valor teòric, el resultat hauria sigut un poc més aproximat sí haguérem habilitat l'algoritme de compensació dels efectes de l'acceleració sobre els giroscopis que incorpora aquesta IMU. També s'ha de tenir en compte que els valors especificats com a velocitat de gir del plat de tocadiscs són valors teòrics. En la pràctica és d'esperar que estes velocitats siguem lleugerament majors degut a la tolerància dels components utilitzats per a fer girar el plat.

L'última cosa que ens va quedar per provar va ser el rendiment dels magnetòmetres. Per desgracia, al no poder implementar un mètode per calibrar aquests sensors, les proves que vam realitzar es van limitar a comprovar com l'eixida del sensor canviava bruscament al situar un imant prop de la IMU. És una llastima no haver aconseguit fer res més amb ells ja que hauria resultat interessant comprovar fins quin punt podrien ser útils per a mantenir l'orientació del sistema.

De totes maneres, la informació recollida per els acceleròmetres i giroscopis és més que suficient per a que, al combinar-la, obtinguem tot el necessari per a tenir una idea prou aproximada dels moviments experimentats pel sistema i quina és l'orientació del mateix pel que resultaria interessant veure fins on pot arribar un sistema amb sensors de característiques similars, però controlades per un processador de major potència.



## 12.- Conclusions i últimes paraules

Després de tot aquest temps que hem estat treballant en el sistema podem dir que hem complert els objectius marcats, doncs no sols hem aconseguir un sistema capaç de detectar canvis en la posició d'una plataforma i localitzar-la, sí no que, a més, li hem donat la possibilitat de poder registrar aquestes lectures en un dispositiu de memòria externa i de poder accedir parcialment al sistema des de l'exterior, ja siga mitjançant el teclat o des de el port USB.

Tot açò a segut possible gracies a l'estudi del funcionament del sensor inercial ADIS16400 i a l'ús d'un receptor de GPS els quals s'han utilitzat en un sistema microcontrolat el qual hem dissenyat i del que hem arribat a fabricar un prototip totalment funcional. Tot i que la potència del microcontrolador utilitat no permet utilitzar el sistema en aplicacions més complexes, sí ens permet fer més del que ens havíem proposat ja que al final el sistema també realitza les funcions de *Datalogger*, funció de la que segurament traurà un bon rendiment l'empresa que ens va cedir el sensor inercial.

A manera de resum, quan vam iniciar aquest treball de final de carrera ens vam proposar dissenyar un sistema electrònic que fos capaç de localitzar una plataforma al mateix temps que detectava i controlava l'estabilitat de la mateixa. En un principi, la idea pareixia factible doncs, després de tot, un receptor GPS és capaç de calcular la posició d'un objecte en qüestió de minuts i l'exactitud dels giroscopis i acceleròmetres actuals és més que suficient com per a complir els requisits més exigents, però a mesura que hem anat profunditzant en el projecte hem vist que la cosa no és tan simple com pensàvem.

Per a començar, vam iniciar el projecte utilitzant alguns components que ja teníem al nostre abast, concretament una placa de demostració amb un microcontrolador de 8 bits, un receptor de GPS i una pantalla LCD de 20x2 caràcters.

El primer pas va ser fer funcionar la pantalla amb aquest microcontrolador. El procediment no va ser complicat ja que en Internet hi ha prou informació sobre el controlador en el que es basa i, a més, en un dels llibres de la bibliografia hi havia rutines per a fer-lo funcionar de forma bàsica. Però a pesar d'eixa simplicitat, el procés no va ser instantani, pues vam interpretar mal la funció d'un dels terminals de control de la pantalla pel que quan vam anar escriure el nostre primer *Hola món!* vam fracassar.

Una vegada vam solucionar el problema amb la pantalla, ens vam posar a treballar en la comunicació del microcontrolador amb el receptor de GPS. Després d'uns quants dies provant configuracions de la UART del PIC i no obtenir cap classe de resultat, vam començar a preguntar-nos sí no hi hauria cap problema amb el receptor ja que les llums d'estat no actuaven com s'esperava, així que després de fer totes les proves pertinents, vam connectar un oscil·loscopi a l'eixida de dades del receptor per a veure sí podíem deduir alguna cosa a partir d'ella. Gràcies a açò descobrirem que el receptor estava com bloquejat ja que l'eixida era una tensió continua de 2 volts. Després de verificar que continuava comportant-se de la mateixa manera estant aïllat del sistema microcontrolat, certificarem el seu mal-funcionament pel que, al estar en garantia, procedirem a enviar-lo a RS per a que ens el canviara, però com no ens van explicar la seva política de devolució, no ens el van canviar i al vore que no ens arribava vam haver de contactar amb ells. Poc més d'una setmana després de l'incident tornàvem a tenir en les nostres mans un receptor de GPS. Com este sí presentava variacions en la forma d'ona de l'eixida de dades ens vam posar a treballar amb ell, però sense cap resultat satisfactori, doncs nosaltres suposàvem que tan prompte com rebia alimentació començava a enviar les 5 frases NMEA que tenia programades, però en realitat el que fa és enviar en cada posta en marxa informació bàsica sobre ell i, seguidament, comença a enviar les frases NMEA. Caure en aquest detall tampoc va ser instantani ja que, tot i que nosaltres començàvem

capturant els 200 primers caràcters capturats per la UART, quan mostràvem el *buffer* de recepció per pantalla apareixen caràcters absurds que no enteníem. Després de buscar una explicació a aquest fet decidirem augmentar el *buffer* de recepció per a veure si se'ns escapava alguna cosa. Al fer-ho vam descobrir aquest detall i que esporàdicament teníem problemes de sincronisme. Vam intentar solucionar aquest últim problema, però al veure que per molt de temps que es passara el receptor encès, mai arribava a fixar la posició, vam pensar que aquest també ens havia arribat defectuós de fàbrica pel que decidirem apartar eixa part i posar-nos a investigar sobre els sensors d'inèrcia.

L'anàlisi de les diferents tecnologies de sensat de moviment i el no poder accedir a res que no fos un sensor MEMS ens va forçar a decantar-nos per la IMU ADIS16350 d'*Analog Devices*. Lamentablement, quan ens vam decidir a demanar-la en RS, estos ja no la tenien en estoc i, en *Farnell*, l'única que hi havia disponible era l'ADIS16400, però el seu preu era elevat pel que, donada la situació econòmica del moment, vam haver de buscar un "patrocinador". Va ser la penúltima setmana de Juliol quan el vam localitzar, pel que entre unes coses i altres, el sensor ens arriba a dos dies d'Agost pel que no van tenir prou temps material per a crear una placa que poder utilitzar per a connectar-lo amb el nostre sistema de 8 bits i poder fer proves durant les vacances.

Durant el més d'Agost vam estar intentant solucionar sense cap èxit els problemes de recepció del GPS al mateix temps que anàvem documentant-nos sobre com accedir al sensor inercial ADIS164000. Com en la fulla de característiques no donava una idea clara de que era el que teníem que esperar al llegir l'eixida d'algun dels 9 sensors que el formen, vam pensar que el més lògic seria començar llegint el registre d'identificació del mateix, pel que tan prompte com vam tornar de vacances i tinguérem feta la placa de proves, ens posarem mà a l'obra i escriguérem el codi encarregat de llegir aquest registre. Per a variar, esta part no funcionava pel que abans de tocar el codi, ens vam assegurar que tot estiguera connectat correctament. Al descobrir que ens havíem equivocat identificant un dels terminals del sensor vam pensar que eixe era el motiu de l'intent fallit, però no, no va ser eixe i ens va tocar provar els diferents modes de comunicació del protocol SPI. Després de dies sense aconseguir treure res en clar, contactarem amb el suport *online* del fabricant, però com ens vam identificar com estudiants, ens facilitaren un PDF bàsic sobre la comunicació amb les IMU's del fabricant i es disculpaven per no poder atendre a universitats. Com aquest PDF no ens va proporcionar informació fresca, contactarem amb l'enginyer d'aplicacions d'*Analog Devices* i, després d'identificar-nos com a treballadors de l'empresa que ens va facilitar el sensor, ens va començar a ajudar.

Tot i que la política d'*Analog Devices* de no atendre a estudiants universitaris ens pareix molt qüestionable, cal dir que *Mark Looney*, l'enginyer que ens va atendre, va ser prou ràpid i eficient contestant les nostres preguntes. Gràcies a ell vam esbrinar que el valor mostrat en la fulla de característiques del sensor no era el correcte i confirmarem les sospites que la base dels nostres problemes en la comunicació amb el sensor es devien a una corrupció de dades. A més, també ens va confirmar que les etapes de filtrat d'aquesta IMU no estava limitat a 128 com donava a entendre la fulla de característiques, ens facilità informació més actual i ens va aclarir alguns conceptes al mateix temps que ens va fer veure més clarament els problemes d'utilitzar dispositius de tecnologia MEMS: els efectes *Random Walk* i *Bias Stability*.

Siga com siga, quan ens va confirmar que un dels valors que estàvem rebent, era el valor correcte, vam assumir que el problema de corrupció estava relacionat amb la velocitat de comunicació entre els dos dispositius, així que, com estàvem treballant un poc per dalt del valor mínim exigít per la IMU, augmentarem la freqüència de relloige del bus per provar i finalment, després de quasi dos mesos de conversacions i proves, aconseguirem comunicar-nos correctament amb el sensor.

Una vegada vam aconseguir comunicar-nos amb l'ADIS16400, el següent pas va ser mostrar per pantalla les lectures del sensor. El procés no va ser complicat, però vam haver de tirar mà d'algunes

funcions de C que havíem utilitzat molt poc, com per exemple *Itoa()* o *Modff()*. Després d'obtenir les lectures en pantalla, vam observar que estes no eren estables i no enteníem la raó. Parlant amb *Mark Looney* sobre el tema ens va suggerir que el problema podria ser de soroll, pel que, al veure que la resposta millorava al col·locar filtres de ferrita en el cablejat de connexió entre les dos plaques, vam deduir que el problema podia ser perfectament de soroll i, com la plataforma de 8 bits començava a mostrar les seues limitacions, decidirem fer bé les coses i dissenyar una placa específica pel sistema.

Esta nova placa estaria formada per un microcontrolador PIC de 16 bits, al qual li connectaríem el sensor d'inèrcia ADIS16400, el receptor de GPS, el *display* LCD de port paral·lel i, a mode d'extres, un lector de targetes Micro-SD i un port USB. El disseny d'aquesta placa no va ser excessivament complicat tot i que la vam dissenyar amb més característiques de les necessàries i amb la possibilitat de poder canviar el microcontrolador per un més potent sense tenir que alterar el disseny original.

L'únic problema vam tenir va ser produït per un descuit en la manipulació de la petjada del sensor inercial mentre manipulaven l'*Orcad Layout*. Aquest error va fer que tirarem a la brossa els tres dies que vam estar treballant en ella. Però com no hi ha mal que per bé no vinga, vam aprofitar l'ocasió per canviar les errades de disseny que havíem detectat en la mateixa. Com era d'esperar, una vegada teníem tots els components soldats, vam trobar un parell d'errors més en el disseny: el connector de programació del microcontrolador estava invertit i els forats d'ancoratge de la pantalla estaven lleugerament desplaçats. Realment no eren problemes importants, però sí un tant incòmodes, sobretot a l'hora de programar el sistema amb el *PicKit 2*, doncs els indicadors de funcionament i els controls del mateix es quedaven mirant cap a la taula de treball.

Detalls estètics a part, tant el funcionament del disseny electrònic com el de la placa ha sigut impecable pel que podem estar satisfets. L'únic canvi que s'hauria de fer en el circuit elèctric per a garantir el correcte funcionament del sistema té a veure amb el port USB. Resulta que ens vam deixar portar pel disseny suggerit per *Microchip* en la fulla de característiques del microcontrolador i no ens vam adonar que el circuit proposat per a detectar una connexió/desconnexió del bus s'adaptava pels pèls als requisits mínims de funcionament. Al provar a utilitzar el citat port i veure que no funcionava, vam caure en este detall i per sí de cas, vam modificar el valor d'una de les resistències involucrades en el circuit de detecció per tal d'assegurar-nos de complir les especificacions mínimes.

En canvi, on sí vam tenir problemes va ser en la part de programació. El no tenir una informació excessivament detallada de l'ús de les biblioteques d'accés dispositius de memòria *flash* externs, com les targetes Micro-SD, o de les llibreries USB proporcionades per *Microchip*, va fer que dedicarem quasi totes les vesprades d'un mes a fer funcionar aquestes biblioteques. És cert que, després de tant de temps intentant fer-la funcionar, t'alegres molt d'haver-ho aconseguit, però és igual de cert que fot prou esbrinar que el motiu pel qual no funciona una llibreria és una funció que no acaba de fer el que suggereix el seu nom: inicialitzar el mòdul que li dona nom.

Un altre problema que vam tenir mentre implementàvem la part *software* d'aquesta nova placa va estar relacionat amb la manipulació de variables de diferent tipus. Sorprenentment el compilador C30 no dona un missatge d'avís quan es realitzen operacions lògiques amb variables de diferents tipus, pel que mentre realitzàvem la funció de calibratge dels sensors ens vam trobar que el procés no acabava correctament perquè en la conversió d'un tipus de variable a un altre, es perdia la mitat de la informació.

El calibratge del sensor també ens va portar uns quants problemes perquè en la fulla de característiques del sensor no es donava quasi informació sobre el procés, cosa que no s'acaba d'entendre ja que els sensors MEMS necessiten tornar a ser calibrats periòdicament perquè pateixen

del que es coneix com a efecte *Random Walk*. Aquest efecte produeix una desviació en el valor d'eixida dels sensors el qual va acumulant-se en el temps i que, de no ser corregida, pot fer pensar que el sistema s'ha desplaçat del lloc quan en realitat ha estat en repòs una bona estona. A més, les lectures d'aquests sensors també es veuen afectades per la temperatura, pel que al igual que hi ha que compensar periòdicament els efectes del *Random Walk* sobre l'eixida dels sensors, també hi ha que fer el mateix amb els efectes de la temperatura. Per aquest motiu és importantíssim repetir el calibratge del dispositiu cada cert temps. El problema d'açò, és que per a donar per bo el calibratge, és necessari que el sistema estiga totalment en repòs, ja que de no estar-ho, les lectures obtingudes després del calibratge no seran mai zero, inclús quan el sistema estiga en repòs.

Siga com siga el principal problema que ens vam trobar a l'hora d'implementar l'algoritme de calibratge va ser la quantitat de RAM disponible per a implementar-lo. Açò es deu a que per a realitzar-lo s'han de prendre 820 mostres per segon durant un temps que va dels 30 segons per al cas dels acceleròmetres als 100 que necessita un giroscopi. Com el PIC utilitzat sols disposa de 16KB de RAM sols tenim lloc per a uns 8 segons de mostres. Esta quantitat és clarament insuficient pel que és necessari o bé utilitzar la funció *malloc()* per a assignar dinàmicament la memòria que ens fa falta, o bé establir un *buffer* fixe en la RAM i quedar-nos sense una gran part de la RAM disponible. Nosaltres vam optar per utilitzar la primera opció però, esporàdicament, el sistema es reiniciava soles pel que per a no perdre el temps esbrinant quin era el problema, vam optar per utilitzar un *buffer* fixe.

L'elaboració de l'algoritme de programació no va ser excessivament complicat, però ens va obligar a demanar la participació de l'usuari ja que teòricament, quan es calibren els acceleròmetres, la placa ha de ser inclinada en cada eix de manera que experimente en eixe eix una força de +1G i de -1G. En la pràctica, seguir al peu de la lletra aquest procés no és imprescindible perquè aquest procés serveix per a establir la posició en la qual els sensors donaran una resposta nul·la. Això vol dir que, sí deixem el sistema damunt de una taula i ens limitem a confirmar tots els passos que se'ns mostra per pantalla, al acabar, les lectures dels sensors oscil·laran al voltant de zero, quan en realitat la lectura de l'acceleròmetre de l'eix Z hauria d'estar mesurant la força de la gravetat.

Durant el procés de depuració de la funció de calibratge, ens vam adonar que la duració del procés era prou menor del que es suposava. Buscant la possible causa d'aquest fenomen ens vam trobar amb que no es devia a un sol motiu, sí no a dos: una inicialització incorrecta dels temporitzadors del PIC i el no haver tingut en compte que la llargària del polse *Data Ready* de la IMU era major que el que durava el procés de lectura, pel que al no esperar a que aquest tornara al seu valor de repòs, quan tornàvem a accedir a l'ADIS16400, aquest continuava a nivell alt pel que tornàvem a llegir les mateixes dades que en la lectura anterior, o siga, que realitzàvem una doble lectura.

Cal destacar que l'algoritme de calibratge implementat no s'encarrega de calibrar els magnetòmetres. Açò es deu a que, per a realitzar aquest procés, és necessari moure el sistema de manera que dibuixe una circumferència sí anem a fer un calibratge bidimensional o bé una esfera sí el calibratge és tridimensional. Degut a les interferències magnètiques, al representar les lectures respecte els punts en els que han sigut obtingues, no obtindrem una circumferència o una esfera centrada en l'origen, sinó que, depenent del cas, obtindrem una el·lipse o el·lipsoide desplaçada de l'origen. Al no disposar d'un mètode fiable per a realitzar el procés i, al necessitar molta potència de càlcul per a realitzar-lo, vam tenir que deixar esta funció de costat, ja que inclús seguint el model utilitzat pels mòbils actuals de dibuixar una figura en forma de 8 ens hauria sigut excessivament costós. És una llastima perquè hauria resultat interessant veure fins quin punt era útil utilitzar els magnetòmetres per a corregir els errors d'orientació experimentats pels giroscopis a mesura que augmenta l'altura, però amb les característiques actuals del sistema és impossible fer-ho.

Al acabar l'algoritme encarregat de realitzar el calibratge i veure que el resultat obtingut era l'esperat, vam començar a documentar-nos sobre els sistemes de navegació inercial i sobre com mesurar les ones de l'oceà.

Durant l'etapa de documentació ens vam adonar que les característiques tècniques del microcontrolador elegit eren excessivament modestes per a realitzar qualsevol d'aquestes funcions pues, en el cas de la navegació inercial, es fa necessari integrar les lectures dels sensors i resoldre equacions diferencials per a obtenir informació que puga ser útil per a calcular la posició actual a partir de la distància recorreguda i de l'última posició vàlida coneguda. El cas de mesurar ones no és molt més simple ja que, donada la natura aleatòria de la superfície del mar, per a poder analitzar l'estat del mar a partir de les ones el que es fa és digitalitzar els canvis d'altura de la superfície del mar i, finalment, mitjançant la transformada ràpida de *Fourier*, calcular la Densitat Espectral de Potència per a, d'esta manera, obtenir l'energia del mar en eixe punt. A més, com les ones poden tenir períodes molt llargs, es fa necessari estar emmagatzemant dades d'altura durant almenys 3 hores a freqüències de mostreig baixes.

Aquests detalls ens van fer veure que el microcontrolador anava a limitar el nostre sistema a la simple recollida de dades, pel que per a veure sí seria factible utilitzar-lo per a mesurar ones ens vam posar en contacte amb l'empresa de boies que ens va subministrar el sensor per tal que ens donara detalls més específics sobre els paràmetres més importants de les ones.

Mentres esperàvem resposta per part de l'empresa ens vam tornar a posar a treballar una altra vegada amb el receptor de GPS. Per a la nostra sorpresa, poc després de reiniciar el treball, va aconseguir fixar la nostra posició, pel que decidirem fer més robusta front a errors la funció encarregada de processar el *buffer* de la UART, però a pesar dels esforços realitzats, la recepció de caràcters va continuar sent problemàtica i, moltes vegades, impedia el correcte funcionament del sistema en mode *datalogger*. La causa d'aquest efecte és el procediment que hem utilitzat per mostrar i capturar les dades dels sensors. Al no disposar de prou línies en pantalla i, com el receptor ens envia frases a cada segon, ens les hem hagut d'arreglar per a mostrar les 8 línies necessàries en 2, pel que en el moment en el que en el *buffer* no conté la frase GPRMC o esta està incompleta o corrupta, la rutina encarregada de processar el *buffer* es queda esperant a l'arribada d'una frase vàlida i, fins que no l'obté, no torna a cedir el control a la funció encarregada de llegir el sensor, frenant d'aquesta manera el procés de registre de lectures.

Solucionar aquest problema no hauria de ser complicat, desgraciadament ens vam donar compte d'aquest efecte una vegada havíem donat per tancat el projecte i no ens va donar temps a solucionar-ho. A la vista d'aquest problema ens queda el dubte de saber sí hauria solucionat el problema el haver pogut configurar el GPS per a enviar-nos una única frase NMEA a una freqüència major de 1Hz, doncs tot i que en la fulla de característiques no proporcionava la informació necessària per a configurar el receptor, nosaltres, mentre fèiem proves amb el microcontrolador de 8 bits, vam obtenir el fabricant i model real del receptor i inclús vam localitzar una aplicació de *Windows* que teòricament servia per a, entre altres coses, configurar el receptor, però com per a connectar el receptor al PC calia crear un circuit que adaptara els 3.3 Volts de l'eixida del receptor als 12 volts que utilitza el port serie d'un ordinador, ho vam deixar per al final sí ens sobrava temps.

Òbviament tot el temps que vam estar esperant una resposta per part de l'empresa no el vam dedicar a perfeccionar la funció encarregada de processar el *buffer* de recepció de la UART, en eixe temps, com ja sabíem que el nostre disseny no anava a servir per a un projecte tan ambiciós com el que ens havíem plantejat, vam crear unes rutines encarregades de processar l'entrada del teclat de manera que, al ser utilitzat junt a un sistema de menús en pantalla, l'usuari poguera escollir alguns dels paràmetres del sistema. La implementació d'aquesta part va ser un poc enrevessada perquè ni havíem pensat afegir esta funció al sistema ni mai havíem fet una cosa pareguda, però després d'uns

dies de dur treball vam aconseguir fer-la funcionar pel que la vam deixar en el projecte a l'espera d'alguna notícia per part de l'empresa externa.

A la vista dels resultats experimentats del sistema, podem dir que la precisió i exactitud que ens donen el receptor GPS i el sensor inercial és més que suficient per implementar un sistema que s'encarregue de registrar els moviments d'una plataforma, doncs com hem vist és capaç de detectar amb prou exactitud canvis d'orientació ràpids i canvis d'acceleració relativament petits. Per a acabar de treure-li més rendiment al sistema, ens hauríem d'oblidar de mostrar dades per pantalla periòdicament i veure d'utilitzar un altre receptor de GPS, o bé configurar el nostre al mateix temps que fem un poc més robusta la funció encarregada de processar el *buffer* de recepció. D'aquesta manera compliríem un dels objectius que ens vam marcar. Per a acabar de complir-los hauríem d'utilitzar un processador més potent o bé un DSP sí ens interessara controlar la superfície del mar.

Tot i que al tancar la memòria no havíem rebut els detalls sol·licitats a l'empresa, tot fa pensar que seria factible utilitzar l'ADIS16400 per a detectar i registrar l'estat de l'oceà.

Independentment de l'ús que se li vaja a donar el sistema, hem de tenir en compte que, en cas d'utilitzar-lo durant períodes de funcionament llargs, s'ha d'implementar un mecanisme que permeti al sistema compensar els efectes del *Random Walk* i *Bias Stability* típics dels sensors de tipus MEMS. Una vegada implementat aquest mecanisme podem estar segurs que el sensor i el sistema respondran com s'espera d'ells. Al cap i a la fi, la combinació d'un receptor de GPS i d'un sensor d'inèrcia ens pot portar tan lluny com la potència del processador utilitzat ens permeti.

## 13- Bibliografia

Aquesta secció conté la bibliografia de més rellevància utilitzada per a la realització d'aquest treball de final de carrera. Tot i que alguns d'aquests document han sigut utilitzats en més d'una secció del document, sols apareixen anomenats en la secció en la que han tingut un paper més rellevant.

### 13.1- Localització d'un objecte mitjançant satèl·lits

-*Còmo funciona el GPS en 5 pasos logicos*

<http://www.elgps.com/documentos/comofuncionagps/comofuncionagps.html>

-*GPS Navigation Satellite message format and protocol details*

<http://gpsinformation.net/gpssignal.htm>

-*Sistema NAVSTAR-GPS*

<http://tel.abloque.com/telecomunicaciones/tutorial/contenido.html>

-*Global Positioning System*

[http://en.wikipedia.org/wiki/Global\\_Positioning\\_System](http://en.wikipedia.org/wiki/Global_Positioning_System)

-*GLONASS*

<http://en.wikipedia.org/wiki/GLONASS>

-*GLONASS Summary*

[http://www.spaceandtech.com/spacedata/constellations/glonass\\_consum.shtml](http://www.spaceandtech.com/spacedata/constellations/glonass_consum.shtml)

-*GLONASS Technical Description*

[http://www.nis-glonass.ru/en/glonass/technical\\_descript/index.php](http://www.nis-glonass.ru/en/glonass/technical_descript/index.php)

-*Galileo Satellite Navigation*

[http://en.wikipedia.org/wiki/Galileo\\_%28satellite\\_navigation%29](http://en.wikipedia.org/wiki/Galileo_%28satellite_navigation%29)

-*Galileo - European Satellite Navigation System*

<http://www.aatl.net/publications/galileo.htm>

-*Galileo - European Satellite Navigation System*

<http://www.navnin.nl/NIN/Downloads/Galileo%20-%20European%20SNS.pdf>

-*Galileo Navigation Satellite System*

<http://www.globalsecurity.org/space/world/europe/nav.htm>

-*National Marine Electronics Association, NMEA 0183 v. 2.00 1/1/1992*

<http://www.ik4hdq.net/doc/testi/nmeaprotocollo.pdf>

-*GPS Reciever GPS-610F*

<http://www.rfsolutions.co.uk/acatalog/DS-GPS610F-2.pdf>

## 13.2- Sensors de moviment

-Accelerometer

<http://en.wikipedia.org/wiki/Accelerometer>

-A Beginners Guide to Accelerometers

<http://www.dimensionengineering.com/info/accelerometers>

-Introduction to Accelerometers

<http://www.omega.com/prodinfo/accelerometers.html>

-Accelerometer Principles

<http://zone.ni.com/devzone/cda/ph/p/id/12>

-Lecture #8, Capacitive Accelerometers

[http://www.stanford.edu/class/me220/data/lectures/lect08/lect\\_4.html](http://www.stanford.edu/class/me220/data/lectures/lect08/lect_4.html)

-Introduction to Piezoelectric Accelerometers

<http://www.thermofisher.com.au/Uploads/file/Environmental-Industrial/Process-Monitoring-Industrial-Instruments/Sound-Vibration-Stress-Monitoring/Precision-Accelerometers/PCB-Vib-Modal/pdf/Introduction-to-Piezoelectric-Accelerometers.pdf>

-Inclinometer

<http://en.wikipedia.org/wiki/Inclinometer>

-Gyroscope

<http://en.wikipedia.org/wiki/Gyroscope>

-Gyroscope

<http://sensorwiki.org/doku.php/sensors/gyroscope>

-El Efecto Sagnac y sus Consecuencias

<http://www.relatividad.org/bhole/sagnac.html>

-Fiber Optic Gyroscope

[http://en.wikipedia.org/wiki/Fibre\\_optic\\_gyroscope](http://en.wikipedia.org/wiki/Fibre_optic_gyroscope)

-Fiber Optic Gyroscope

[http://www.es.northropgrumman.com/by\\_division/navigationssystem/whitepapers/assets/Fiber-Optic\\_Gyros.pdf](http://www.es.northropgrumman.com/by_division/navigationssystem/whitepapers/assets/Fiber-Optic_Gyros.pdf)

-Vibrating Structure Gyroscope

[http://en.wikipedia.org/wiki/Vibrating\\_structure\\_gyroscope](http://en.wikipedia.org/wiki/Vibrating_structure_gyroscope)

-Uses for Gyroscopes

<http://www.gyroscopes.org/uses.asp>

-An introduction to inertial navigation. Universitat de Cambridge, ISSN 1476-2986

<http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>

-Inertial Sensors

[http://www.artechhouse.com/uploads/public/documents/chapters/Groves\\_255\\_CH4.pdf](http://www.artechhouse.com/uploads/public/documents/chapters/Groves_255_CH4.pdf)



*-Magnetometer, The History*

<http://www.ctsystems.eu/gauss.htm>

*-Magnetometer:*

<http://en.wikipedia.org/wiki/Magnetometer>

*-Magnetometers:*

[http://www.lund.irf.se/helioshome/geomag\\_magnetometers.html](http://www.lund.irf.se/helioshome/geomag_magnetometers.html)

*-Hall Effect Sensor*

[http://en.wikipedia.org/wiki/Hall\\_effect\\_sensor](http://en.wikipedia.org/wiki/Hall_effect_sensor)

*-Semiconductor-based Sensors*

<http://es.scribd.com/doc/58302595/54/Magnetodiode>

*-Fluxgate Magnetometer*

<http://www.earthsci.unimelb.edu.au/ES304/MODULES/MAG/NOTES/fluxgate.html>

*-Magnetic Field Measurement*

<http://dsp-book.narod.ru/MISH/CH48.PDF>

*-SQUID*

<http://en.wikipedia.org/wiki/SQUID>

*-Overview of MEMS Gyroscopes*

<http://www.alexandertrusov.com/uploads/pdf/2011-UCI-trusov-whitepaper-gyros.pdf>

*-iMEMS Angular Rate Sensing Gyroscope*

<http://www.analog.com/library/analogdialogue/archives/37-03/gyro.pdf>

*-MEMS Magnetic Field Sensor*

[http://en.wikipedia.org/wiki/MEMS\\_magnetometer](http://en.wikipedia.org/wiki/MEMS_magnetometer)

*-Inclinometers with MEMS Sensors*

<http://can-newsletter.org/assets/files/ttmedia/raw/0cf5ab33dfcb995d99894f7b9c7e560f.pdf>

*-Quasistatic Young's Modulus Values as a Function of Temperature*

<http://twm.mpei.ac.ru/tthb/1/Tab-5-5-8-2-3-Young-Modul.html>

*-Bias Stability Measurement: Allan Variance*

[http://www.xbow.com/pdf/Bias\\_Stability\\_Measurement.pdf](http://www.xbow.com/pdf/Bias_Stability_Measurement.pdf)

### **13.3- Sistemas de Navegació Inercial**

*-Guidance Systems*

[http://en.wikipedia.org/wiki/Guidance\\_system](http://en.wikipedia.org/wiki/Guidance_system)

*-Inertial Navigation System*

[http://en.wikipedia.org/wiki/Inertial\\_navigation\\_system](http://en.wikipedia.org/wiki/Inertial_navigation_system)

*-Inertial Navigation System*

<http://accessscience.com/content.aspx?id=342700>

*-Basic Inertial Navigation*

<http://www.fas.org/spp/military/program/nav/basicnav.pdf>

### **13.4- Exemple d'aplicació**

*-Energia de las olas*

<http://libros.redsauce.net/EnergiasAlternativas/mar/PDFs/01Olas.pdf>

*-Introduction to Physical Oceanography*

[http://oceanworld.tamu.edu/resources/ocng\\_textbook/contents.html](http://oceanworld.tamu.edu/resources/ocng_textbook/contents.html)

*-An Invitation to Marine Science (4th Edition, Chapter 10)*

<http://www4.ncsu.edu/eos/users/c/ceknowle/public/chapter10/index.html#Discussion>

*-The Coastal Data Information Program, Wave Measurement*

<http://cdip.ucsd.edu/?nav=documents&sub=index&xitem=waves>

*-Coastal Engineering Book, part II: Coastal Hydrodynamics, Chapter 1: Water Wave Dynamics*

<http://chl.erdc.usace.army.mil/chl.aspx?p=s&a=ARTICLES;101>

### **13.5- Descripció general dels components escollits**

<http://www.microchip.com>

*-PIC24FJ256GB110 Family DataSheet*

*-PIC24FJ256GB110 Family Silicon/Data Sheet Errata*

*-dsPIC33EPXXXMU806/810/814 Data Sheet*

*-PIC18F46J50 Family Data Sheet*

*-GPS 610F Datasheet*

<http://www.rfolutions.co.uk/acatalog/DS-GPS610F-2.pdf>

*-Application Note AN0003: Binary Messages of SkyTraq Venus 6 GPS Receiver*

[http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus/638/doc/AN0003\\_v1.4.19.pdf](http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus/638/doc/AN0003_v1.4.19.pdf)

*-Fordata FDC2002C LCD Display Data Sheet*

<http://docs-europe.electrocomponents.com/webdocs/0f25/0900766b80f25e58.pdf>

*-SkyTraq GPS Software*

[http://www.onetalent-gnss.com/ideas/usb-hw-receivers/yuan10/Skytraq\\_GPS\\_Software.zip](http://www.onetalent-gnss.com/ideas/usb-hw-receivers/yuan10/Skytraq_GPS_Software.zip)

*-ADIS16400 Tri-Axial Inertial Sensor with Magnetometer Datasheet Rev. B*

[http://www.analog.com/static/imported-files/data\\_sheets/ADIS16400\\_16405.pdf](http://www.analog.com/static/imported-files/data_sheets/ADIS16400_16405.pdf)

*-ADIS16400/16405 Silicon Anomaly*

[http://www.analog.com/static/imported-files/data\\_sheets/ADIS16400\\_16405\\_Anomaly.pdf](http://www.analog.com/static/imported-files/data_sheets/ADIS16400_16405_Anomaly.pdf)

-ADIS16407 Ten Degrees of Freedom Inertial Sensor

[http://www.analog.com/static/imported-files/data\\_sheets/ADIS16407.pdf](http://www.analog.com/static/imported-files/data_sheets/ADIS16407.pdf)

-TFC: Diseño y construcción de bus de datos y sensores para las prácticas de NACC

<http://upcommons.upc.edu/pfc/bitstream/2099.1/4679/1/Mem%C3%B2ria.pdf>

-Compensating for Tilt, Hard-Iron, and Soft-Iron Effects

<http://www.sensormag.com/sensors/motion-velocity-displacement/compensating-tilt-hard-iron-and-soft-iron-effects-6475>

-Technical Document: Compensating for Tilt, Hard-Iron, and Soft-Iron Effects

[http://www.memsense.com/docs/MTD-0802\\_1.2\\_Magnetometer\\_Calibration.pdf](http://www.memsense.com/docs/MTD-0802_1.2_Magnetometer_Calibration.pdf)

-Hard and soft iron magnetic compensation explained

<http://blogs.freescale.com/2011/03/14/hard-and-soft-iron-magnetic-compensation-explained/>

-Magnetometer Soft- and Hard-Iron Calibration

<http://diydrones.com/forum/topics/magnetometer-soft-and-hard-iron-calibration>

-The Magic Figure 8

<http://hillcrestlabs.com/makingmotion/2012/06/20/the-magic-figure-eight/>

### **13.6- Esquema elèctric i distribució dels components del sistema**

-Diseño de circuitos impresos con OrCAD Capture y Layout v. 9.2 de M<sup>a</sup> Auxilio Recasens Bellver y José González Calabuig. Ed. Thomson (Paraninfo). ISBN 84-9732-071-9.-

-LP3871/LP3874 0.8A Fast Ultra Low Dropout Lineal Regulagors Datasheet

<http://www.ti.com/lit/ds/symlink/lp3874.pdf>

-microSD Card Pinout

[http://www.interfacebus.com/MicroSD\\_Card\\_Pinout.html](http://www.interfacebus.com/MicroSD_Card_Pinout.html)

-microSD Atmega32 Data-Logger

<http://www.dharmanitech.com/2011/05/microsd-atmega32-datalogger.html>

-SanDisk SD Card Product Manual V2.2

<http://teslabs.com/openplayer/docs/docs/specs/ProductManualSDCardv2.2final.pdf>

### **13.7- Firmware**

-Learning to Fly the PIC: Programming 16 Bit Microcontrollers in C. de Lucio Di Jasio Editorial Newnes. ISBN-13: 978-0-7506-8292-3

-El lenguaje de programación C de Brian W. Kernighan y Dennis M. Ritchie. Editorial Prentice Hall. ISBN 968-880-205-0

<http://www.microchip.com>

-PIC24F Family Reference Manual

-MPLAB C Compiler for PIC24 MCUs and DSPIC DSCs User's Guide

-16 Bit Language Tools Getting started  
-16 Bit Language Tools Libraries  
-Microchip Application Libraries & Release Notes  
-Microchip USB Device Firmware Framework User's Guide  
-Application Note AN1003: USB Mass Storage Device Using a PIC MCU-Application Note AN1045: Implementing File I/O Functions Using Microchip's Memory Disk Drive File System Library  
-Application Note AN1142: USB Mass Storage Class on a Embedded Host  
-Application Note AN1189: Implementing a Mass Storage Device Using the Microchip USB Device Firmware Framework

-ST7066U Dot Matrix LCD Controller/Driver  
<http://docs-europe.electrocomponents.com/webdocs/0f25/0900766b80f25db8.pdf>

-Binary Messages of SkyTraq Venus 6 GPS Receiver  
[http://www.sparkfun.com/datasheets/GPS/Modules/AN0003\\_v1.4.8.pdf](http://www.sparkfun.com/datasheets/GPS/Modules/AN0003_v1.4.8.pdf)

-ADIS16400 Tri-Axial Inertial Sensor with Magnetometer Datasheet Rev. B  
[http://www.analog.com/static/imported-files/data\\_sheets/ADIS16400\\_16405.pdf](http://www.analog.com/static/imported-files/data_sheets/ADIS16400_16405.pdf)

-ADIS16407 Ten Degrees of Freedom Inertial Sensor  
[http://www.analog.com/static/imported-files/data\\_sheets/ADIS16407.pdf](http://www.analog.com/static/imported-files/data_sheets/ADIS16407.pdf)

-Analog Devices Application Note AN1041  
[http://www.analog.com/static/imported-files/application\\_notes/AN-1041.pdf](http://www.analog.com/static/imported-files/application_notes/AN-1041.pdf)

-Analog Devices Application Note AN1045  
[http://www.analog.com/static/imported-files/application\\_notes/AN-1045.pdf](http://www.analog.com/static/imported-files/application_notes/AN-1045.pdf)

-Sprintf in C18 (FTOA)  
<http://www.microchip.com/forums/m55402.aspx>

### **13.8- Implementació pràctica**

-Tiertex TDSDB146J50 Demo Board Datasheet  
[http://www.tiertex.com/pub/files/Manuals/1313065955\\_TDSDB146J50.pdf](http://www.tiertex.com/pub/files/Manuals/1313065955_TDSDB146J50.pdf)

-Explorer 16 Development Board User's Guide  
<http://ww1.microchip.com/downloads/en/DeviceDoc/Explorer%2016%20User%20Guide%2051589a.pdf>

-MPLAB Starter Kit for PIC24F  
[http://ww1.microchip.com/downloads/en/DeviceDoc/PIC24F%20Starter\\_UG\\_DS-51725a.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/PIC24F%20Starter_UG_DS-51725a.pdf)

### **13.9- Prova experimental del sistema**

-GPS Visualizer  
<http://www.gpsvisualizer.com/>

-Movimiento circular  
<http://www.profesorenlinea.cl/fisica/MovimientoCircular.html>

## Apèndixs

### Apèndix 1: Matisacions i errades en la fulles de característiques

#### ADIS 16400/16405 *Tri-Axis Inertial Sensor with Magnetometer*

La versió de la fulla de característiques del ADIS16400 que hem utilitzat per a desenvolupar aquest projecte de final de carrera és la més recent disponible en la web del fabricant: la B. A pesar de ser la més actual, va ser publicada en l'estiu del 2009 i ni és perfecta ni està lliure d'errors.

Com hem dit en la descripció d'aquest sensor d'inèrcia, la fulla de característiques és prou simple i s'ometen alguns detalls que és interessant saber. La majoria d'aquests han sigut exposats en aquesta memòria, però n'hi han alguns que es fan apreciables quan s'utilitza aquesta fulla de característiques junt a la del ADIS16407 i la nota d'aplicació AN-1041. Per això es recomanable utilitzar la fulla de característiques del sensor que estiguem utilitzant junt a la fulla de característiques d'una altra IMU més avançada i/o moderna, pues pel que es pot veure, *Analog Devices* no cuida molt les seues fulles de característiques i les millores i correccions les apliquen directament al model més nou.

De tots els errors, el que més ens va afectar va ser el del valor del registre d'identificació. El valor mostrat en la taula 8 de la pàgina 11 sols és vàlid en cas d'utilitzar l'ADIS16405, pel que sí estem utilitzant un ADIS16400, el valor que hem d'esperar és 0x4100.

Table 8. User Register Memory Map

| Name       | R/W | Flash Backup | Address <sup>1</sup> | Default | Function                                 |
|------------|-----|--------------|----------------------|---------|--|
| FLASH_CNT  | R   | Yes          | 0x00                 | N/A     | Flash memory write count                 |
| SUPPLY_OUT | R   | No           | 0x02                 | N/A     | Power supply measurement                 |
| ...        | ... | ...          | ...                  | ...     | ...                                      |
| GPIO_CTRL  | R/W | No           | 0x32                 | 0x0000  | Auxiliary digital input/output control   |
| MSC_CTRL   | R/W | Yes          | 0x34                 | 0x0006  | Miscellaneous control                    |
| SMPL_PRD   | R/W | Yes          | 0x36                 | 0x0001  | Internal sample period (rate) control    |
| SENS_AVG   | R/W | Yes          | 0x38                 | 0x0402  | Dynamic range and digital filter control |
| SLP_CNT    | W   | No           | 0x3A                 | 0x0000  | Sleep mode control                       |
| DIAG_STAT  | R   | No           | 0x3C                 | 0x0000  | System status                            |
| GLOB_CMD   | W   | N/A          | 0x3E                 | 0x0000  | System command                           |
| ALM_MAG1   | R/W | Yes          | 0x40                 | 0x0000  | Alarm 1 amplitude threshold              |
| ALM_MAG2   | R/W | Yes          | 0x42                 | 0x0000  | Alarm 2 amplitude threshold              |
| ALM_SMPL1  | R/W | Yes          | 0x44                 | 0x0000  | Alarm 1 sample size                      |
| ALM_SMPL2  | R/W | Yes          | 0x46                 | 0x0000  | Alarm 2 sample size                      |
| ALM_CTRL   | R/W | Yes          | 0x48                 | 0x0000  | Alarm control                            |
| AUX_DAC    | R/W | No           | 0x4A                 | 0x0000  | Auxiliary DAC data                       |
|            |     |              | 0x4C to 0x55         |         | Reserved                                 |
| PRODUCT_ID |     |              | 0x56                 | 0x4105  | Product identifier                       |

Figura 151 : Detall del error en la taula del mapa de memòria.

Una altra de les coses que ens hem trobat en la fulla de característiques durant la realització del PFC és l'explicació de la configuració del filtrat de la pàgina 14, ja no per ser inexistent i tenir que utilitzar la nota d'aplicació AN-1041 per a treure alguna cosa de trelat, sinó perquè en cap moment específica que el nombre màxim d'etapes del filtre *Bartlett* és  $2^6$  i no  $2^7$  com fa pensar el fet de disposar de 3 bits per a configurar la quantitat d'etapes que utilitzarà el filtre. Destacar que aquest detall no està corregit/especificat en cap de les fulles de característiques de les IMU's d'*Analog Devices* pel que s'ha de tenir en compte en cas d'utilitzar aquesta característica.

## GPS-610F GPS Receiver with Ext Antenna

El receptor GPS-610F és l'exemple perfecte dels inconvenients que té comprar un dispositiu barat: el suport tècnic. No és que la fulla de característiques no ens permeta utilitzar el receptor de GPS, però sense dubte no aclareix tot el necessari per a aprofitar decentment el receptor.

Des del nostre punt de vista, la fulla de característiques estaria completa sí especificara aquests detalls:

-El connector d'antena externa es pot utilitzar tant amb antenes passives com actives. Sí s'utilitza una antena activa, aleshores s'haurà d'alimentar amb 5 volts el terminal 10 del connector del receptor.

-Hauria de donar més detalls sobre l'entrada de dades diferencial DGPS i recordar que, sí no va a utilitzar-se, aleshores hauria de connectar-se a l'alimentació del receptor (terminal 3).

-Més que dir que alimentar la bateria de back-up és opcional, hauria de dir que és molt recomanable, ja que segurament, el receptor va tardar vora un any en localitzar-nos per no haver-se tornat a utilitzar des que va ser fabricat.

A més, també haurien de corregir, o almenys avisar, que la longitud dels camps de les frases pot variar depenent de la versió del *firmware* intern del receptor. En el cas del nostre receptor hi havien dos camps de la frase GPRMC que no coincidien de cap de manera amb el que s'indicava la fulla de característiques: velocitat i rumb sobre la superfície. Segons la fulla, aquests camps ocupen 3 caràcters cadascun, però els registres de les captures mostren que ocupen 7 i 5 caràcters respectivament.

```
$GPRMC,hhmmss. sss,A,dddmm. mmmm,a,dddmm. mmmm,a,x.x,x.x,ddmmyy,x.x
,a,a*hh<CR><LF>
```

|         |            |   |            |   |             |   |         |        |        |  |  |   |    |
|---------|------------|---|------------|---|-------------|---|---------|--------|--------|--|--|---|----|
| \$GPRMC | 053347.001 | A | 3850.42320 | N | 00000.06721 | W | 0052.14 | 347.00 | 090812 |  |  | A | 40 |
| \$GPRMC | 053349.001 | A | 3850.45133 | N | 00000.07586 | W | 0051.95 | 346.59 | 090812 |  |  | A | 40 |
| \$GPRMC | 053351.001 | A | 3850.47951 | N | 00000.08441 | W | 0052.19 | 346.76 | 090812 |  |  | A | 48 |
| \$GPRMC | 053353.001 | A | 3850.50800 | N | 00000.09294 | W | 0052.96 | 346.70 | 090812 |  |  | A | 47 |
| \$GPRMC | 053355.001 | A | 3850.53687 | N | 00000.10165 | W | 0053.82 | 346.76 | 090812 |  |  | A | 44 |
| \$GPRMC | 053357.001 | A | 3850.56642 | N | 00000.11067 | W | 0054.83 | 346.72 | 090812 |  |  | A | 4A |
| \$GPRMC | 053359.001 | A | 3850.59658 | N | 00000.11973 | W | 0055.82 | 346.86 | 090812 |  |  | A | 47 |
| \$GPRMC | 053401.001 | A | 3850.62724 | N | 00000.12883 | W | 0056.66 | 346.75 | 090812 |  |  | A | 47 |
| \$GPRMC | 053403.001 | A | 3850.65823 | N | 00000.13825 | W | 0057.31 | 346.62 | 090812 |  |  | A | 42 |

Figura 152 : Comparativa entre l'estructura de la frase GPRMC especificada per la fulla de característiques i la registrada.

## **Apèndix 2: Conversacions amb personal d'Analog Devices**

**3 d'octubre del 2011**

Contacte amb el suport tècnic web de *Analog Devices*

Full Name: Josep Gil Mora  
Company Name: Escola Politècnica Superior de Gandia  
Country: SPAIN  
Telephone: [+34629720\\*\\*\\*](tel:+34629720***)  
Product Category: MEMS and Sensors  
Part number, if relevant: ADIS16400  
Volume: 1-999

Please write your questions or comments here: Hi, i'm trying to read an ADIS16400 with a pic18f46j50's SPI hardware module with no luck. I guess that the problem is that this pic spi module is only 8 bits so between each read/write to the sensor, SCLK stays high more time than it should in a regular 16bit transmission because the pic leaves it inactive (high) while it's preparing the second part of the transmission and that makes the adis lose synchronization, isn't it? thanks in advance :) Josep p.d i don't know if i'm explained my self properly so here's a pic wich should help:  
<http://i25.photobucket.com/albums/c91/piteta/altres/capturasclock.jpg>

Resposta per part de *Mike Heffernan* el 4 d'Octubre del 2011:

Hello Josep,

I regret we have very limited resources available to support Universities but in this case I think the attached presentation on getting started with SPI for iSensors should solve your problem.

16 SCLKs per data cycle are required for ADiS devices. When using 8-bit processors, use two 8-bit cycles and keep the chip select line low in between them. See the attached presentation for details and timing diagrams etc.

Best Regards,  
Mike

Mike Heffernan  
Applications Engineer  
European Centralized Application Center (CAC)  
Analog Devices

---

**19 d'octubre del 2011**

Email a *Mark Looney*, enginyer d'aplicacions de sensors *Analog Devices*:

**From:** Josep Gil [mailto:[piteta1@gmail.com](mailto:piteta1@gmail.com)]  
**Sent:** Wednesday, October 19, 2011 1:18 PM  
**To:** Looney, Mark  
**Subject:** Questions about using ADIS 16400

Hi, my name is Josep, i'm from Spain and recently we bought an ADIS16400 from Farnell for testing purposes. In order to read it we've connected it to a 8 bit pic microcontroller (PIC18F46J50). This microcontroller is 44 pin device wich runs (internally) at 48MHz, has 2 8bit SPI ports (one of them with DMA capabilities), uart, usb, several timers etc and i guess it should be enough to read the sensor... but the truth is that i've been 'playing' with it for two weeks with no luck.

What i'm trying to do is to make sure that everything is properly configured by reading product id register (0x56), but each time i read it, i get different values being 0xF211 the most usual together with 0x4010, 0x4080 or 0x0011, something that doesn't make much sense as i'm doing this in a loop so i guess i should getting always the same values as i'm always requesting the same adress. This happens too with all registers i've tried like status or miscellaneous control. This makes me think that it could be an incorrect spi mode being used during the transmission, but i've tried all of them and the sensor: a) doesn't reply b)still bad data. I've also longered time between each read (about 175us) and even used dma mode to reduce time to short the inactive clk status between 1st and 2nd byte, but still the same readings, and if i read the register once (2 16bits cycles: one for selecting the register and the other for reading the actual value) what i get is 0x00 (normal as it's the 1st read after reseting the sensor) and 0x4010 wich again is not what i was expecting. I've also tried to give a meaning to what i read and it's, obviously, no sense because if we consider that EA is set as some readings state, we could say that power suply is below 4,75 when that is not true... but again this register is no sense because as i can read in the data sheet Status register uses the full frame for its data while in the data sheet says that the two first bit are used for ND and EA. ¿does it mean that with this register that's not true? Another thing that i don't like is that clock stays high to much time between 1st and 2nd byte, i know that this shouldn't be a problem but right now i can't find another reason wich explains why is not working properly so any help will be welcome. Btw, i asked for help on microchip's forum so have a look at <http://www.microchip.com/forums/m608268.aspx> for all the info i haven't supplied you in this email.

Thanks in advance and regards from Spain!

Josep Gil

---



► Resposta de *Mark Looney* al nostre correu del 19 d'Octubre:

**de:** "Looney, Mark" <[Mark.Looney@analog.com](mailto:Mark.Looney@analog.com)>

**Fecha:** 19 Oct. 2011 20:13

**Asunto:** RE: Questions about using ADIS 16400

Hello Josep,

I am sorry that this hasn't come together more smoothly for you. Can you send me a scope shot of your repeating read with DIN = 0x5600, with the trigger on the falling edge of CS, the horizontal scale adjusted to show all 16-SCLK cycles and the vertical aligned to see all four SPI signals?

What company are you with? Any particular application in mind?

Best,

Mark

**Mark Looney**

iSensor Application Engineer

Analog Devices, Inc.

---

## 20 d'octubre del 2011

► Resposta al missatge de *Mark Looney* del 19 d'Octubre:

Hi, thanks for replying so quickly :)

About what is happening with this sensor.... well, is what happens the first time you work with something wich you've never used so nothing new. Btw, i've attached the pic you asked me<sup>63</sup>. I have to say that this reading, although it's inside a loop, hasn't been done the same way that i used when i posted at microchip forum my problem because while surfing the web i found a french guy who, in order to read it, waited for two flags to be set in the microcontroller, one being the busy flag (wich sets when transmission has ended) and ss2p interrupt flag (wich is set when the in/out buffer is empty), it shouldn't be a big differene but yesterday when i added that little part, i had a repetitive sequence of 0xF211 while today, although my code is still the same, i get different (repetitive) values. I also replaced the read of 57, 56 for 56, 56. This is the code i have right now:

```
PORTAbits.RA2=1; //put high adis reset terminal
```

```
while(PORTBbits.RB0==0); //wait until DIO1 pulses
```

```
while(PORTBbits.RB3==1){ //debug: i think my adis 16400 has a firmware bug wich prevents it to use DIO1 as data ready terminal
```

```
//so i give it "manually" time to get properly initialized before reading/writing to it
```

```
while(PORTBbits.RB2==1){ //Send 2x16bits while no body presses switch on RB2
```

```
PORTBbits.RB4=0; //CS low
```

---

<sup>63</sup> La imatge que es va adjuntar amb aquest correu correspon a la figura 99 d'aquesta memòria.

```

SSP2BUF=0x56;    //Register to read (ID)
while(SSP2STATbits.BF==0);    //wait until buffer register is empty
while(PIR3bits.SSP2IF==0); //debug: wait until rx/tx is completed
PIR3bits.SSP2IF=0; //debug: reset complete rx/tx
hbyte=SSP2BUF;    //not necessary as it's not real data but recommended
SSP2BUF=0x00;    //dummy byte
while(SSP2STATbits.BF==0);    //wait until buffer register is empty
while(PIR3bits.SSP2IF==0); //debug: wait until rx/tx is completed
PORTBbits.RB4=1; //CS high
PIR3bits.SSP2IF=0; //debug: rest completed rx/tx
lbyte=SSP2BUF;    //not necessary as it's not real data

Delay10TCYx(3);    //wait about 12us

PORTBbits.RB4=0; //CS low
SSP2BUF=0x56;    //ID
while(SSP2STATbits.BF==0);    //wait until buffer empty
while(PIR3bits.SSP2IF==0); //debug
PIR3bits.SSP2IF=0; //debug
hbyte=SSP2BUF;    //store first 8bit
SSP2BUF=0x00;    //dummy
while(SSP2STATbits.BF==0);    //wait until buffer empty
while(PIR3bits.SSP2IF==0); //debug
PORTBbits.RB4=1; //CS high
PIR3bits.SSP2IF=0; //debug
lbyte=SSP2BUF;    //store last 8bit

Delay10TCYx(3);    //wait about 12 until CS can be put low again

} //while porb rb2

```

Changes on the code, as i imagined, do no change what is happening so what now? do you think that the IMU is losing synchronization and that's why i can't read it properly? if not, would it be because i haven't written any routine to initialite it? as far as i understand when reading the data sheet is that is not necessary because as soon as it has finished his initialization routines it starts to pulse DIO1 and waits until the master processor toggles CS to read it to send data via spi.

Btw, i'm working in a company named "Mediterraneo Señales Maritimas" and our intention is to check wheter this sensor will help us or not to design a stabilization control system for a floating plataform.

Regards and thanks again for your help!

Josep

p.d if you need anything else, just let me know



## 24 d'octubre del 2011

► Resposta de *Mark Looney* al nostre correu del 20 d'Octubre:

*Hello Josep,*

When you say that you are getting different (repeating) values now, what are you receiving? The attached plot still showed 0xF211, unless I misinterpreted it.

Best regards,  
Mark

**Mark Looney**  
iSensor Application Engineer  
Analog Devices, Inc.

---

► Resposta al missatge de *Mark Looney* del 24 d'octubre:

**From:** Josep Gil [<mailto:piteta1@gmail.com>]  
**Sent:** Monday, October 24, 2011 6:06 AM  
**To:** Looney, Mark  
**Subject:** Mediterraneo Señales Maritimas; ADIS16400; SPI errors

Hi, what i meant with that is that it's like if the sensor was sending values inside a loop because all the values i've seen are always the same: 0xF211, 0x4010, 0x4080, 0x0011 or 0x0000 and sometimes it seems they come in the same sequence... so what i meant is that it could be sending a sequence.

Regards,

Josep Gil

---

► Resposta de *Mark Looney* al nostre correu del 24 d'octubre:

Looney, Mark <[Mark.Looney@analog.com](mailto:Mark.Looney@analog.com)>

Hello Josep,

0x4010 is the correct answer. I would step through each 16-bit, once shot at a time. Make sure that DIN = 0x5600 and DOUT = 0x4010, every time and look for any evidence of “glitches” or noise on the SCLK lines.

Regards,  
Mark

**Mark Looney**  
iSensor Application Engineer  
Analog Devices, Inc.

► Resposta al missatge de *Mark Looney* del 24 d'Octubre:

**From:** Josep Gil [<mailto:piteta1@gmail.com>]  
**Sent:** Monday, October 24, 2011 10:21 AM  
**To:** Looney, Mark  
**Subject:** Re: Mediterraneo Señales Maritimas; ADIS16400; SPI errors

Hi, if you say that dout has to be 0x4010 when i send 0x5600, then there's a mistake in the data sheet because it says that the default value for that adress is 0x4105. About the other things you say, 'll check them when i get back to the office and i let you know how has this ended. Meanwhile i have one question: if i send 0x5700 instead of 0x5600 i should also get the value of the adress, shouldn't I?

Regards and thanks again for your help!

Josep

---

► Resposta de *Mark Looney* al nostre correu del 24 d'Octubre:

**From:** Looney, Mark <[Mark.Looney@analog.com](mailto:Mark.Looney@analog.com)>  
**Date:** 2011/10/24  
**Subject:** RE: Mediterraneo Señales Maritimas; ADIS16400; SPI errors  
**To:** Josep Gil <[piteta1@gmail.com](mailto:piteta1@gmail.com)>

I agree. 0x4105 needs to be 0x4015 for the ADIS16405 and 0x4010 for the ADIS16400. This has been submitted to our corporate editors, but it has not made it to the web. Yes, 0x5600 and 0x5700 should yield the same result, on the DOUT cycle after they are issued.

**Mark Looney**  
iSensor Application Engineer  
Analog Devices, Inc.

---

► Resposta al missatge de *Mark Looney* del 24 d'Octubre:

yup, then somebody should make appear that error on the web because i guess i wont be the only one who will fall for that...  
btw, thanks for confirming that error and also for the part about sending 0x5700 instead of 0x5600.

Regards,

Josep

► Resposta de *Mark Looney* al nostre correu del 24 d'Octubre:

You bet!

**Mark Looney**  
iSensor Application Engineer  
Analog Devices, Inc.

---

## **25 d'octubre del 2011**

► Correu del 25 d'Octubre del 2011

**From:** Josep Gil [<mailto:piteta1@gmail.com>]  
**Sent:** Tuesday, October 25, 2011 1:57 PM  
**To:** Looney, Mark  
**Subject:** Re: Mediterraneo Señales Maritimas; ADIS16400; SPI errors

Hi, it's me again. I forgot to ask you if there was any other errata in the data sheet that i should keep in mind. Also, 'ive tried to read the id register and most of the times i can read it properly, but when i try to read other registers with default values specified at the data sheet (those with 0x800) readings do not never match or look a bit a like :S

Regards,

josep

---

► Resposta de *Mark Looney* al nostre correu del 25 d'Octubre:

Hello Josep,

This sounds like the SPI communication is getting corrupted. It looked like you had your coding correct before, so it may be time to start looking at noise and timing.

We normally publish "Silicon Anomaly" sheets, which are available with the datasheets online. Here is the latest version for the ADIS16400:

[http://www.analog.com/static/imported-files/data\\_sheets/ADIS16400\\_16405\\_Anomaly.pdf](http://www.analog.com/static/imported-files/data_sheets/ADIS16400_16405_Anomaly.pdf)

hese do not cover errors in the datasheet, as we normally just fix them and list the change in the table of contents.

I hope that helps.

Best,  
Mark

Mark Looney  
**26 d'octubre del 2011**

► Resposta al missatge de *Mark Looney* del 25 d'Octubre:

**From:** Josep Gil [<mailto:piteta1@gmail.com>]  
**Sent:** Wednesday, October 26, 2011 1:00 PM  
**To:** Looney, Mark  
**Subject:** Re: Mediterraneo Señales Maritimas; ADIS16400; SPI errors

Hi, i don't think its code related because changes are minimal (read process moved to a function) so it must be what you suggest: spi communication is getting corrupted.

About errors i meant in the data sheet, not at the sensor itself. I asked this because when reading SENS\_AVG register i get 0x406 instead of 0x402... ¿Spi corruption or datasheet errata?

Regards and thanks again for your help :)

Josep

---

► Resposta de *Mark Looney* al nostre correu del 26 d'Octubre:

Hello Josep,

I will check with our Production Test Engineers, to see if they have experienced any issues with this. If so, I will definitely update the silicon anomaly to reflect it. You are sure that this has not been changed in your use? Could you send me a picture of the top label, or just give me what is on each line in the label? If there are any lot-specific issues, this will be important for traceability.

Best regards,  
Mark

Mark Looney  
iSensor Application Engineer  
Analog Devices, Inc.

---

► Resposta al missatge de *Mark Looney* del 26 d'Octubre:

**From:** Josep Gil [<mailto:piteta1@gmail.com>]  
**Sent:** Wednesday, October 26, 2011 1:18 PM  
**To:** Looney, Mark  
**Subject:** Re: Mediterraneo Señales Maritimas; ADIS16400; SPI errors

Hi, i don't know what to say about if i changed anything because of the communication problems i'm having. The only things i've done is try to set SPS=1 and to activate DIO1 as Data Ready

terminal because on my unit this pin never pulses. I also tried to do a gyroscope automatic bias null calibration after you told me that 4010 was ok but don't know if it worked because of my communication errors :S

About the data you ask:

ADIS16400BLMZ  
0059  
AD051110-3  
USA 34031  
1048

knowing this, could you tell me wich is the manufacturing data? i ask this because i haven't seen anywhere how to get that and i'd like wich silicon anomalies will i have to face...

Regards,

josep

---

► Resposta de *Mark Looney* al nostre correu del 26 d'Octubre:

The only documentation we offer on “known issues” is in the link below. If this is a new issue, then it will be added. Need the test engineer’s input before offering any further comment, outside of it would be surprising. If you can’t read 0x4010 every time on the PROD\_ID register, then you have communication issues. If the SENS\_AVG register was the wrong value, when it was shipped, that would be troubling to us, but not hard to overcome for our customers. Every part that goes out has been accessed 100s of times during calibration and test. We have no known issues of basic function in devices are in our stock or in the field. Occasionally, we will need to update our process and a register value may end up incorrect or early in the product life, we find some minor operational issues (see Silicon Anomaly below), but that isn’t the norm. We have 10s of thousands of these in the field and the return rate are very low. Most returns actually have no evidence of failure and of those that do, most are due to mishandling. The instance of shipping a part that does not meet specs is extremely low. I can only think of 2 or 3 cases. 100-200ppm or better, which is typically for statistical prediction on “human intervention.” I understand where you are coming from but I don’t think that you are going to find an answer to the communication errors in the manufacturing feedback.

Best regards,  
Mark

**Mark Looney**  
iSensor Application Engineer  
Analog Devices, Inc.

---

► Resposta al missatge de *Mark Looney* del 26 d'Octubre:

Hi, i don't think i have a faulty device because it replies my answers, the problem is that is not what

i expect so it's more likely to be a communication error than a faulty device, it could be the last option but i agree with you and it'd be surprising if my problems were related with that option.... the problem is that i don't know what to think because right now i want to be sure that i have no longer communication errors and each time i've tried to read supply out and i have no reply while if i try to read product id or sens\_avg i receive every time what i think it's ok (4010 and 406) :S

Regards,

josep

---

## **8 de Novembre del 2011**

► Correu del 8 de Novembre del 2011

**From:** Josep Gil [<mailto:piteta1@gmail.com>]

**Sent:** Tuesday, November 08, 2011 1:22 PM

**To:** Looney, Mark

**Subject:** Re: Mediterraneo Señales Maritimas; ADIS16400; SPI errors

Hi Mark, It's Josep again.... may be you forgot about me so i'll refresh your memory. Some weeks ago i wrote you because i was having some trouble when i tried to communicate with an Adis 16400. You told me that what was happening was because data was getting corrupted and you were right because i changed the spi clock to 750Khz (instead of 180khz) and those problems disappeared. But then, i found out that the value i read from Sens\_Avg register didn't match with the default values specified on its data sheet. I told you about this and you said you'd ask your production test engineers and let me know about their answer if they had one. Later i found out that that wasn't the only register with different value so i assumed this was because while i was trying to read the sensor, what i was really doing was a combo of read/write because of the communication problems i had at the beginning so after writing the default values to the most important registers, i started to read the data output (gyro/accl) registers. At first, as i only made one read (X Accl) and walked a bit with the system on my hands, i thought everything was OK, but later when i read the remaining registers, i realized that, from time to time, the readings from the gyroscope suddenly change from 3FFx to 0x000x although i'm not moving the system. This also happens with the accelerometer, but less often and unlike the gyroscope, only on the Y axis.... and that's why i'm contacting you again: is this normal? i mean, if i'm not moving shouldn't i have readings with only the LSB changing?

Btw, while i was trying to figure out why this is happening, i thought again of some sort of communication problem and decided to read the sensor a bit slower (500ms vs 1s) but still happens the same: randomly reads jump from 3FFxx to 0x000x. I've also performed the self test, memory test and a fast gyroscope bias calibration, but still the same. I also thought that perhaps it was because i hadn't written all the registers with its default values so I wrote the code to check and do it if necessary and there's something a bit odd: GPIO ctrl never updates with 0x0000, not even doing a manual flash update and keeps always being configured as 0x0F00. I guess this doesn't have anything to do with what is happening, but i can't understand why i can't change it to 0x00. ¿Is it because this register is the one with less priority when setting the function DIOx pins? or may be is that i have to do something special? I ask this because i use a (c) function to write to the adis and with the other registers it works fine, and also because in the datasheet, when it's talking about doing a manual self test, it says that we have to wait 50ms after writing Sens AVG and before



reading X Gyro, but in the data sheet, when it talks about filtering and dynamic range, it doesn't say that there must be a delay after writing it... so do i have to wait with that (and other config registers) one too? By the way, going back to the jumps between readings, i've also thought of performing more calibrations but i've realized that i can only calibrate the gyroscope. What about the accelerometer? should it be calibrated before use? I've also seen that on the data sheet, there's a bit on the msc ctrl register named linear accelerometer origin alingment the function of witch is not explained in the data sheet. what is it exactly? I think it hasn't anything to do with my problem, but i'd like to know what does it do because it may come handy in my project. Also i'd like to ask you if you had more detalied info about this IMu because to me and my mates the data sheet available on the web is too simple, with some errors and also a bit confusing as it doesn't give much detail on the sensor... so if you have it, it'd be great to have it because controlling this small beast is taking more than we thought....

Regards,

Josep Gil

---

► Resposta de *Mark Looney* al nostre correu del 8 de Novembre del 2011

Hello Josep,

I do remember you and am glad that you sent me this message. I appreciate the detail you provide in asking your questions because it saves so much time on our end. Thank you!

It looks like your unit had the appropriate programming so I am not sure why the register had the wrong value. As you said, perhaps it was accidentally changed during your initial development. If you see this on future parts, please let me know. With respect to your data, when using 14-bit, twos complement, 0x3FFF and 0x0000 are only 1 LSB apart. We are working to update this in the ADIS16400 datasheet, but the ADIS16407 datasheet offers numerical examples in Tables 12, 16 and 20, starting on page 11.

[http://www.analog.com/static/imported-files/data\\_sheets/ADIS16407.pdf#Page=11](http://www.analog.com/static/imported-files/data_sheets/ADIS16407.pdf#Page=11)

I suspect that the noise will cause your readings to change by more than an LSB.

I hope that helps.

Let me know if there is anything else we can do to help. We appreciate your investment in designing the ADIS16400 into your system.

Best regards,  
Mark

**Mark Looney**  
iSensor Application Engineer  
Analog Devices, Inc.

## 9 de Novembre del 2011

► Resposta al missatge de *Mark Looney* del 8 de Novembre del 2011:

**From:** Josep Gil [mailto:[piteta1@gmail.com](mailto:piteta1@gmail.com)]

**Sent:** Wednesday, November 09, 2011 11:27 AM

**To:** Looney, Mark

**Subject:** Re: Mediterraneo Señales Maritimas; ADIS16400; data format

Hi, thanks for your reply. It has helped more than i first thought when i had a quick look to your reply last night. Right now i've placed an emi filter and shortened some cables and the accelerometer output its quite stable (for exemple 2/3LSB between 1st and 5th read). Gyroscope output its more stable than at the beggining, but the output still varies about 10LSB so i guess that i'll have to try different filter settings and protect better my prototype against the effects of noise/interferences. Seeing the datasheet you've linked in your mail and what you've remembered about twos complement, i guess that it'll be also necessary a manual calibration of the sensors because as i suspected, output while not moving should be near 0.

Regards and thanks again for your comments :)

Josep Gil

---

► Resposta de *Mark Looney* al nostre correu del 8 de Novembre del 2011:

Hello Josep,

I am glad that helped and am glad that improving the EMI exposure has helped. There will always be some noise but it is good eliminate influencers that you can control to get the most value out of the solution.

The ADIS1640x gyroscopes are factory calibrated but do have rate random walk characteristics, like all MEMS gyroscopes, so they may need periodic re-calibration, if you application requires them to be at optimum levels. The typical performance is 10-20 deg/hour/sqrt(hr). So, over a period of 256 hours, the rate can drift  $20 \times \sqrt{256} = 320$  deg/hour. Exposure to temperature extremes and other forms of mechanical stress can accelerate this. This does not appear to influence the bias vs. temperature behaviors (which were calibrated at the factory), so you only typically need to reset this at one temperature to "restore" factory performance.

Hope that helps.

Best regards,

Mark

**Mark Looney**

iSensor Application Engineer

Analog Devices, Inc.

## **27 de Juny del 2012**

► Correu del 27 de Juny del 2012:

On Jun 27, 2012, at 4:37 AM, "Josep Gil" <[piteta1@gmail.com](mailto:piteta1@gmail.com)<mailto:[piteta1@gmail.com](mailto:piteta1@gmail.com)>>> wrote:

Hi Mark, it's me again fighting against an ADIS16400... well, right now it shouldn't be considered as that because i'm almost done with the project i was working on, but in the final steps, while i was "playing" with Digital filtering i came across something i just can't understand: i've allowed the user to chose how many taps will each stage of the barlett filter have and while i was checking everything was OK i observed that the highest value i can select is 6 (110). Shouldn't it be 7 (111) because there're 3 bits dedicated to that setting? what am i missing?

Regards and thanks in advance for your help!

Josep

---

► Resposta de *Mark Looney* al nostre correu del 27 de Juny del 2012:

Hello Josep,

It is great to hear from you. The filter tap settings are limited to 110, the 111 option should revert to 110 automatically. That is just the way the part was defined.

Best regards, Mark  
Sent from my iPhone

---

## **14 d'agost del 2012**

► Resposta al missatge de *Mark Looney* del 7 de Juny del 2012:

On Aug 14, 2012, at 8:41 AM, "Josep Gil" <[piteta1@gmail.com](mailto:piteta1@gmail.com)<mailto:[piteta1@gmail.com](mailto:piteta1@gmail.com)>>> wrote:

Hi mark, i thought i had got back to you some weeks ago with one more question but it seems i didn't :\$

Btw, thanks for your reply. When i saw that each time i wrote 111 to it and it reverted to 110 i suspected that there was no 111 option, but i wanted to get sure.

At the time i wrote you i also wanted to ask you about GPIO\_CTRL register. I don't know if you remember, but when i started to deal with this IMU i had some data corruption on the communication. That data corruption caused random writes on the IMU and those random writes were stored on the IMU flash memory, or at least that's what we deduced because i was able to restore all registers to its default value, except for GPIO\_CTRL. This registers fails to be 0,

whenever i write 0 to it, it reverts to 0x0F00. Is there any limitation with this register too?

Regards from Spain,

Josep

---

► Resposta de *Mark Looney* al nostre correu del 14 d'Agost del 2012:

Hello Josep,

I remember. It is great to hear from you! Are the I/O pins tied high? When the I/O pins are configured as inputs, these bits will reflect their logic level, per Table 18 on page 14 of the datasheet. I hope that helps. Let me know what you find out.

Best regards, Mark  
Sent from my iPhone

---

► Resposta al missatge de *Mark Looney* del 14 d'Agost del 2012:

On Aug 14, 2012, at 12:55 PM, "Josep Gil" <[piteta1@gmail.com](mailto:piteta1@gmail.com)<mailto:[piteta1@gmail.com](mailto:piteta1@gmail.com)>>>  
wrote:

Hi Mark, thanks for replying as fast as usually :)

I checked what you asked because i didn't remember where did i connect DIOx pins none of them are used as inputs, in fact DIO3 and DIO4 are not connected and DIO1 and DIO2 are connected to two pins configured as inputs at the microcontroller side, although only DIO1 is being used because of its Data ready function. Btw, if you say that those register bits are the value of those inputs, then is no sense to write that part of the register while those pins are configured as inputs so i guess i shouldn't care about that... should i?

Regards and thanks again for your help :D

Josep

## **15 d'agost del 2012**

► Resposta de *Mark Looney* al nostre correu del 15 d'Agost del 2012:

Hello Josep,

If the pubs are configured as inputs (they are when bits 3:0 are zero), then these bits will not provide write access, so you are correct.

Best regards, Mark  
Sent from my iPhone



► Resposta al missatge de *Mark Looney* del 14 d'Agost del 2012:

That's what i suspected.

Thanks again for your help!

Josep



