

DISEÑO E IMPLEMENTACIÓN DE UN ROBOT CUADRÚPEDO 3GDL CON MICROCONTROLADOR DE 32 BITS

Autor del trabajo: Juan Daniel Miret Rubio

Director de la tesis: D. Juan José Serrano Martín

CONTENIDO

- Marco teórico y Objetivo
- Selección de componentes
- Diseño mecánico
- Diseño electrónico
- Ensamblaje y Estudio del par
- Conexiones y programación principal
- Cinemática
- Calibración de servomotores
- Sensor IMU o placa MEMS
- Comunicación por RF
- Movimiento y rotación del robot
- Programación de código con μ Vision5
- Conclusiones

MARCO TEÓRICO Y OBJETIVO

- Con el paso de los años los robots han ido evolucionando hasta convertirse en humanoides. Los robots están presentes en todos los ámbitos de nuestra vida:
- En la industria, en la sanidad, en el hogar, al aire libre, en otros planetas.

- La aspiración de este proyecto era diseñar e implementar un robot cuadrúpedo con el fin de darle diferentes utilidades a posteriori.
- La idea era seguir la evolución del robot mejorándolo por ejemplo con un controlador inalámbrico.

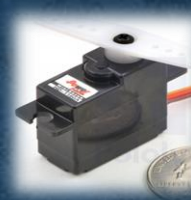
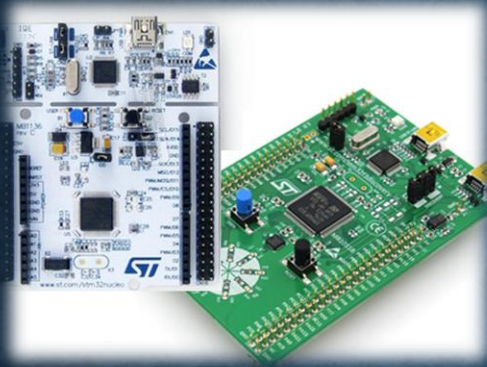
¿Porque un cuadrúpedo y no otro tipo de robot?

Era una buena forma y relativamente más sencilla que un humanoide (que podría resultar más engorroso debido a que tiene más grados de libertad) para empezar a construir un robot ya que el autor nunca había construido ninguno.

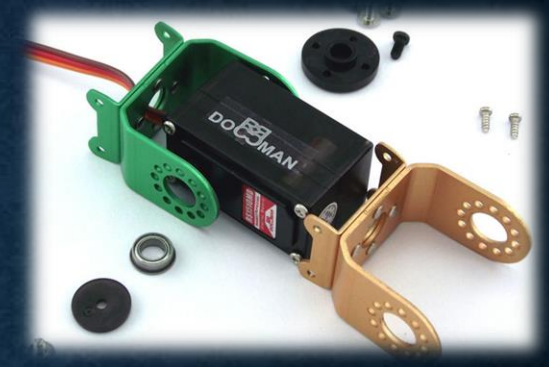


SELECCIÓN DE COMPONENTES

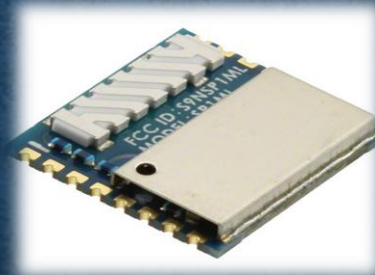
Placas electrónicas
con μ controlador STM



Servomotores DOMAN 270°



Batería lipo Gens Ace
2C 1800mah

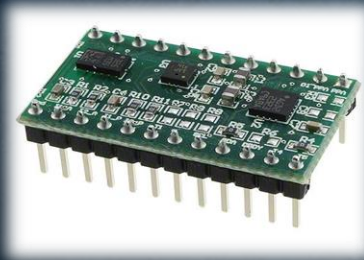


Módulos inalámbricos
de RF 1GHZ SP1ML



Joystick

Sensor IMU
Acel.+Giros. MKI124V1



Medidor de distancia Sharp GP2D12

DISEÑO MECÁNICO

- La mejora del diseño mecánico vendría dada por 3 aspectos que dotarían al robot de una evolución y un toque diferencial.

1. Dotar al robot de 1 GDL mas en cada pata.

Mayor Tamaño

Mejor Movilidad 3D

2. Diseñar un cuerpo simétrico, envolvente y unos pies modernos y resistentes.

Omnidireccional

Diseño compacto y robusto

3. Crear estas partes con impresora 3D.

Diseño inteligente y eficiente

Toque innovador y elegante

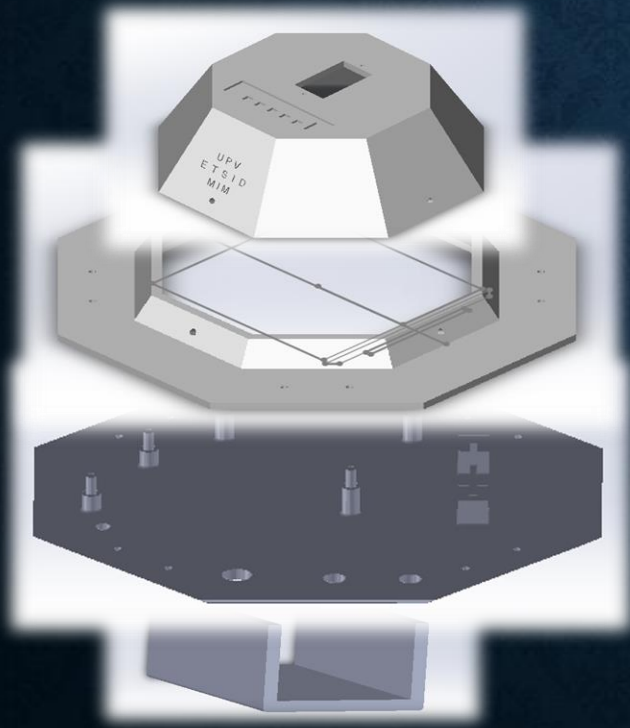
DISEÑO MECÁNICO

1. Añadiendo 1 servo mas por pierna, pasamos de 2GDL a 3GDL en cada pierna. De 8 servos a 12.
2. Se diseño un cuerpo con forma octogonal con soportes para placas y batería y con tapa.

Para los pies se fue mejorando un diseño haciendo que los puntos de rotura fueran menores.

ROBOT 3GDL

SOLIDWORKS

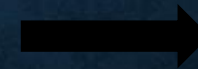
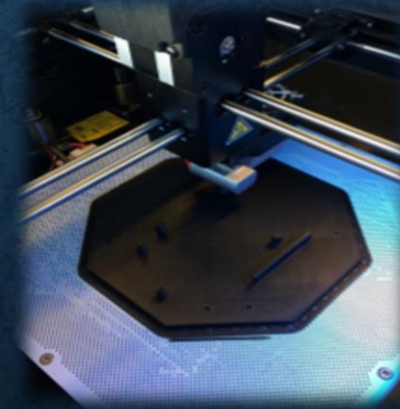
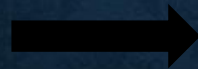
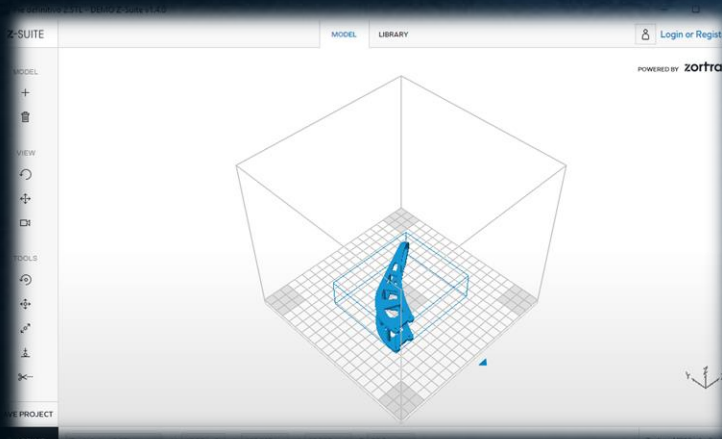


DISEÑO MECÁNICO

3. Gracias a las impresoras Zortrax de la ETSID se pudieron crear todas las piezas con acabado perfecto.

- El programa Z-suite sirvió para elegir el grosor de capa, la calidad el relleno , la posición de impresión etc....
- También proporcionaba información sobre el tiempo estimado que duraría la impresión. Pudiendo visualizar el soporte que realizaba para poder imprimir la pieza en tal posición.
- Se eligió el material de impresión Z-HIPS ya que proporcionaba una resistencia alta al impacto y a la tracción.

Para los pies se fue mejorando un diseño haciendo que los puntos de rotura fueran menores.



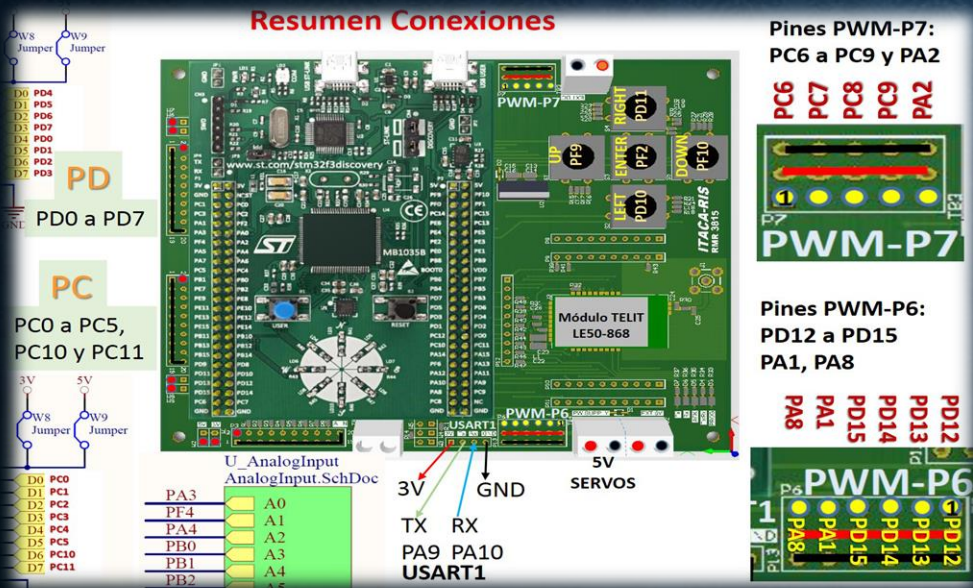
DISEÑO ELECTRÓNICO

Para empezar con el diseño electrónico del robot 2GDL se eligió el bloque STMF3DISCOVEY+RMR3815.

1. Este bloque ya había sido visto en clase. Era una forma rápida de empezar a conectar y programar.
2. La F3DISCOVERY lleva incorporado un MCU CortexM4, USB ST-link y proporciona salida de 3 y 5 V.
3. La RMR 3815 es una PCB acopladora para la Discovery. Fue creada por ITACA para practicas.

3.1 Ofrece conexiones mas organizadas y mayor cantidad de pines de salida, de USART's, de 3 y 5 V.

3.2 Tiene una entrada de alimentación adicional para alimentar correctamente a los servomotores.



Para seguir con la evolución electrónica del robot se creó un mando con el que el usuario pudiera controlarlo.

La configuración electrónica del mando tiene este mismo bloque electrónico.

La botonera que ofrecía la RMR 3815 más un Joystick eran esos elementos que necesitaba el mando.

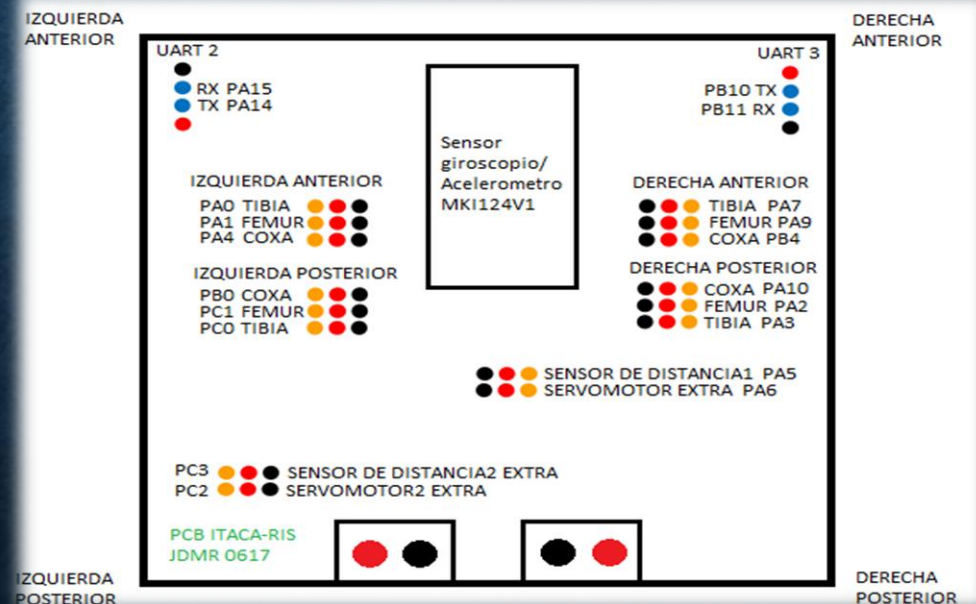
DISEÑO ELECTRÓNICO

Siguiendo a la evolución del robot de 3GDL se pensó en utilizar un bloque electrónico de menor tamaño .

1. La Placa Núcleo STMF334R8 fue la elegida por su menor tamaño e iguales características.
2. Se diseño una PCB con las mismas características de la RMR3815 y que acoplara con la F334R8.
La Placa JDMR 0617.

2.1 Para diseñar esta PCB el autor de este proyecto tuvo que familiarizarse previamente con el programa Altium Designer.

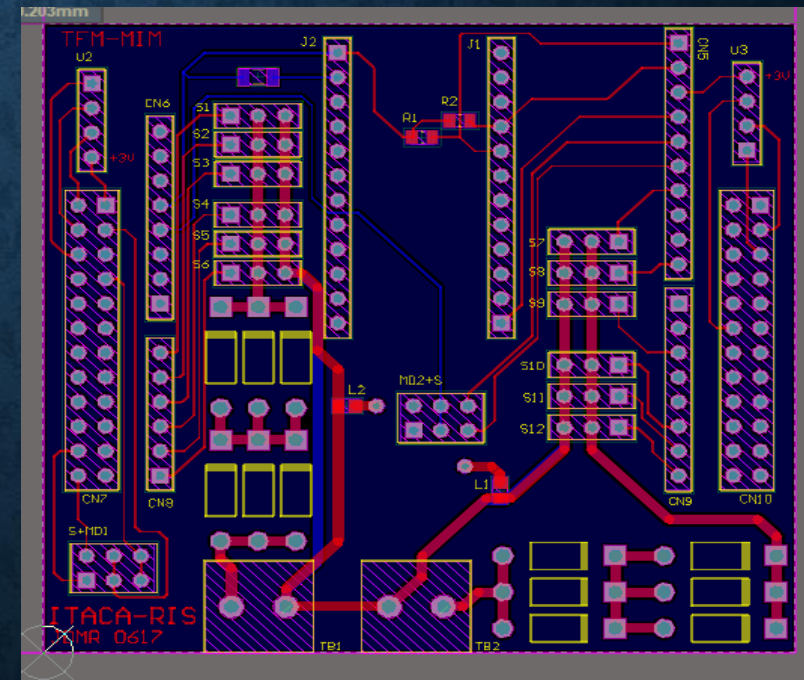
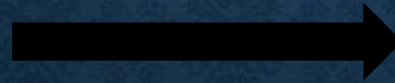
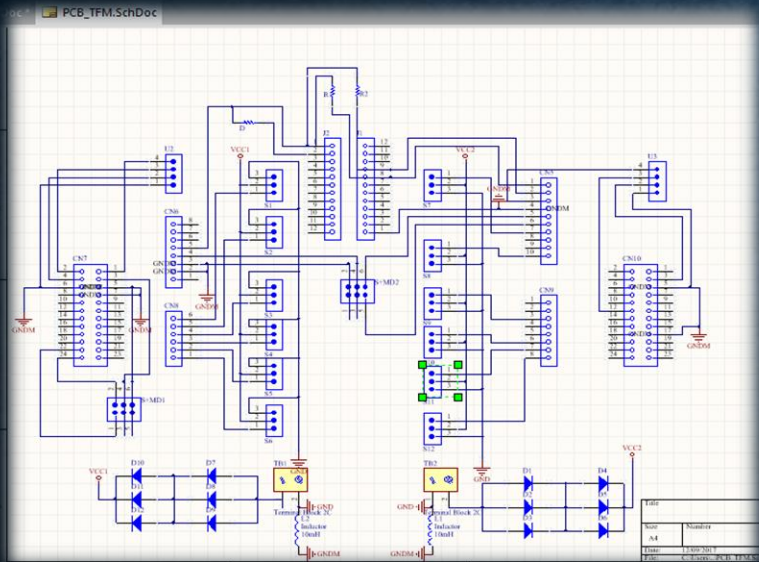
2.3 También se realizo un boceto con las conexiones que se querían realizar.



DISEÑO ELECTRÓNICO

Para el diseño en Altium de la PCB JDMR0617 se siguieron estos pasos principales:

1. Creación de un Esquemático: Elección de conectores , pines, señales y posterior cableado del circuito.
2. Creación de Tablero con sus diferentes capas y transferencia del esquemático al Tablero PCB.
3. Posicionamiento de los elementos electrónicos en el tablero y Enrutamiento de las vetas.
4. Eliminar posibles interferencias o ruidos mediante un plano de masa y dos bobinas.
5. Prevenir cortocircuitos y rebajar la Tensión de la batería con diodos rectificadores.
6. Serigrafiar la PCB con su nombre , crear los archivos GERBER de cada capa y enviarlos para su fabricación .

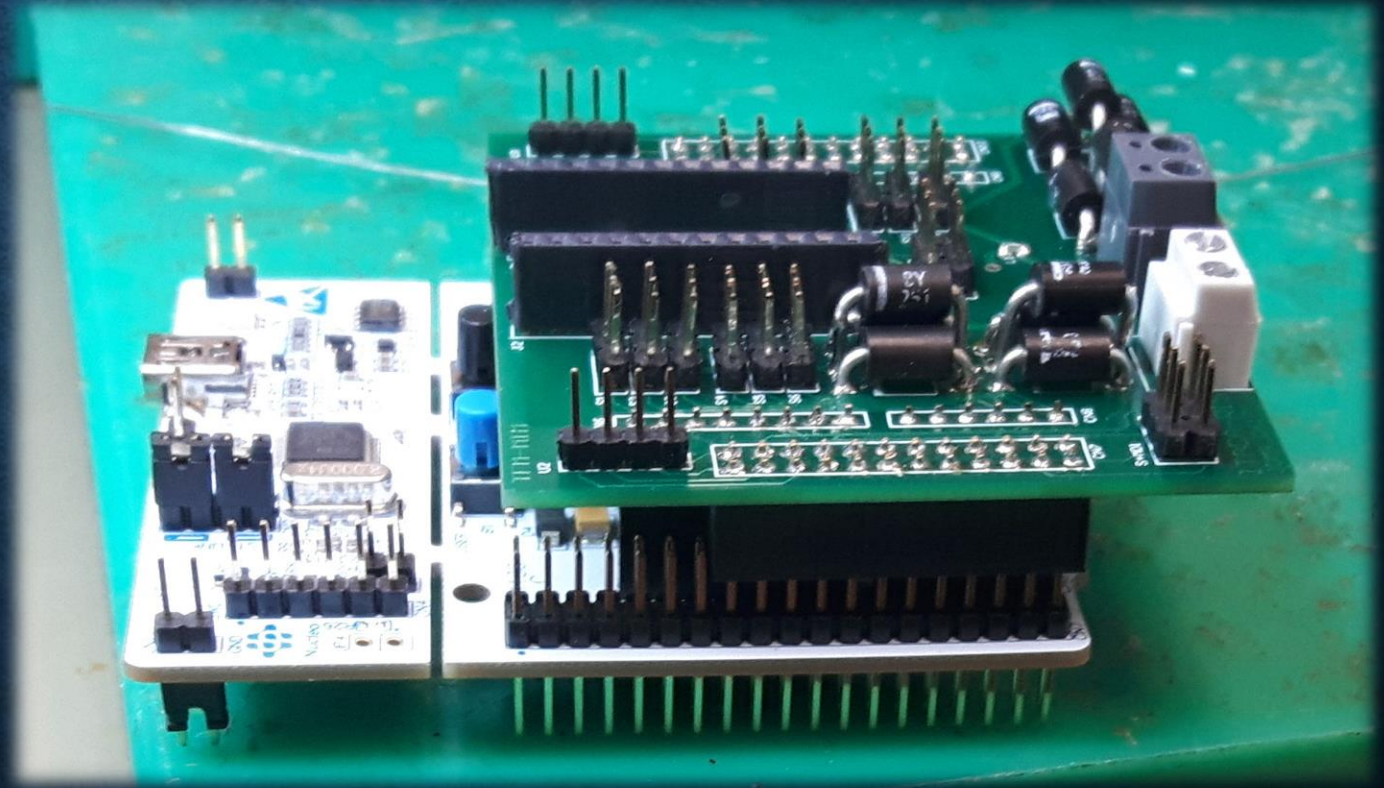


DISEÑO ELECTRÓNICO

La PCB tardo 2 semanas en ser fabricada.

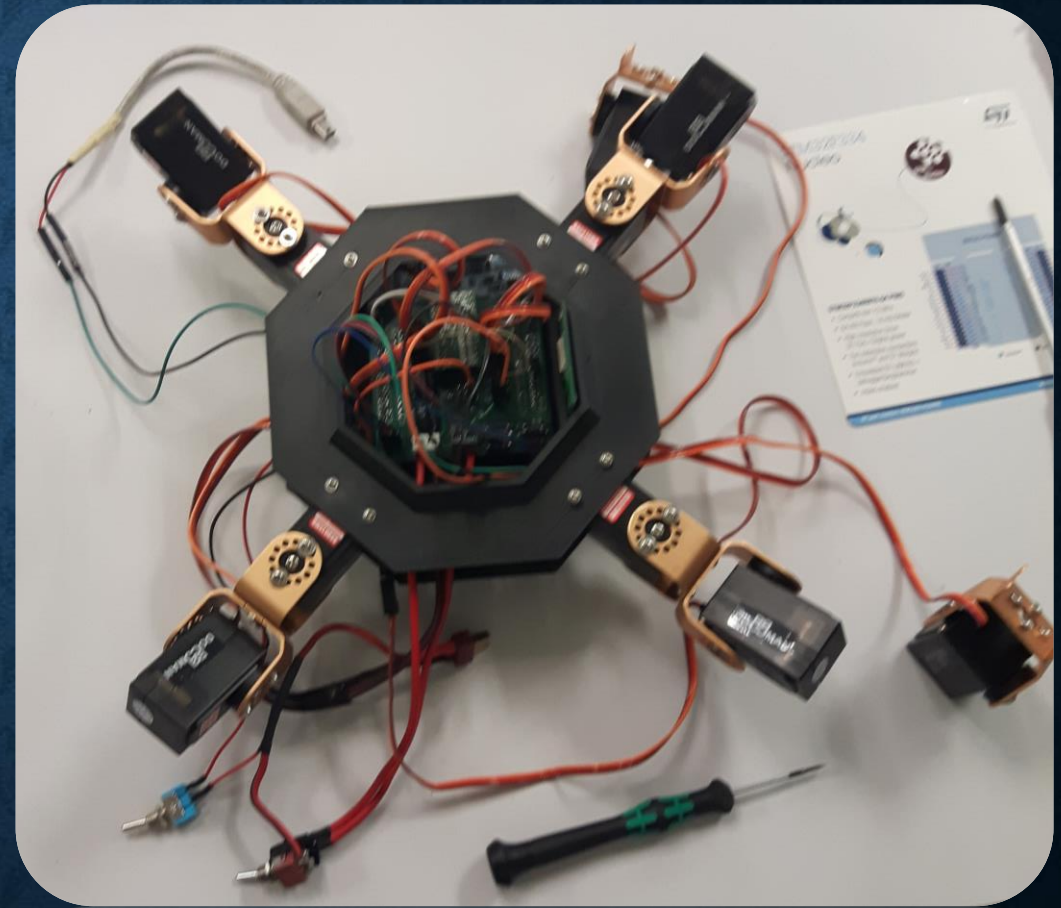
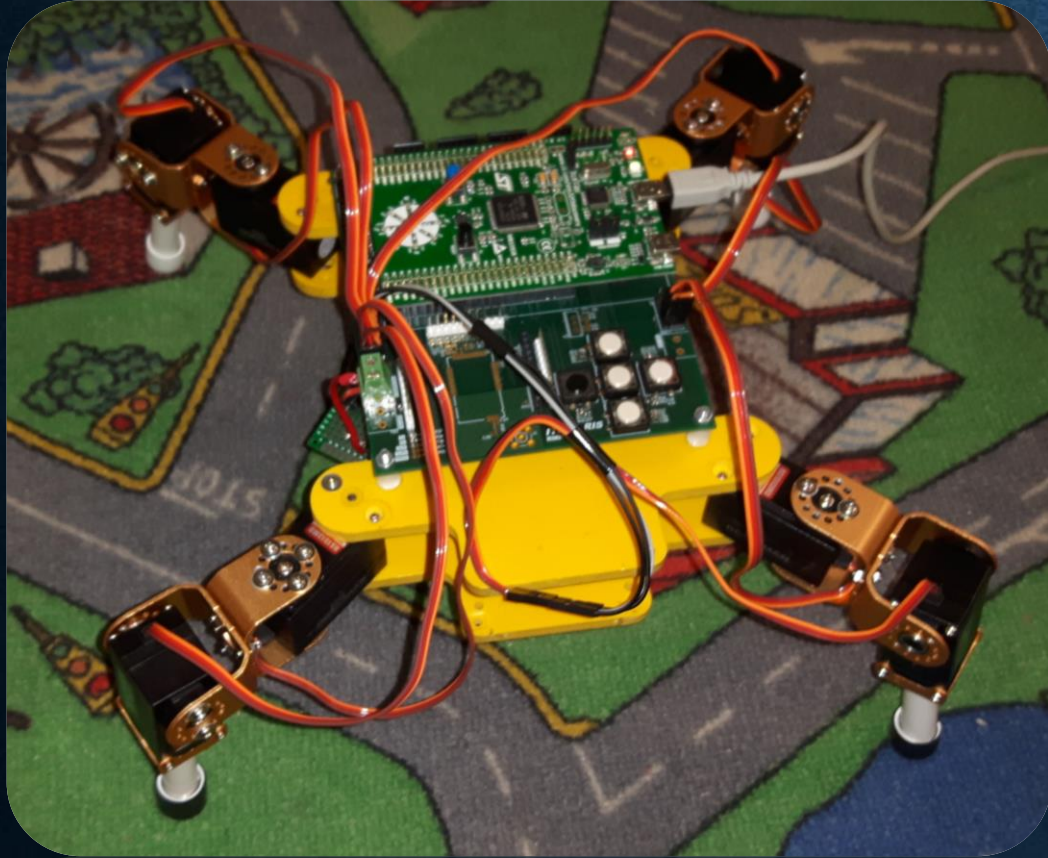
El último paso fue soldar los componentes en sus respectivas huellas digitales.

Esta tarea se realizo en los puestos de soldadura que ofrece el laboratorio de ITACA.



ENSAMBLAJES

Los ensamblajes de los robots 2GD1 y 3 GD1 quedaron de tal forma:



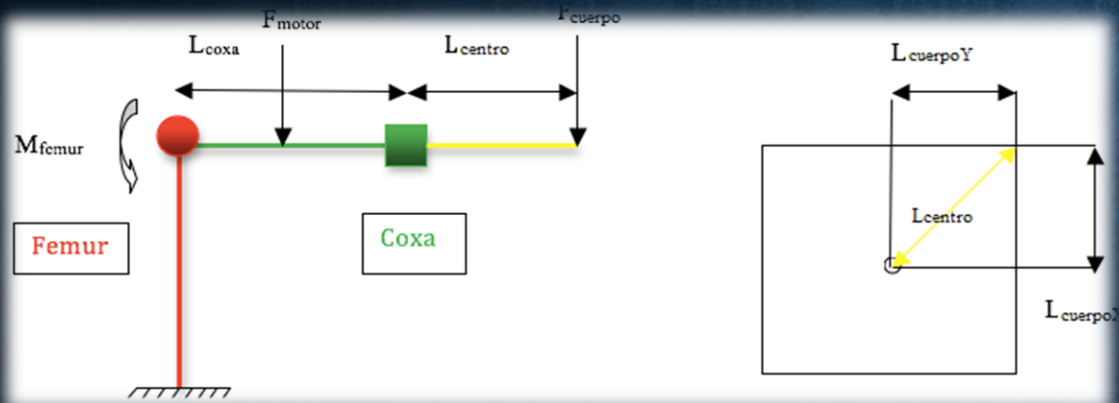
ESTUDIO DEL PAR

En la lista de servomotores se podían encontrar dos modelos, los RS0613MD y los RS1513MD

- Los RS0613MD tiene un par de aguante de 6,5 kg.cm con 6V y 7,5 kg.cm a los 7.4V
- Los RS1513MD tienen un par de aguante de 14,4 kg.cm con 6V y 16,5 a los 7.4V

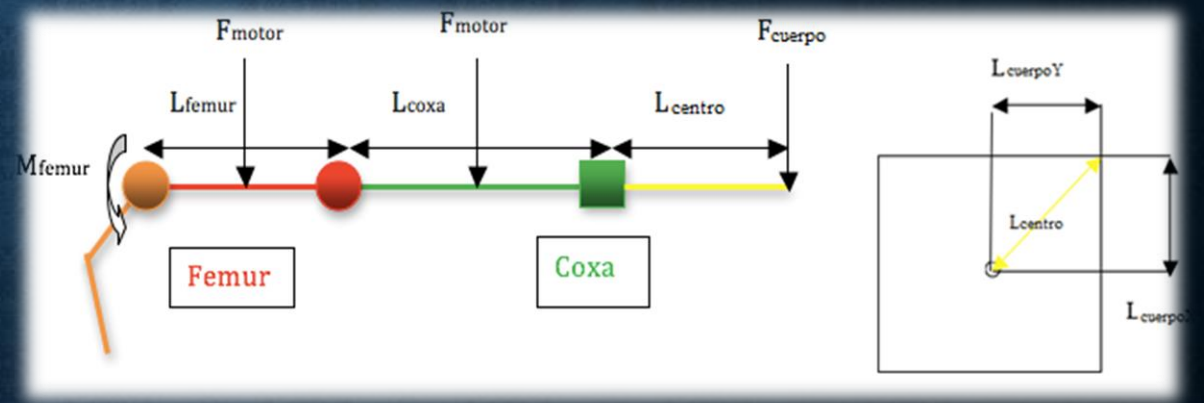
Se realizo entonces un estudio del par con las dimensiones de las patas de nuestros robots, ya que una sobrecarga de torque podría dar problemas de movilidad o dañar los servos.

- El estudio se hizo sobre el peor de los casos, es decir, se utilizo la posición de la pierna donde la distancia desde el centro del cuerpo hasta la ultima articulación (fémur, tibia) fuera máxima .



$$M_{femur} = F_{motor} \cdot \frac{L_{coxa}}{2} + F_{body} \cdot (L_{coxa} + L_{center}) = 5.4 \text{ kg} \cdot \text{cm}$$

RS0613MD



$$M_{coxa} = F_{motor} \cdot \left(\frac{L_{coxa}}{2} + \frac{3 \cdot L_{femur}}{2} \right) + F_{body} \cdot (L_{coxa} + L_{center} + L_{femur}) = 10 \text{ kg} \cdot \text{cm}$$

RS1513MD

CONEXIONES Y PROGRAMACIÓN CON CUBEMX

La velocidad de reloj de la STM32F3Discovery es de 48MHz

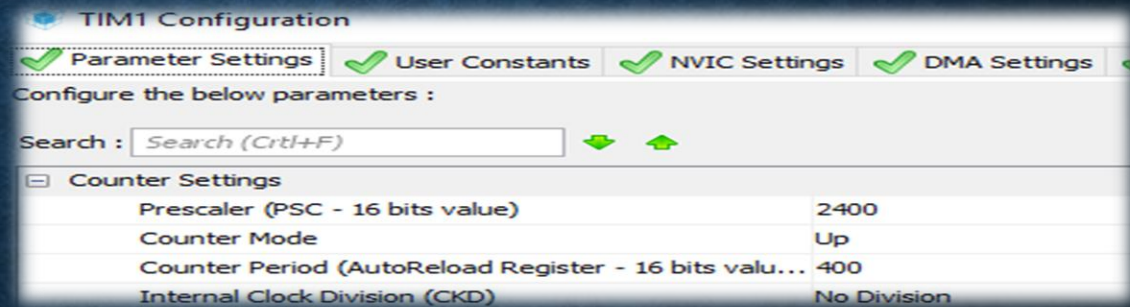
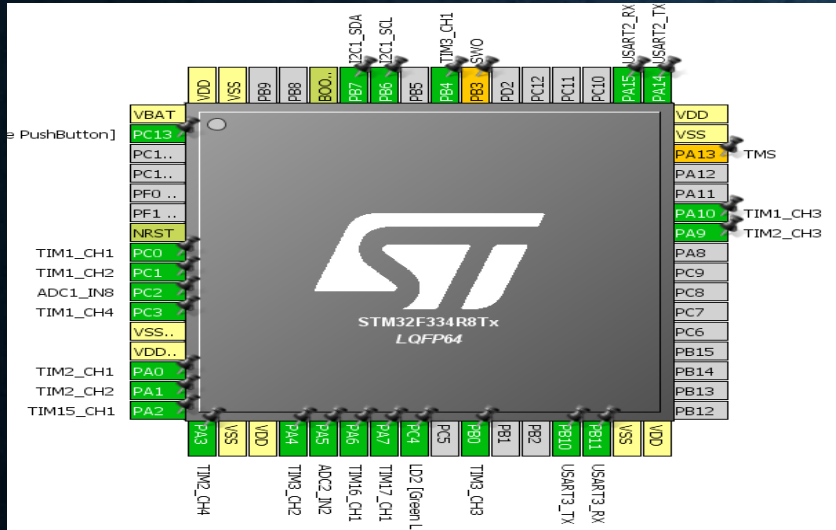
Los servomotores se controlan mediante una señal PWM modulación de ancho de pulso

Se envía una señal pulsada de 50Hz al servomotor.

$$\frac{48000000}{50\text{Hz}} = 960000 = \text{prescaler} \cdot \text{periodo de contador}$$

Por lo tanto se tendrá un prescaler de 16bits y un período de contador para cambiar la señal

El prescaler y el periodo de contador de los TIMERS que enviara estas señales PWM se pueden configurar en CUBEMX.



En CUBEMX también se pueden configurar la velocidades de las UART'S

Generación de archivos .c y .h con la programación en código.

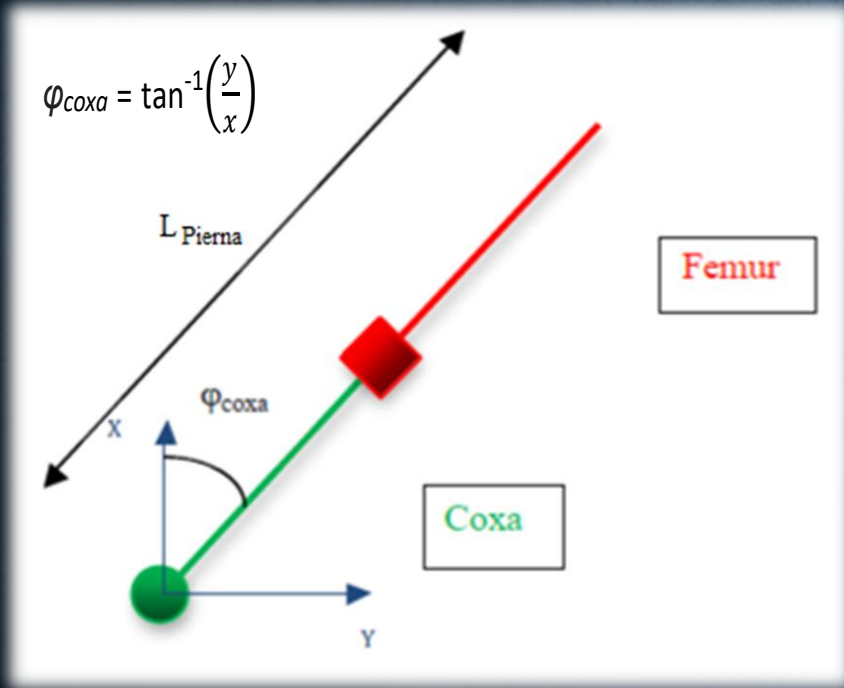
Posteriormente abriremos estos archivos para seguir con μVisión5.

CINEMÁTICA

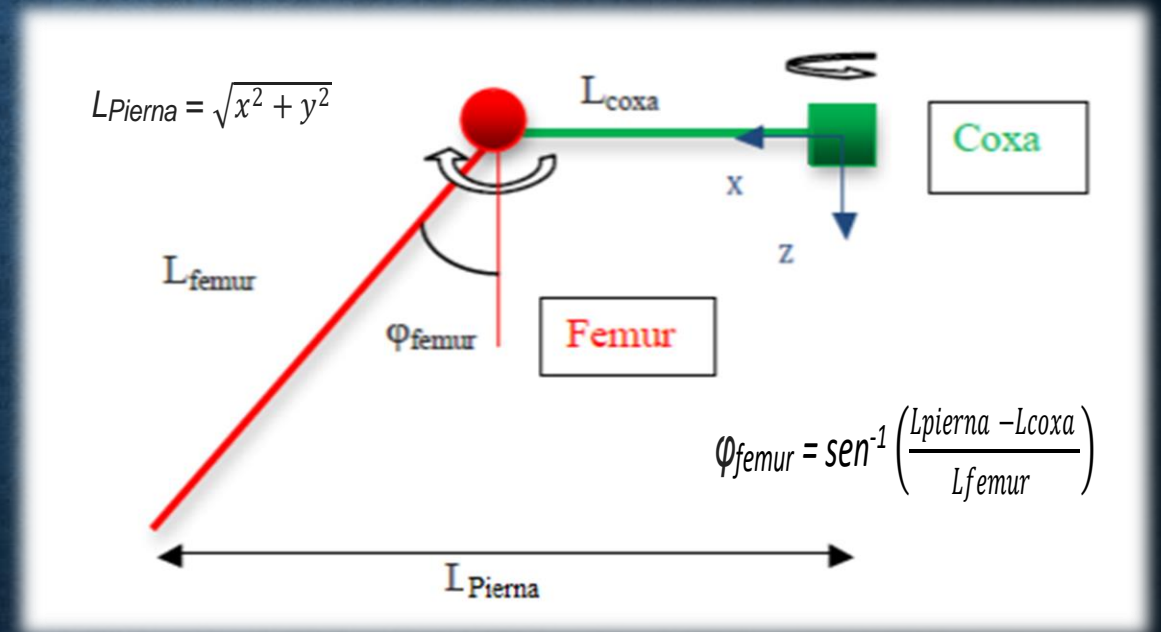
Se necesitó distinguir la cinemática inversa y directa:

- La cinemática de avance calcula las posiciones dadas los ángulos de articulación.
- La cinemática inversa calcula los ángulos de articulación dados las posiciones.

Cinemática inversa de las piernas del robot 2GDL



Para el calculo del φ_{coxa} se debe prestar atención al cuadrante en el que se encuentra la pierna.



La C. Inversa de las piernas del robot de 3GDL será mas compleja ya que tendrá también un φ_{tibia}

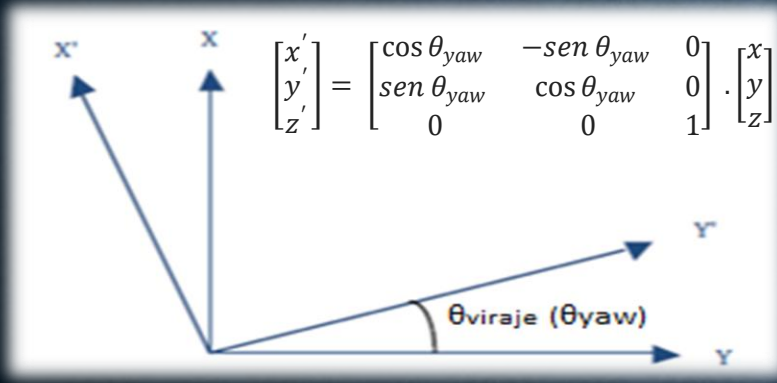
CINEMÁTICA

Se podría ver la cinemática inversa del cuerpo como el pegamento que une las piernas y hace que el cuerpo se mueva en cierta posición.

Es importante decir que se ha utilizado un sistema de coordenadas para el cuerpo y otro para cada pata.

Cinemática inversa del cuerpo del robot 2GDL y del 3GDL

El cuerpo 2GDL solo tendrá rotación en el plano XY, alrededor del eje Z - Yaw - Viraje :



El cuerpo del robot 3GDL si que podrá rotar en los tres planos.

Rotación Plano XZ, alrededor del eje Y - Pitch -Inclinación

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta_{pitch} & 0 & \text{sen } \theta_{pitch} \\ 0 & 1 & 0 \\ -\text{sen } \theta_{pitch} & 0 & \cos \theta_{pitch} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Rotación Plano YZ, alrededor del eje X - Roll - Balanceo

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{roll} & -\text{sen } \theta_{roll} \\ 0 & \text{sen } \theta_{roll} & \cos \theta_{roll} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

La matriz de rotación total seria la multiplicación de estas 3.

$$M_{totR} = M_{yaw} \cdot M_{pitch} \cdot M_{roll}$$

$$\begin{bmatrix} \cos \theta_{pitch} \cos \theta_{yaw} & \cos \theta_{yaw} \text{sen } \theta_{roll} \text{sen } \theta_{pitch} - \text{sen } \theta_{yaw} \cos \theta_{roll} & \cos \theta_{roll} \cos \theta_{yaw} \text{sen } \theta_{pitch} + \text{sen } \theta_{roll} \text{sen } \theta_{yaw} \\ \cos \theta_{pitch} \text{sen } \theta_{yaw} & \cos \theta_{roll} \cos \theta_{yaw} + \text{sen } \theta_{roll} \text{sen } \theta_{pitch} \text{sen } \theta_{yaw} & -\cos \theta_{yaw} \text{sen } \theta_{roll} + \cos \theta_{roll} \theta_{roll} \text{sen } \theta_{pitch} \text{sen } \theta_{yaw} \\ -\text{sen } \theta_{pitch} & \cos \theta_{pitch} \text{sen } \theta_{roll} & \cos \theta_{roll} \cos \theta_{pitch} \end{bmatrix}$$

CALIBRACIÓN DE SERVOMOTORES

Con la C.inversa se calculan los ángulos que deben girar los servomotores para llegar a cierta posición.

Ahora bien, los servomotores necesitan como entrada un Servo-valor y no un ángulo para girar.

Por lo tanto se requiere una transformación de estos ángulos en Servo-valores.

Los servomotores se controlan mediante una señal PWM modulación de ancho de pulso PWM500-2500us 270°

Según esta señal pulsada que varia entre 0,5 y 2,5 milisegundos, los servos tomaran una posición u otra.

$$\frac{48000000}{50Hz} = 960000 = \text{prescaler} \cdot \text{periodo de contador}$$

En el R. 2GDL se tomo un valor de prescaler de 2400 y Periodo de contador de 400. $2400 \times 400 = 960000$.
 $960000 / 48Mhz =$ la señal de 50Hz(20ms). Cada 20ms el contador cuenta 400.

Si se compara el periodo del contador con un servo valor de 10. Tenemos : $20ms \times 10 / 400 = 0,5ms$.

Con un servo-valor de 10, el ángulo del servo estará en uno de sus extremos (a 0 °) ya que va de 0,5 y 2,5ms.

Con uno de 51, obtendríamos $20ms \times 51 / 400 = 2,5ms$, es decir los 270°, el otro extremo.

De todo esto se deducen 2 cosas:

1. El rango de servo-valores ira de 9 o 10 a 51 o 52 aprox.
2. Cuanto mayor es el Periodo de contador mayor será el rango de servo-valores y mayor exactitud tendrán los servos para alcanzar el ángulo deseado. Aumentarlo se hizo en el R.3GDL



CALIBRACIÓN DE SERVOMOTORES

Ahora si podemos hacer fácilmente la calibración de cada servo. Se cogerán 2 medidas:

- Cuando el brazo del servo forma un 90° el servo-valor dado es de 30
- Para 180° el servo-valor es de 44

Por lo tanto con la ecuación de una Recta $y = mx + b$

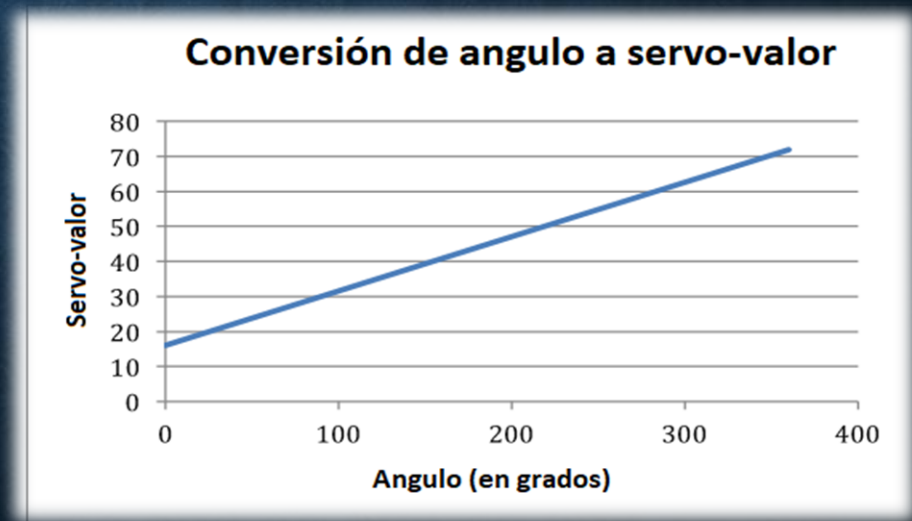
se obtendrá la pendiente $m = \frac{44-30}{\pi - \frac{\pi}{2}} = 8,9126$

Y el intercepto con el Eje Y será $b = 44 - 8,9126 \cdot \pi = 16$



↓

Servo valor = $8.912\varphi + 16$



SENSOR IMU

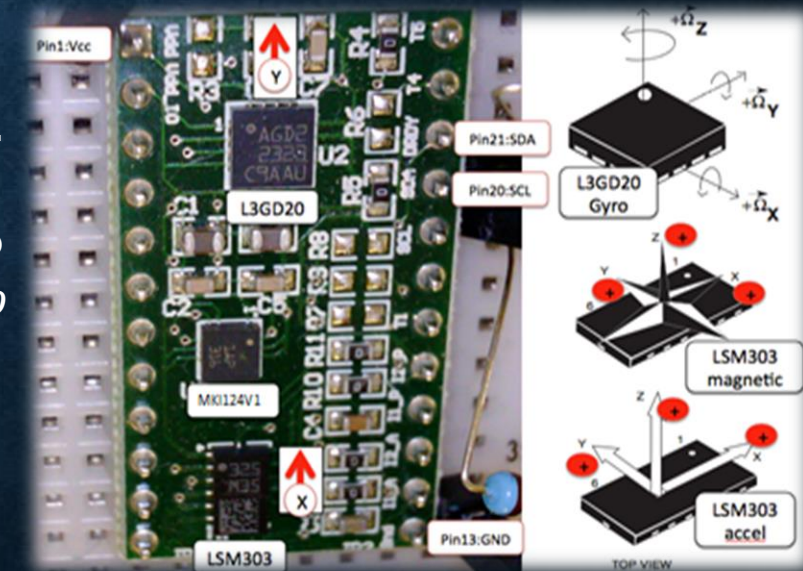
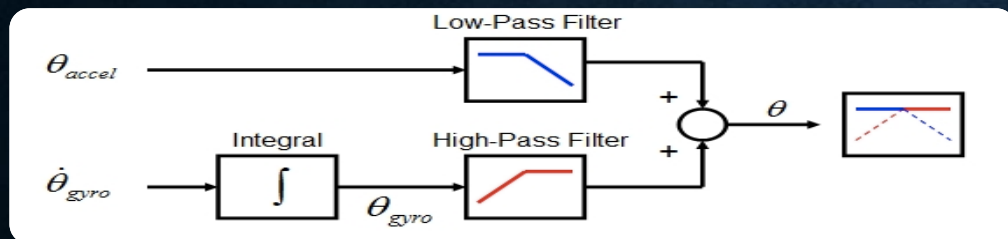
Con la idea de seguir evolucionando al cuadrúpedo se utilizó un sensor IMU en el cuadrúpedo 3GDL para estabilizarlo en terrenos accidentados. (Esta no era la mejor opción)

- La IMU detecta cuando el cuerpo del cuadrúpedo no es horizontal.
- Cuando esto sucede, la IMU dará los ángulos de inclinación y de balanceo.
- Estos ángulos serán utilizados inversamente en la C. Inv. del cuerpo.
- De esta forma, el cuerpo volverá a ser horizontal y permanecerá estable.

La conexión de la IMU se configuro por I2C con Acel y Giros como esclavos.

La IMU utilizada era la placa MEMS STM STEVAL MKI124V1

- Va conectada a la JDMR0617 preparada con un Socket24p y RPUP y RPD.
- Cuenta con un Acelerómetro lsm303 y un giróscopo l30gd20.
- Para la combinación de los dos se implemento un filtro complementario
- $angulo = a \cdot (angulo + ProporciónGiros \cdot dt) + (1 - a) \cdot anguloAcelero$
- Después de varios experimentos se eligió el valor de 'a' = 0.95.



COMUNICACIÓN INALÁMBRICA MANDO - ROBOT

Se utilizaron 2 módulos de Radiofrecuencia STM SP1ML.

La transmisión entre los micros SP1ML y STM32 fue por UART 115200bit/s.

La configuración de los SP1ML fue mediante comandos AT con 'TERMITE3.3' y un cable USB-Serial

1. El registro s00 BAUD_RATE debía ser el mismo en los 2 módulos. También se puso a 115200.
2. Se necesitaba conocer la direcciones de los 2 módulos. El del Robot 0x01, el del Mando 0x20.
3. Por último, la dirección de destino de uno módulo debía ser justo la del otro y viceversa.

La DESTINATION_ADDR del robot seria 0x20 y la del Mando 0x01.

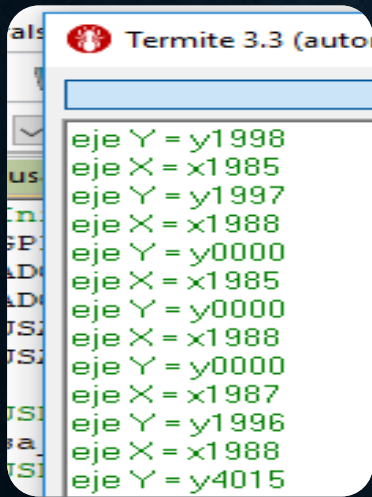


JOYSTICK en el Mando

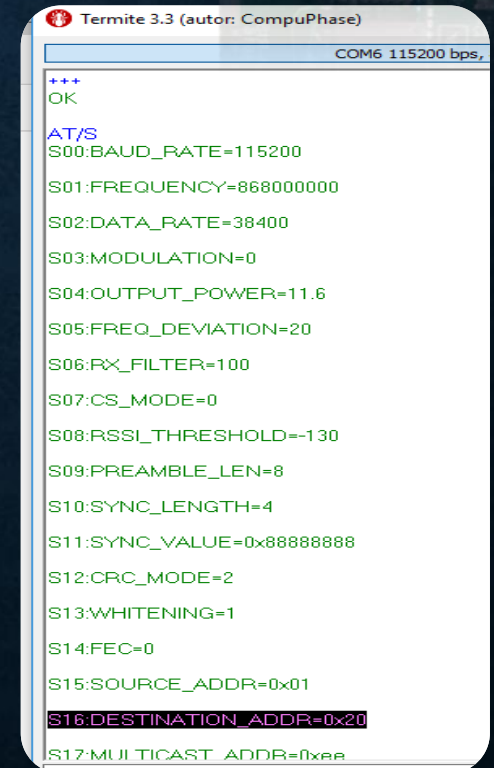
El joystick se puede ver como 2 potenciómetros, uno en el eje X y uno en el Y.

Utilizando un ADC de 12 bits el valor obtenido por el Joystick será entre 0 y $2^{12}-1=4095$

Esta será la información que el SP1ML n°20 le envié al SP1MLn°01



```
Termito 3.3 (autor: CompuPhase)
COM6 115200 bps,
+++
OK
AT/S
S00:BAUD_RATE=115200
S01:FREQUENCY=868000000
S02:DATA_RATE=38400
S03:MODULATION=0
S04:OUTPUT_POWER=11.6
S05:FREQ_DEVIATION=20
S06:RX_FILTER=100
S07:CS_MODE=0
S08:RSSI_THRESHOLD=-130
S09:PREAMBLE_LEN=8
S10:SYNC_LENGTH=4
S11:SYNC_VALUE=0x88888888
S12:CRC_MODE=2
S13:WHITENING=1
S14:FEC=0
S15:SOURCE_ADDR=0x01
S16:DESTINATION_ADDR=0x20
S17:MUI_TICAST_ADDR=flxxx
eje Y = y1998
eje X = x1985
eje Y = y1997
eje X = x1988
eje Y = y0000
eje X = x1985
eje Y = y0000
eje X = x1988
eje Y = y0000
eje X = x1987
eje Y = y1996
eje X = x1988
eje Y = y4015
```



```
Termito 3.3 (autor: CompuPhase)
COM6 115200 bps,
+++
OK
AT/S
S00:BAUD_RATE=115200
S01:FREQUENCY=868000000
S02:DATA_RATE=38400
S03:MODULATION=0
S04:OUTPUT_POWER=11.6
S05:FREQ_DEVIATION=20
S06:RX_FILTER=100
S07:CS_MODE=0
S08:RSSI_THRESHOLD=-130
S09:PREAMBLE_LEN=8
S10:SYNC_LENGTH=4
S11:SYNC_VALUE=0x88888888
S12:CRC_MODE=2
S13:WHITENING=1
S14:FEC=0
S15:SOURCE_ADDR=0x01
S16:DESTINATION_ADDR=0x20
S17:MUI_TICAST_ADDR=flxxx
```

COMUNICACIÓN INALÁMBRICA MANDO - ROBOT

La velocidad de comunicación entre Mando y Robot fue la misma para el 2GDL y el 3GDL. De 115200

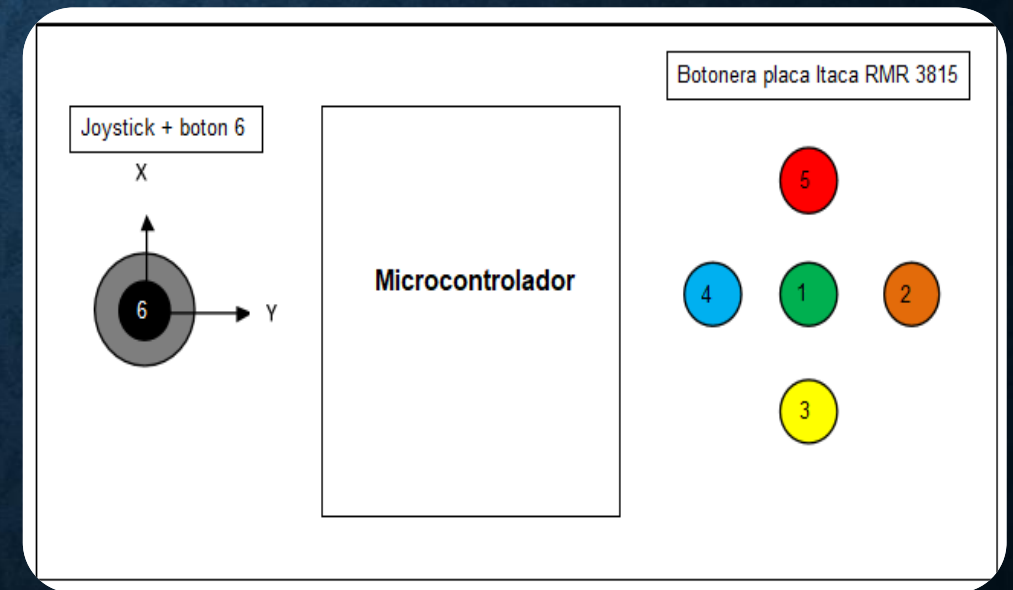
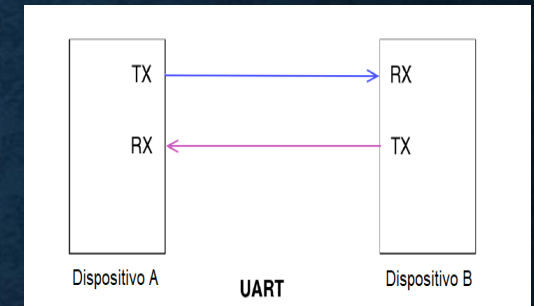
La comunicación inalámbrica del R. 2GDL consistió únicamente en el joystick inalámbrico.

La comunicación del cuadrúpedo 3GDL se actualizó el Mando con 6 botones:

- 5 botones de la botonera de la placa ITACA RMR3815.
- 1 botón implementado en el propio Joystick.

Estos botones se utilizaron para darle al robot unas funcionalidades concretas:

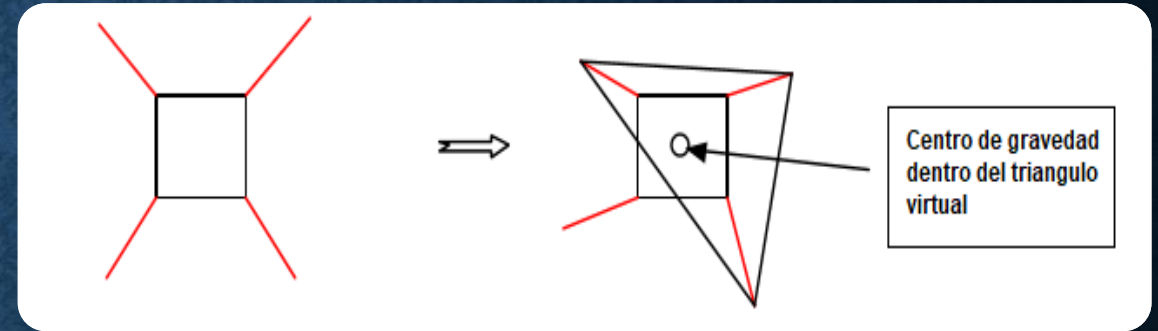
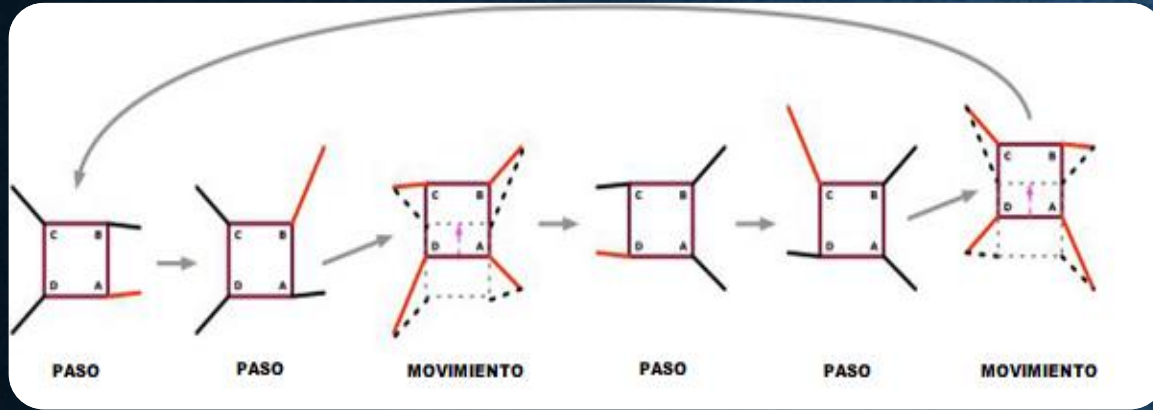
1. **Botón verde:** Ir a la posición de almacenamiento o guardado del robot.
2. **Botón marrón:** Mover el robot en la dirección Z y rotar alrededor del eje Z (quedándose el cuerpo y patas en el mismo punto).
3. **Botón amarillo:** Rotar alrededor de los ejes X e Y (balanceo-roll e inclinación-pitch).
4. **Botón azul:** Hacer que el cuerpo se mueva en la dirección X e Y con las piernas en la misma posición.
5. **Botón rojo:** Hacer que el robot camine en la dirección X y pueda girar.
6. **Botón negro del Joystick:** Hacer que el robot camine en dirección X e Y.



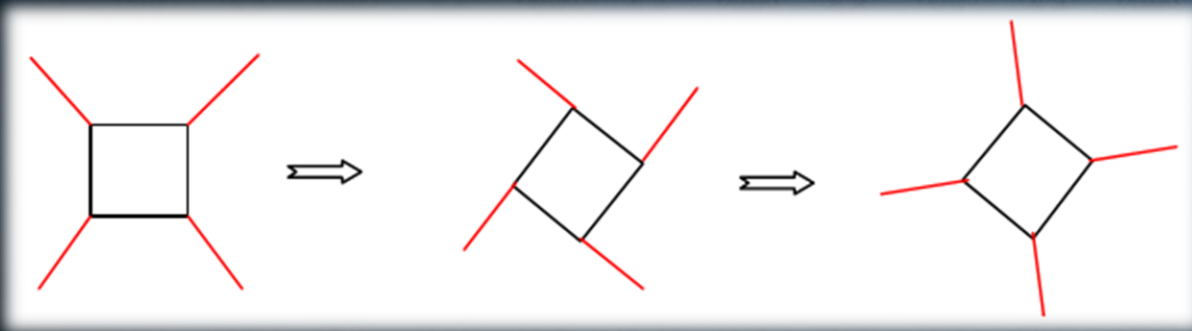
FORMAS DE DAR LOS PASOS

Hay varias formas de dar los pasos las mas comunes son la marcha de trote y la de gatear.

Marcha para caminar del R. 2GDL fue la de deslizamiento o gateo:



La forma de giro se realizaría girando el cuerpo primero y luego girando las patas una por una.

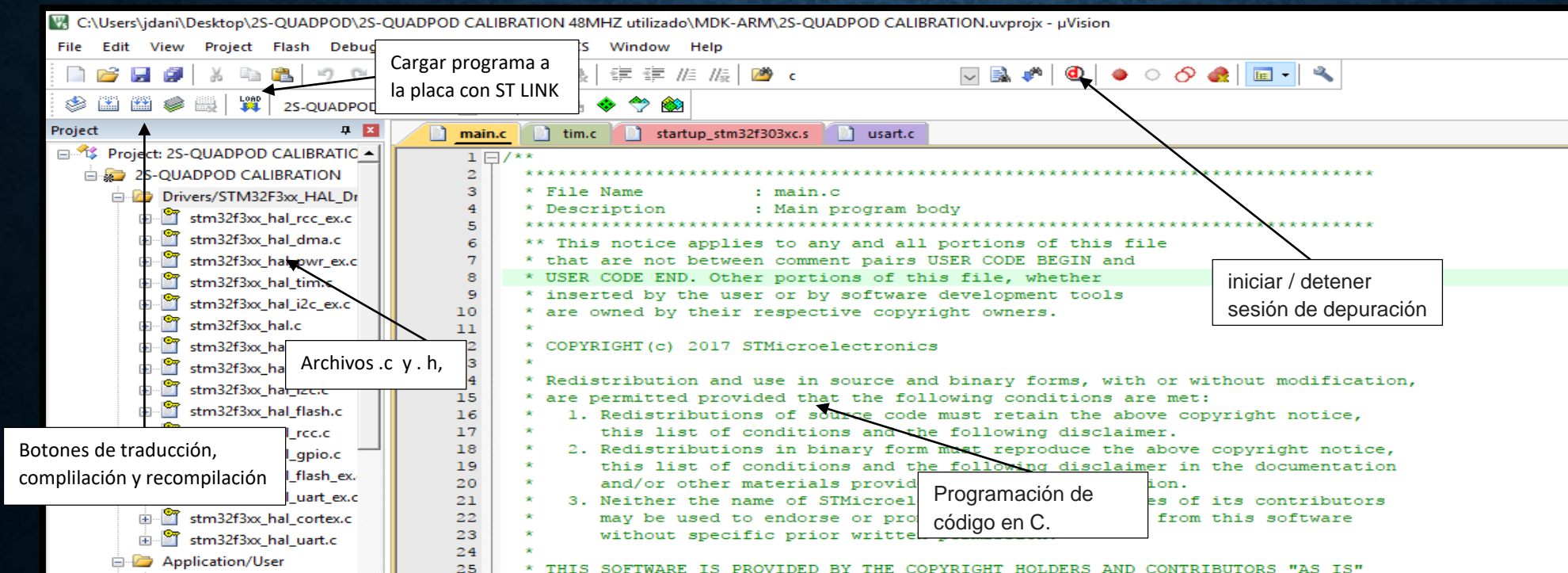


PROGRAMACIÓN CON MICROVISION5

Antes de empezar a programar, primero se tuvo que repasar cómo programar en C.

Se utilizó μ Vision5 de Keil, para añadir todo el código de programación al robot y al mando:

- Toda la parte de C.INVERSA
- las diferentes formas de dar los pasos y los giros.
- La calibración de los servos de los sensores de la IMU
- El Medidor de distancia con su ADC.
- Configuración del joystick y la botonera para el mando etc....



ESTIMACIONES Y CONCLUSIONES

Como futuro ingeniero industrial con una especialización en Electromecánica se ha aprendido mucho sobre Electrónica.

Se ha adquirido un gran conocimiento previo sobre la electrónica y la tecnología de la información y las comunicaciones TIC.

Se ha adquirido mucho manejo para el diseño de piezas mecánicas...etc.

El robot 3GDL funciona bien, pero está claro que se podrían hacer muchas más mejoras:

- Para evitar que una pierna se quede en el aire, por ejemplo, los sensores de presión podrían ser implementados en los pies del cuadrúpedo.
- Dotar al robot de un brazo con pinzas para poder coger objetos también sería buena mejora.

La IMU tampoco funciona perfectamente, solo cuando se opera con el robot en estático nada más empezar, reacciona bien a los cambios de altura en el suelo y logra equilibrar el cuerpo.

- La combinación con una cámara o la implementación de un filtro Kalman sería una mejora.

La programación en C tampoco es buena y hay estructuras que se podrían escribir de mejor forma.

Las cosas buenas podrían ser: el diseño mecánico simétrico y resistente y el diseño electrónico con PCB's.

**MUCHAS GRACIAS POR VUESTRA
ATENCIÓN 😊**