



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DISEÑO DE UN SISTEMA ELECTRÓNICO INALÁMBRICO CENTRALIZADO PARA LA MEDIDA DE UN CONJUNTO DE BARRERAS LUMINOSAS

AUTOR: RAFAEL GÓMEZ HORTELANO

TUTOR: JULIO GOMIS-TENA DOLZ

Curso Académico: 2017-18

AGRADECIMIENTOS

A mi madre y a mi padre, por su cariño, su apoyo y su inquebrantable confianza

A mis hermanos y amigos por comprender mis ausencias

Y en especial, a Julio, mi director de proyecto, por haberme dado esta oportunidad y por responder siempre a mis dudas de forma eficiente y amable

Índice

Memoria descriptiva	2
1. Objetivo y justificación del proyecto.....	3
I. Antecedentes.....	3
II. Objeto del proyecto.....	3
III. Marco actual.....	4
IV. Técnicas existentes para la detección de movimiento.....	6
2. Herramientas.....	8
3. Proceso proyectual/Proyecto.....	15
I. Descripción de la solución adoptada.....	15
II. Diseño del circuito electrónico.....	15
➤ Elementos principales o activos.....	17
➤ Elementos auxiliares o pasivos.....	21
➤ Otros elementos.....	21
III. Programación del sistema.....	22
IV. Montaje del conjunto.....	30
4. Pruebas y resultados.....	33
5. Conclusión y líneas futuras.....	38
Presupuesto	40
Pliego de condiciones	43
Bibliografía	45
Manual de usuario	46
Anexos	47

MEMORIA DESCRIPTIVA

Memoria descriptiva

1. Objetivo y justificación del proyecto

I. Antecedentes

El Departamento de Ingeniería Electrónica y el Departamento de Ciencia Animal están colaborando en una investigación sobre pequeños rumiantes. En dicha investigación, mediante un sistema complejo, se pretende monitorizar, transmitir y almacenar una serie de magnitudes. Actualmente se están realizando estudios de calorimetría indirecta sobre los sujetos para la obtención de gasto metabólico.

En esta misma línea de trabajo, se decide introducir nuevas magnitudes a monitorizar: la posición (tumbado o erguido) y el instante de ingesta de alimentos (acceso a los comederos) de animales de granja, además de registrar los datos para su posterior análisis.

El diseño e implementación de este nuevo avance en la investigación constituirá el documento formal "Trabajo Fin de Grado" (TFG) con el cual el autor aspira a la consecución del Título Universitario correspondiente al Grado de Ingeniería en Tecnologías Industriales.

II. Objeto del proyecto

El objeto del proyecto es diseñar un sistema capaz de detectar la apertura y cierre de haces de luz (barreras luminosas) colocados en el interior de un recinto cerrado y transmitir esta información de manera inalámbrica.

Este sistema contará con un circuito electrónico basado en un microcontrolador y en un módulo Bluetooth que recogerá la información de cada detector luminoso de infrarrojos. Ésta será enviada a una central de recogida de datos, que ya se encuentra implementada previamente, para su almacenamiento y estudios posteriores. Habrá tantos circuitos como barreras luminosas, por lo que, el sistema, será escalable a las necesidades del recinto.

En este proyecto se diseñará el circuito electrónico y se programará el mismo para conseguir estos objetivos.

III. Marco actual

Este proyecto, como ya se ha comentado, es una parte dentro de un proyecto de investigación más completo. Con motivo de fundamentar algunas de las decisiones tomadas durante el proceso proyectual, se explicarán brevemente las partes de las que se compone el sistema mayor y el funcionamiento de este.

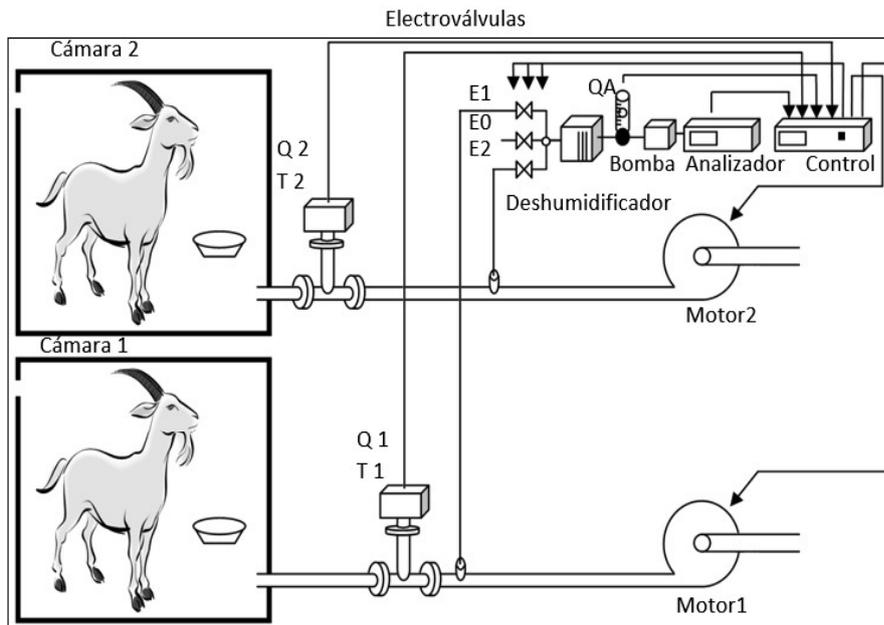


Figura 1 – Diagrama del sistema mayor

El proyecto madre consiste en un equipo que deberá monitorizar y almacenar los valores de caudales y temperaturas (Q1, Q2, T1, T2). Actuará sobre los motores (Motor 1 y Motor 2) fijando de manera dinámica la velocidad de giro para conseguir una renovación de aire constante en los 2 sistemas, que sea la programada, así como el flujo definido de entrada en el analizador. Actuará sobre las 3 electroválvulas almacenando los valores correspondientes a la Cámara 1 (E1), Cámara 2 (E2) y el ambiente (E0), permitiendo realizar en cada medida un valor (blanco) que sirva para ajustar la línea base en cada una de ellas. Los datos serán almacenados en una tarjeta de memoria μ SD en un formato de texto de manera tal que permita el procesamiento mediante programas como Microsoft Excel.

Los parámetros de funcionamiento (caudales Q1, Q2, QA, periodos de muestreo,...) serán programados de manera previa a la toma de datos y la adquisición almacenará los valores de fecha y hora, la tensión de la red de alimentación del analizador, las temperaturas, los caudales, los contenidos de CO₂, O₂ y CH₄ de las 2 cámaras y el ambiente (más adelante, se valorará la posibilidad de añadir medidas adicionales como los datos del peso entregados por unas balanzas electrónicas sobre las que se apoyarán los comederos, o magnitudes como ritmo cardíaco obtenidas por otros dispositivos).

La manera de proceder del usuario sería la siguiente: en cada toma de datos prepara una tarjeta de memoria limpia de datos, que incluye un fichero de texto en el que, con un formato establecido, indica los parámetros de funcionamiento. Introduce la tarjeta en el equipo, lo pone en funcionamiento, el equipo lee los parámetros de funcionamiento de la tarjeta, los indica en un visualizador y espera que el usuario confirme con la pulsación de una tecla (verde 1). Tras la confirmación, se pone en funcionamiento y continúa indefinidamente hasta que el usuario termine la adquisición con la pulsación de otra tecla (rojo 0). En ese momento, el usuario puede apagar el equipo y retirar la tarjeta para llevársela y procesar la información.

En el caso de que exista un corte de red, se apagarán las bombas centrífugas, permaneciendo en funcionamiento el analizador y el equipo de control, con objeto de mantener la estabilización de medida del analizador y continuar con el control del sistema (mediante SAI estabilizado). Si el corte de red permanece en el tiempo, el sistema por completo entrará en suspensión a la espera de una recuperación del suministro eléctrico (se sugiere un mecanismo de seguridad que permita la ventilación natural de las cámaras en ausencia de red eléctrica –y, por lo tanto, de renovación forzada del aire– para evitar la asfixia de los sujetos del estudio).

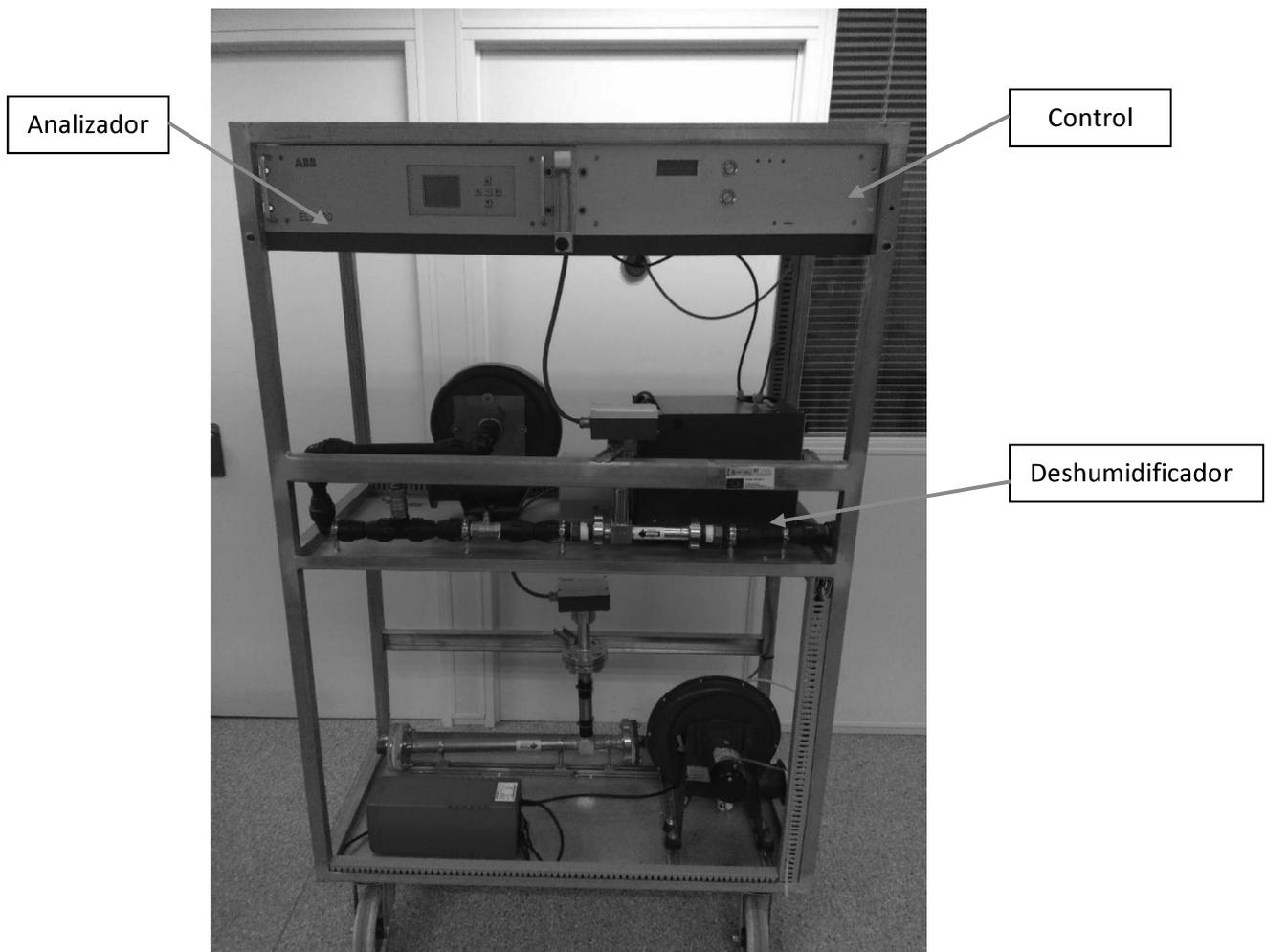


Figura 2 – Fotografía del sistema mayor

IV. Técnicas existentes para la detección del movimiento

Antes de explicar la propuesta, conviene repasar los sistemas que hay actualmente para conseguir funciones parecidas a las presentadas en el objetivo.

La detección del movimiento o detección de presencia en un área exterior o interior es algo que se ha estudiado bastante en los últimos años y es muy solicitada en distintos ámbitos. El ejemplo más habitual es el de detectar personal no autorizado en un local y activar una alarma para avisar de ello.

No obstante, para llevar a cabo esta tarea se pueden emplear diversas técnicas tanto electrónicas como mecánicas. Nos centraremos en analizar las electrónicas pues es el ámbito del proyecto.

Tenemos 3 métodos o sistemas: la detección por infrarrojos, la detección por láser y la detección por radar.

La detección por infrarrojos consiste en medir una variación del calor (radiación infrarroja) emitida por el ambiente y detectada en un sensor de infrarrojos. Así pues, si se detecta alguna anomalía, ésta es comunicada inmediatamente y se pueden tomar las medidas necesarias.

El sensor utilizado en este caso suele ser semiesférico de manera que la zona que cubre sea máxima y se sitúa en alguna zona plana (el techo, por ejemplo). Esto es útil si se prevé que, por defecto, no va a haber ningún movimiento en la zona, de manera que, si se detecta algo, es importante comprobarlo. No obstante, si lo que se quiere es utilizarlo para controlar el movimiento concreto de algo en medio de otros posibles movimientos también detectables por el sensor, no es muy útil pues no es capaz de diferenciar el origen.

En el caso de la detección por láser, el sistema dispone de un entramado de haces de luz láser. En el caso de que se rompa un haz y por tanto no llegue al receptor, se manda una señal o se activa otro sistema.

Aquí el receptor suele ser el mismo dispositivo que envió la señal luminosa que, reflejada en un espejo, vuelve al origen confirmando que no hay nada en el camino. Esto reduce el número de dispositivos activos que hay que controlar.

Sin embargo, el problema de este sistema es la distancia entre el emisor y la superficie reflectante. Si esta es muy corta, podría darse el caso de que, aunque el haz sí se hubiera cortado, por las dimensiones de este, acabara volviendo parte y diera una falsa medida.

Por último, el radar, funciona enviando ondas en un ángulo sólido determinado (parecido al sistema del sensor de infrarrojos) y recibéndolas de nuevo pasado un tiempo. El sistema calcula ese tiempo de regreso y, si es inferior o superior a un valor, determina que algo (un ser vivo o un objeto) ha hecho rebotar las ondas antes de lo previsto o que las ha ralentizado.

En el caso de los humanos, para el sistema simulamos básicamente “bolsas de agua” que modifican la velocidad de esas ondas y por tanto es fácil detectar la presencia de alguien. Sin embargo, tenemos el mismo problema que con los infrarrojos: no podemos identificar con claridad la fuente de la anomalía.

2.Herramientas

Antes de proceder con la solución adoptada conviene describir las herramientas (software o maquinaria) que se van a usar a lo largo del proyecto en las diferentes fases de este.

Eagle

Eagle (Easily Applicable Graphical Layout Editor) es el programa que se ha utilizado a la hora de diseñar la PCB (Printed Board Circuit o placa de circuito impreso).

Este software permite hacer primero un diseño esquemático, disponiendo todos los elementos, sin medidas, y usando conexiones que, todavía, no implican una distribución concreta de las pistas de cobre.

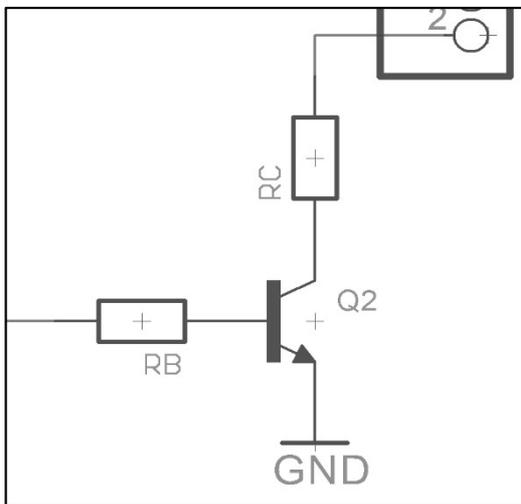


Figura 3 – Ejemplo de esquemático (SCH)

Después, con todos los elementos y sus conexiones dispuestos, se pasa a dibujar las pistas de cobre que conectarán dichos elementos. Es habitual que se puedan usar ambos lados de la placa (conocidos como “top” y “bottom”) para colocar tanto componentes activos como componentes pasivos y, también, se pueden usar “vías” para acceder de un lado al otro, ofreciendo diversas posibilidades a la distribución.

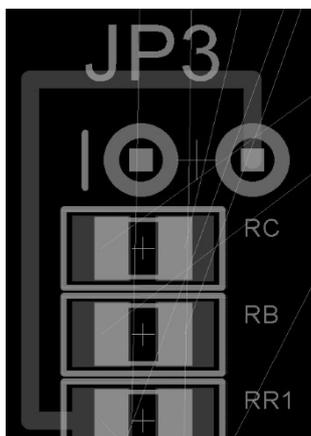


Figura 4 – Ejemplo de generación de BDR (“board”).

Las líneas amarillas finas son conexiones sin realizar mientras que la conexión roja ya es una conexión real. Aquí las medidas ya son reales.

MPLAB IDE, compilador CCS y PicKit 3

Como softwares de programación se han usado el entorno MPLAB IDE en la versión 8.92 y el compilador CCS. Vamos a describir las características de cada uno.

El MPLAB IDE es un software de programación de microcontroladores desarrollado por Microchip y compatible con diferentes compiladores. El entorno trabaja a base de proyectos (una carpeta por proyecto). En esta carpeta deben encontrarse tanto el archivo principal del programa como todos los archivos adicionales de cabecera (".H") que sean necesarios para la compilación. El archivo principal, además, debe ser añadido al proyecto desde el menú superior.

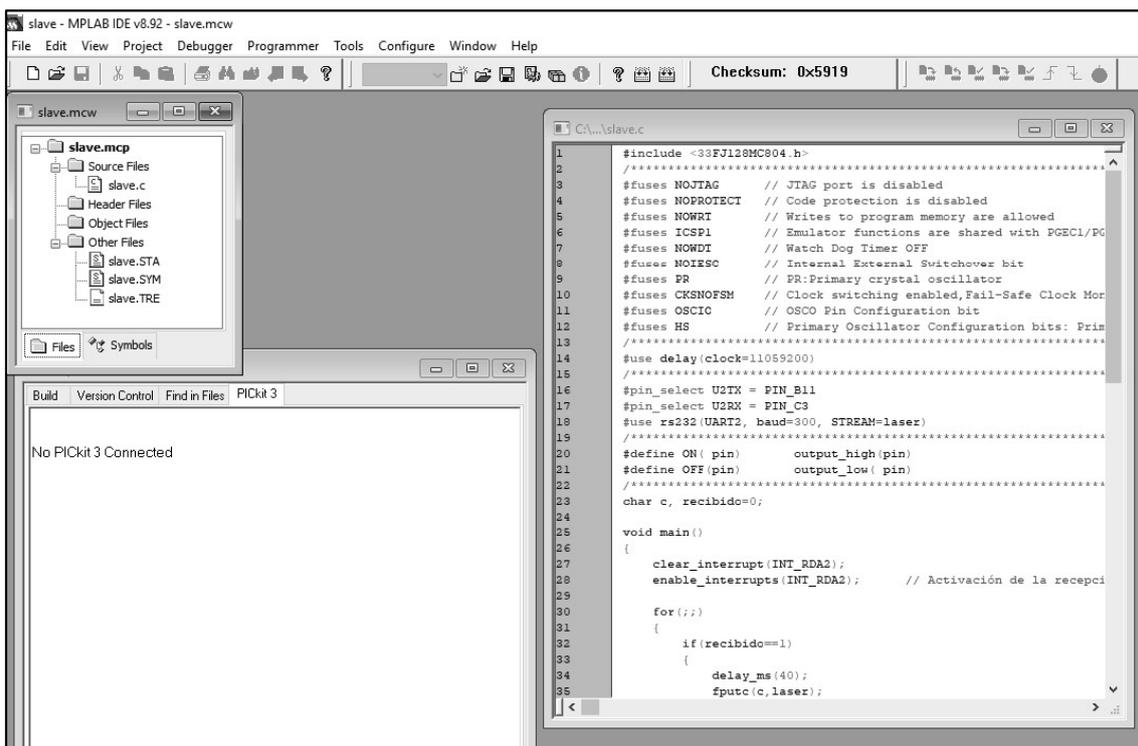


Figura 5 – Ejemplo del entorno

Como el CCS no incluye un método por defecto para enlazar diferentes módulos de un mismo programa ("linker"), además de esos archivos, si se necesitan otros archivos tipo ".C", también deberíamos situarlos en esa carpeta con sus respectivos archivos de cabecera (e incluirlos todos en el programa principal mediante la directiva #include). Es decir, que este compilador no admite programación modular. No obstante, debido a la baja complejidad de los programas que se desarrollarán, esta característica no es importante de cara al proyecto.

En nuestro caso, solo ha hecho falta incluir la librería del microcontrolador para la compilación y, dado que ésta ya viene incorporada en el compilador CCS, dentro de la carpeta del proyecto únicamente está el archivo principal del programa.

Existen maneras de hacer una pseudo-programación modular más fácil a través de directivas específicas, pero, como no se han necesitado, solo se comenta a título informativo.

En cuanto a hardware, hemos utilizado el PicKit 3 y se han habilitado puertos específicos en la PCB para poder conectarlo. Este es un hardware de sencillo manejo, con conexión USB hacia el ordenador y totalmente compatible con el entorno MPLAB tanto para programar como para hacer de “debugger” y comprobar el buen funcionamiento del programa.



Figura 6 – PicKit3 estándar

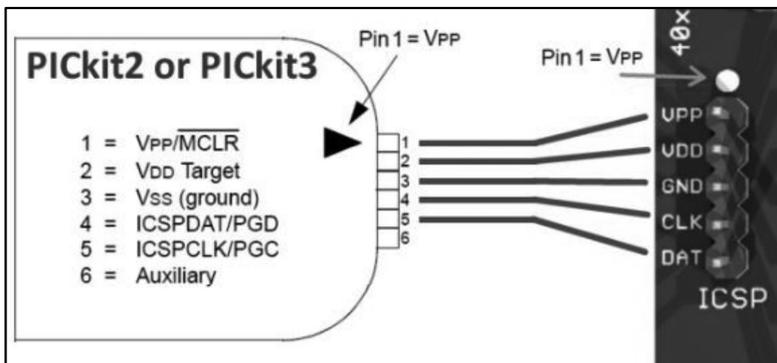


Figura 7 – Ejemplo de conexión

BLE (Bluetooth Low Energy)

Una parte fundamental del proyecto es que la conexión se realiza mediante Bluetooth, por lo tanto, conviene explicar algunas de las peculiaridades del BLE el cual forma parte de la versión 4.0 del “Bluetooth Core Specification”.

Los dos tipos de perfiles que existen en cualquier dispositivo BLE son el GAP (General Access Profile) y el GATT (General Attribute Profile). El primero controla cómo interaccionan los diferentes dispositivos Bluetooth mientras que el segundo controla cómo se intercambia la información (atributos de los dispositivos, paquetes de datos...).

Por la manera en la que se trabajará con los dispositivos, no hará falta concretar qué perfil GATT tienen; únicamente será necesario definir los perfiles GAP.

Existen 4 perfiles tipo GAP: “broadcaster”, “observer”, “central” y “peripheral”. Por parejas, los dos primeros no requieren conexión directa con otro dispositivo, mientras que los dos últimos sí.

El “broadcaster” emite información regularmente en forma de “advertisement packets” (en lugar de paquetes de datos completos) que pueden ser escuchados por otros dispositivos en las cercanías. Un ejemplo podría ser un termómetro que emite la información sobre la temperatura a cualquier dispositivo que quiera escuchar.

El “observer” es el rol opuesto al “broadcaster”. Su función es recoger datos de “advertisement” emitidos por otros dispositivos, lo cual lo hace ideal para funcionar con un “broadcaster”. Un ejemplo de “observer” podría ser el smartphone que recibe lo que emite el termómetro del ejemplo anterior.

El “central” y el “peripheral” están preparados para formar una conexión entre ambos, identificándose primero con “advertisement packets” y posteriormente, para mandar paquetes de datos. Un punto a favor de este tipo de Bluetooth es que permite que un “central” se conecte con varios “peripheral” a la vez y pueda recibir datos de varios a la vez. El “central” es siempre el que inicia la conexión y requiere más CPU para funcionar, mientras que el “peripheral” actúa pasivo hasta que se le requiere para formar conexión y requiere menos potencia de hardware para funcionar.

Qué perfiles/roles y por qué se han utilizado en este proyecto se explicará más tarde en la memoria.

Maquinaria y otras herramientas

Por último, queda describir brevemente la maquinaria utilizada, así como las herramientas de soldadura empleadas y los aparatos de medición y alimentación que han sido necesarios durante las pruebas.

La PCB encargada llega de fábrica sin ningún elemento. Los componentes se han comprado aparte y hay que soldarlos a ella. En nuestro caso se ha empleado una estación de soldadura estándar proporcionada por el Departamento de Electrónica.



Figura 8 – Estación de soldadura del laboratorio (arriba)

Figura 9 – Estación de soldadura en catálogo (abajo)



Por otra parte, para el mecanizado de los agujeros que se necesitarán en la caja se ha utilizado una máquina cortadora por control numérico.



Figura 10 – Colinbus Profiler CNC

Como alimentación durante las pruebas se ha usado una fuente de alimentación como la que se muestra en la figura 11, la cual permite regular la tensión fácilmente y con precisión suficiente.



Figura 11 – Fuente de alimentación

Además, para medir las señales electrónicas se ha utilizado también un osciloscopio (Figura 12).

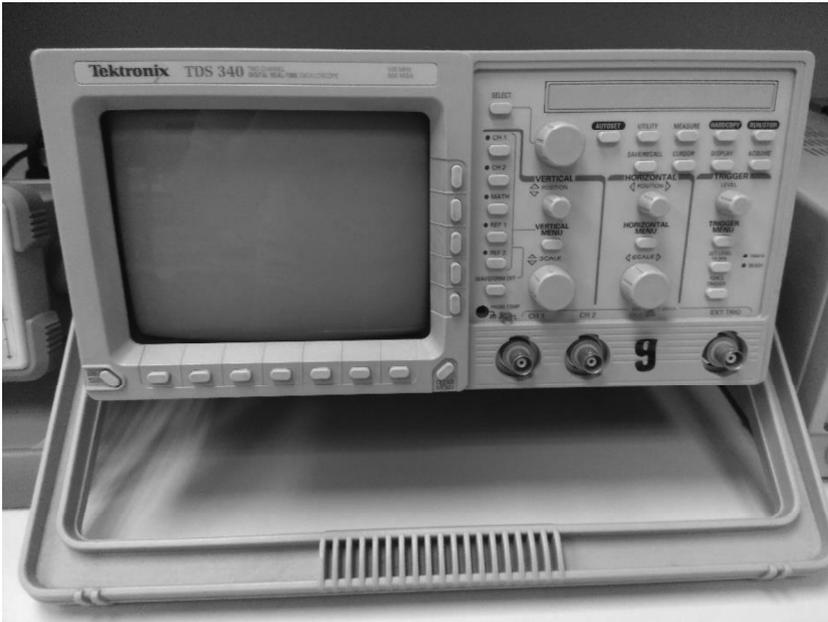


Figura 12 –
Osciloscopio

3. Proceso proyectual

I. Descripción de la solución adoptada

De las técnicas existentes descritas anteriormente, escogemos la detección por láser y, partiendo de ese principio, aplicaremos unas modificaciones para que se adapte mejor a nuestro caso particular.

El sistema propuesto se basa en una serie de barreras luminosas dispuestas en el recinto que el animal cortará con su movimiento habitual. La información de cuándo y en qué parte de la celda se ha cortado (qué barrera se ha cortado) se enviará por comunicación Bluetooth (*BT en siglas*) para disponer de ella en otro dispositivo.

Para evitar el inconveniente que tenía la detección por láser, cada barrera luminosa se compone de dos cajas activas enfrentadas entre sí. Ambas cajas emiten un haz de luz que será recogido por la otra caja. Una de ellas, además, será la encargada de la transmisión de la información por BT.



Dos condiciones de diseño que se quieren conseguir son un tamaño reducido del conjunto final y una autonomía alta en cuanto a lo que consumo de energía se refiere.

La realización de este sistema se ha dividido en dos partes, en primer lugar, el diseño de la placa que contiene el circuito con la elección de sus correspondientes elementos activos y pasivos, y otra de programación del microcontrolador y el BT con ayuda de un software.

II. Diseño del circuito electrónico

El diseño y la distribución de los componentes en la placa deben de estar muy cuidados para cumplir con un tamaño reducido, así como para evitar problemas electrónicos (conexiones indeseadas entre el cobre, por ejemplo).

Por lo tanto, este se deberá adaptar a las especificaciones mínimas necesarias para la fabricación.

El tamaño final de la placa ha sido de 35 x 28 mm.



Normal Condition of PCB capabilities

Modified on: Sat, 31 Jan, 2015 at 3:00 PM

Layers	1 - 4
Material	FR-4
Board Dimension (max)	380mm X380mm
Board Dimension (min)	10mm X10mm
Outline Dimension Accuracy	± 0.2mm
Board Thickness	0.40mm--2.0mm
Board Thickness Tolerance	± 10%
Dielectric Separation thickness	0.075mm--5.00mm
Conductor Width (min)	0.15mm(Recommend>8mil)
Conductor Space (min)	0.15mm(Recommend>8mil)
Outer Conductor thickness	35um
Inner Conductor thickness	17um--100um
Copper to Edge	>0.3mm
Plated Component,Plated via Diameter(Mechanical)	0.3mm--6.30mm
Plated Hole Diameter Tolerance(Mechanical)	0.08mm
Unplated Hole Diameter Tolerance	0.05mm
Hole Space(min)	0.25mm
Hole to Edge	0.4mm
Annular Ring(min)	0.15mm
Aspect Ratio	8:01
Solder Resist Type	Photosensitive ink
Solder Resist Color	Black ,Green, White, Blue ,Yellow
Solder Resist Clearance	0.1mm
Solder Resist Coverage	0.1mm
Plug Hole Diameter	0.3mm--0.65mm
Selective Finish	HASL, ENIG
Silkscreen line width (mim)	6mil

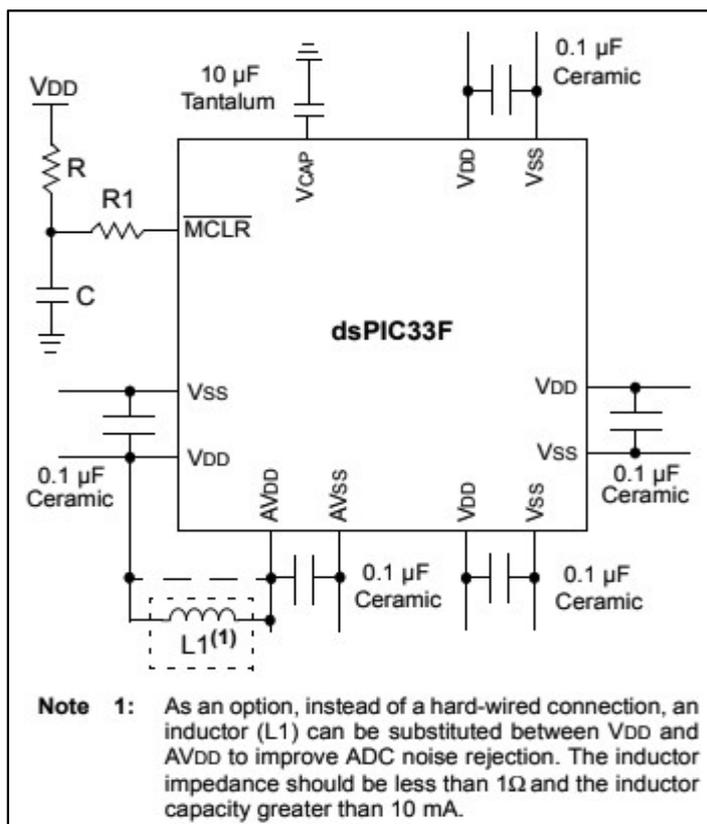
itead.cc



Elementos activos

En esta categoría agrupamos el microcontrolador, el dispositivo Bluetooth, el regulador de tensión, el láser y el receptor utilizados.

El microcontrolador utilizado ha sido el DSPIC33FJ128MC804-PT fabricado por Microchip. Se ha escogido este modelo porque alberga las funcionalidades básicas que se requieren en el proyecto (como puede ser la comunicación UART) y por la experiencia previa que se tiene con el mismo en otros proyectos anteriores.



La conexión de los cables de alimentación y masa, así como la elección de los condensadores de desacoplamiento se ha hecho siguiendo las pautas dadas por el fabricante para un mejor funcionamiento.

Figura 14 -- extraída del manual del fabricante sobre el microcontrolador, muestra los condensadores recomendados para los pines de alimentación

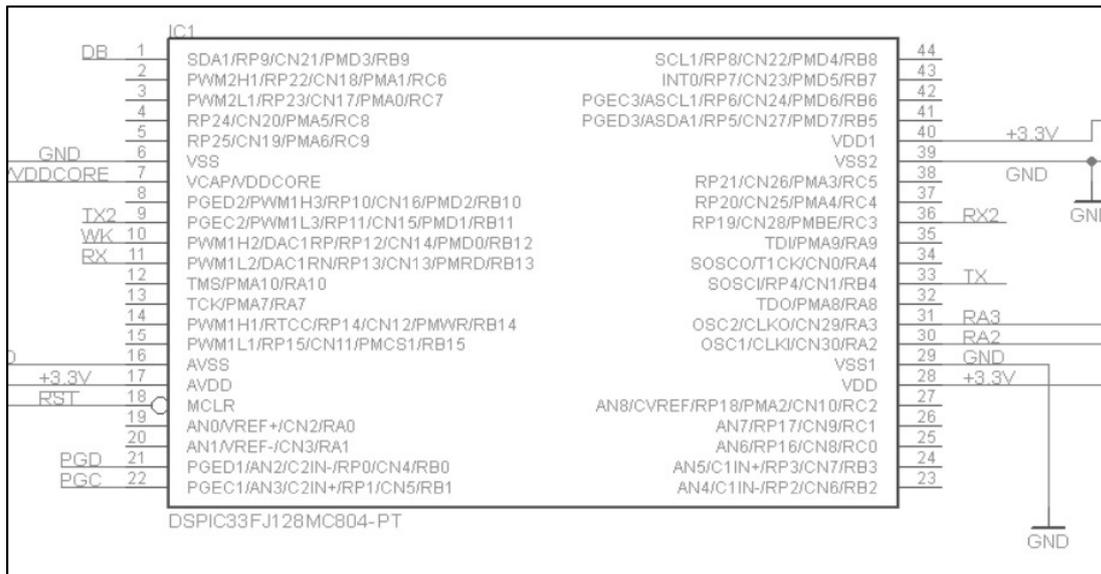


Figura 15 – Esquema en EAGLE de todos los pines y las conexiones del microcontrolador

La lista de pines utilizados con las funciones que tienen y una pequeña descripción es la que sigue:

NUMERO PIN	FUNCION	DESCRIPCION
1 (RB9/RP9)	DB (Debugger)	Pin para hacer pruebas durante la programación
6 (VSS)	GND	Masa
7 (VDDCORE)		-
9 (RB11/RP11)	TX2	Conectado a láser; un nivel alto enciende el láser
10 (RB12/RP12)	WK	Conectado a BT; ¿??
11 (RB13/RP13)	RX	Recepción UART desde BT
16 (AVSS)	GND	Masa
17 (AVDD)	+3,3 V	-
18 (MCLR)	~RST	Pin para resetear el microcontrolador y ayudar en la programación
21 (RB0/RP0/AN2)	PGD	Puerto de programación y posterior entrada analógica
22 (RB1/RP1)	PGC	Puerto de programación
28 (VDD)	+3,3 V	-
29 (VSS1)	GND	Masa
30 (OSC1)	Cristal	Conexión al cristal de 11059200 Hz
31 (OCS2)	Cristal	Conexión al cristal de 11059200 Hz
33 (RB4/RP4)	TX	Transmisión UART hacia BT
36 (RC3/RP21)	RX2	Conectado a receptor; controla las interrupciones
39 (VSS2)	GND	Masa
40 (VDD1)	+3,3 V	-

El modelo de dispositivo Bluetooth es el RN4870-V_RM118, también fabricado por Microchip. Al principio de la fase de diseño se pensó en otros modelos como puede ser el RN42. No obstante, se han desechado debido a que eran más grandes y además no utilizaban la tecnología BLE (Bluetooth Low Energy).

Un detalle importante a la hora de programar es que la versión del Bluetooth en el RN42 es la 2.0, mientras que la del RN4870 es la 4.0, lo cual lo hace diferente para programar y le da más versatilidad de conexión con otros dispositivos.

Durante el diseño de las rutas que deben seguir las conexiones a lo largo de la placa, se ha procurado, entre otras cosas, que estas no pasaran por la zona del Bluetooth que contiene la antena porque, si no, se podría obstruir su comunicación y provocar un mal funcionamiento. En la Figura 16 se muestra una imagen con la recomendación del fabricante al respecto.

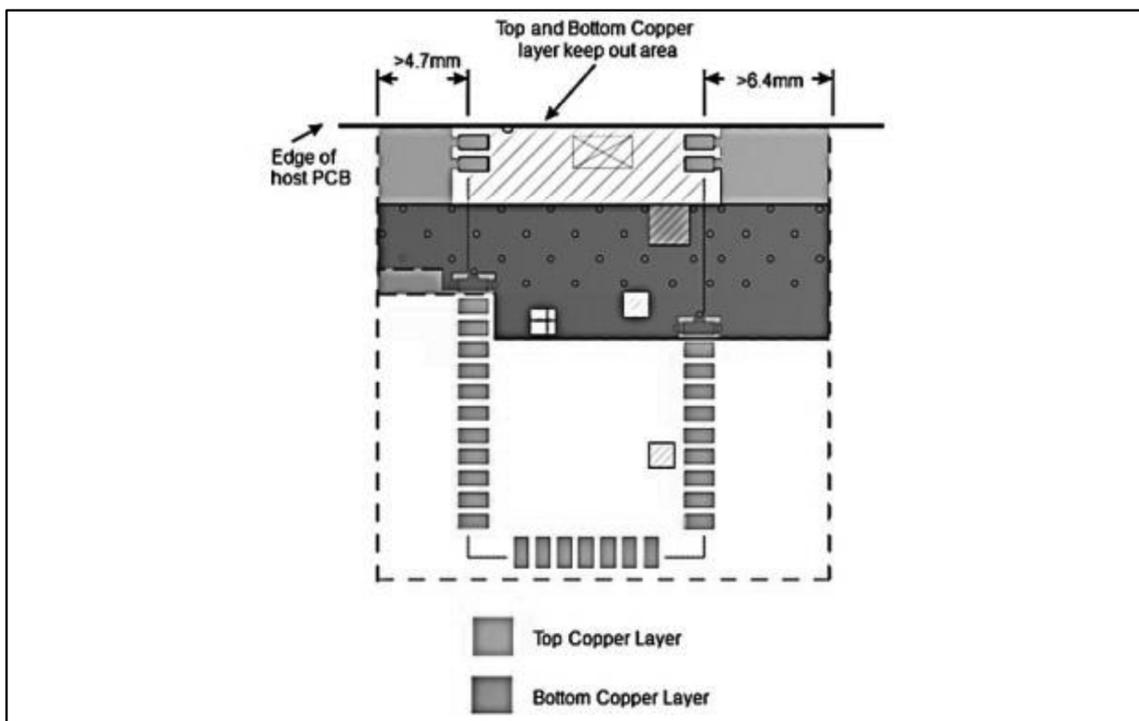


Figura 16 – Zona que debe evitar cubrirse en el Bluetooth (rayada)

Además, de la misma manera que con el microcontrolador, se han dispuesto los elementos pasivos (condensadores) de acuerdo con lo que indica el fabricante (Figura17).

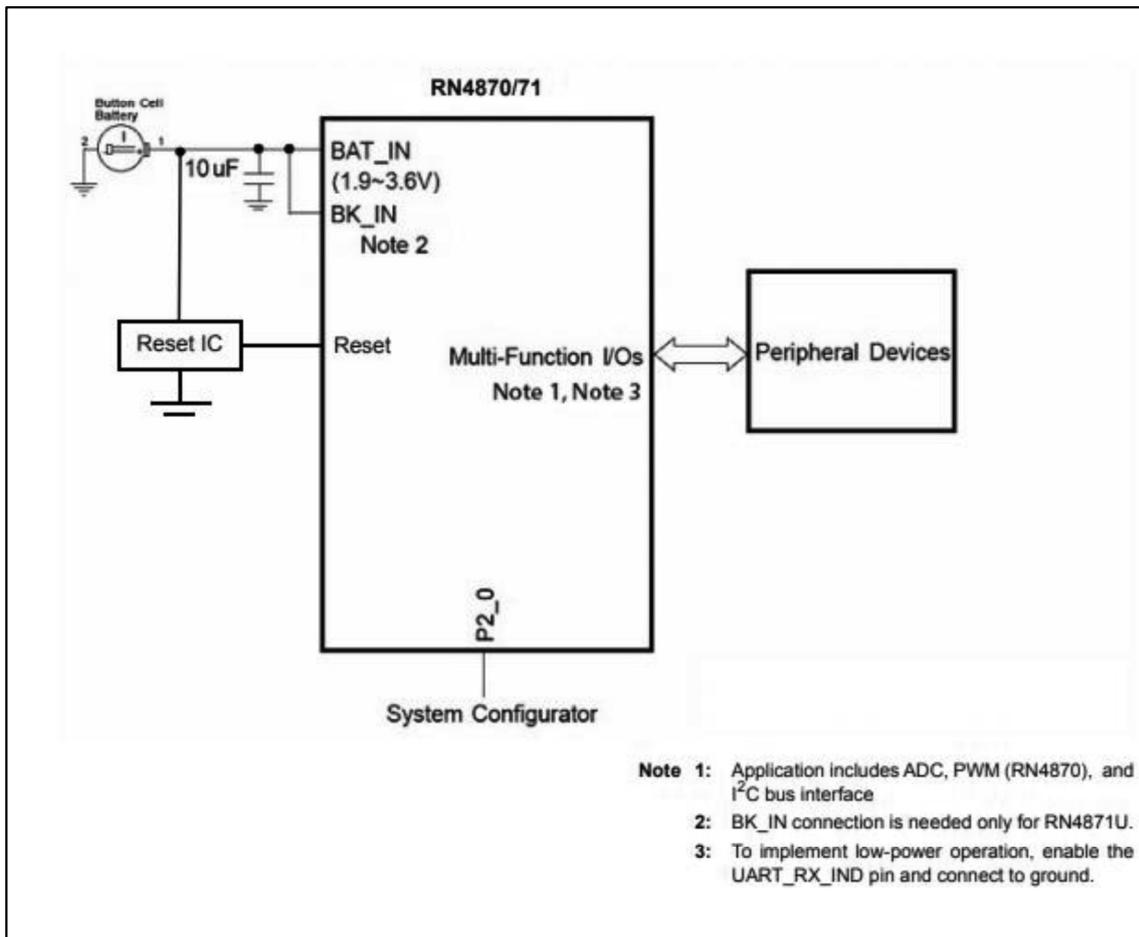


Figura 17 – Elementos pasivos en el RN4870

El regulador de voltaje utilizado ha sido el modelo TPS77001DBVT. Es un componente importante en tanto que no podemos alimentar toda la placa a un mismo voltaje. El láser necesita 4,5 V mientras que el microcontrolador y el Bluetooth expuestos precisan solo de 3,3 V para funcionar (más podría dañarlos).

El diodo láser escogido ha sido uno genérico del mercado de tensión nominal 4,5 V. Lo importante de este componente era el tamaño.

El fotorreceptor utilizado ha sido el OPB732WZ, un modelo común en el mercado y del que también se tienen referencias de trabajos anteriores.

Elementos pasivos

Entre los componentes pasivos de la placa se encuentran los condensadores, las resistencias, el transistor y el cristal.

Los condensadores que se han utilizado son de entre 0,1 y 10 μF , su función principal es la de desacoplamiento de la tensión en el microcontrolador, el Bluetooth y el regulador de tensión.

Las resistencias son 6; dos de 10 $\text{k}\Omega$, dos de 10 $\text{M}\Omega$, una de entre 255 Ω y otra de entre 130 Ω . Las más pequeñas corresponden a la resistencia de base (255 Ω) y de colector (130 Ω) del transistor del láser, que controlan la basculación entre saturación y corte.

En cuanto a las de 10 $\text{k}\Omega$, una resistencia de 10 $\text{k}\Omega$ se conecta al micro y se utiliza para controlar el “reset” de este en el pin MCLR y la otra está en el receptor y controla la corriente que le llega a este.

Por último, las más grandes de 10 $\text{M}\Omega$ se usarán para conseguir un divisor de tensión a la entrada del pin del convertidor analógico-digital.

El transistor es NPN, se ha utilizado a la hora de conectar el láser al microcontrolador de manera que cuando le llega corriente por la base deja pasar corriente colector-emisor y el láser se activa.

El cristal es el RH100, las dos características relevantes a la hora de escogerlo son su reducido tamaño de 3,2 mm x 2,5 mm y su frecuencia de 11,0592 MHz.

Otros elementos

Otros elementos utilizados en el conjunto han sido la caja de color blanco, los tornillos de montaje de esta y los de sujeción de la placa, 3 pilas de 1,5 V tipo LR03 (AAA) como fuente de alimentación y cableado para conectar el receptor, el láser y la alimentación a la placa.

También, entre la alimentación y el resto del circuito, se ha dispuesto un interruptor de encendido y apagado por comodidad a la hora de usar el dispositivo.

III. Programación del sistema

Para la programación se ha utilizado el software MPLAB (distribuido por Microchip) junto con el compilador CCS C. Con un hardware de programación (PicKit 3) se ha conectado la placa al ordenador y realizado la programación. Todos los programas que se van a describir a partir de este punto están completos en el anexo correspondiente. En las páginas siguientes se mostrarán únicamente fragmentos que ayuden a entender la explicación.

Se han desarrollado dos programas diferentes para cada extremo de la barrera luminosa y un tercero para el receptor de la comunicación Bluetooth. En la figura 18 se puede observar un diagrama sencillo de toda la comunicación.

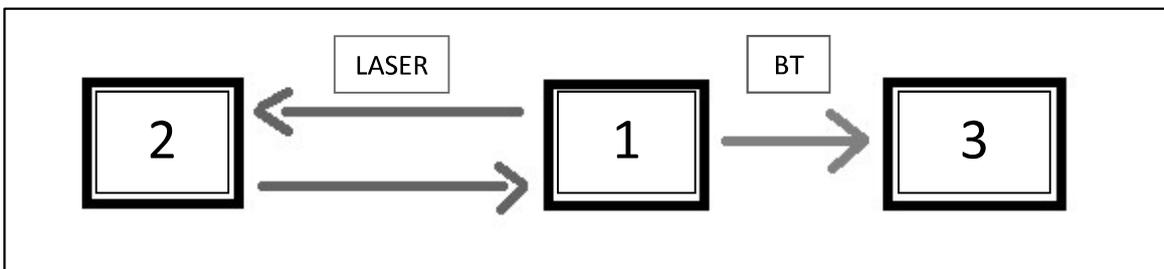


Figura 18 – Esquema de la comunicación

El programa de la caja 1 (comunicadora, maestro) está dividido en varias partes y a continuación se detallará cada una de ellas.

En primer lugar (Figura 19), tenemos la inclusión de la librería correspondiente a nuestro microcontrolador (DSPIC 33FJ128MC804) junto con 4 configuraciones (“fuses”) de este: asignación de los puertos para la programación, inhabilitación del mecanismo de “perro guardián”, habilitación del oscilador primario y, dado el cristal utilizado, hay que incluirle también la opción de alta velocidad del oscilador para que funcione correctamente.

Mediante la directiva #DEVICE ADC podemos cambiar el número de bits con el que queremos que trabaje el convertidor analógico-digital, el cual servirá para medir la batería y se explicará más adelante. En nuestro caso, serán 10, aunque otra posibilidad podrían ser 12 bits.

```
#include "33FJ128MC804.h"
#fuses ICSP1 // Puertos de programación
#fuses NOWDT // Sin WDT
#fuses PR // Oscilador primario
#fuses HS // Alta velocidad

#use delay (clock = 11059200)

#DEVICE ADC = 10
```

Figura 19 – Configuración y librería

Con la directiva `#pin_select` (Figura 20) se indican qué pin del microcontrolador hace qué función. Además, con `#use` se determinan las condiciones de la comunicación, como puede ser la tasa de baudios o el propio nombre del canal que hay que usar luego en las funciones.

```
#pin_select U1TX = PIN_B13
#pin_select U1RX = PIN_B4
#use rs232(UART1, baud=115200, STREAM=BT)

#pin_select U2TX = PIN_B11
#pin_select U2RX = PIN_C3
#use rs232(UART2, baud=300, STREAM=laser)
```

Figura 20 – `#pin_select` y `#use`

En la primera definición establecemos los pines B13 y B4 como la transmisión y recepción del Bluetooth, respectivamente, así como la tasa de baudios en 115200.

En la segunda procedemos de la misma manera, pero con el pin B11 y C3, los cuales, en este caso, son el láser y el fotorreceptor y van a una tasa de baudios de 300.

Previo a la función `main()` se declaran (e inicializan si procede) las variables “corte”, “corte_ant”, “batería” y “contador”.

```
int contador = 10;
int corte = 1, corte_ant=0, bateria;
char recibido;
```

Figura 21 – Variables

- **Corte:** su valor cambia cuando se recibe algo en el fotorreceptor; 1 indica que se ha recibido lo que se envió y 0 en cualquier otro caso. Con esta y con “corte_ant” se maneja que datos debe emitir la baliza.
- **Corte_ant:** permite guardar el estado del corte anterior. Con esto, aunque no haya un cambio en la variable “corte”, como se igualan ambas variables, no se generan nuevas balizas, sino que se mantiene la que ya hay del bucle anterior.
- **Batería:** es la variable que almacena el valor proveniente del convertidor analógico-digital y permite saber cual es el estado de carga de la alimentación.
- **Contador:** se inicializa a 10 y va disminuyendo en 1 con cada bucle (cada bucle contiene un “`delay_ms()`” de 100 ms, con lo que la cuenta total dura 1 segundo). Se actualiza de nuevo a 10 cuando se detecta una recepción en el fotorreceptor. Si se agotara la cuenta, es decir, que no se ha recibido nada en 1 segundo, se establecen las variables “corte” y “corte_ant” al estado de *corte* generando así la baliza correspondiente en el BT para indicarlo.

En el main() se llama a la función IniBT(), se configura el conversor analógico-digital y se habilita la interrupción de recepción de datos. En el ADC se usa el reloj interno y se referencia a la tensión de alimentación del microcontrolador (3,3V) y masa. Como hemos usado el pin AN2 el canal a seleccionar es el 2.

```

setup_adc(ADC_CLOCK_INTERNAL);

setup_adc_ports(sAN2,VSS_VDD);

set_adc_channel(2); // Selección

InitBT();

clear_interrupt(INT_RDA2);
enable_interrupts(INT_RDA2);

```

Figura 22 – Configuración en el main() del programa

Con la función InitBT() se configura el Bluetooth. Entrando en el modo comando se establecen características básicas como el nombre (“SN,xxx”) y la dirección MAC (“S:,0000,xxx”) del dispositivo y también se crea el servicio de baliza (“beacon” en inglés) con el cual nos comunicamos con el dispositivo central (“SS,20”). Con la configuración hecha, lo último que se hace en la función es habilitar el servicio mediante el comando “SC,x”.

```

void InitBT(void)
{

    fprintf(BT, "$$$"); // Command mode from Data mode
    delay_ms(100);

    fprintf(BT, "Y\r\n"); // Stop advertisement
    delay_ms(500);
    fprintf(BT, "PZ\r\n"); // Stop services
    delay_ms(500);
    fprintf(BT, "IB, Z\r\n"); // Stop Beacon
    delay_ms(500);
    fprintf(BT, "IA, Z\r\n"); // Stop advertising
    delay_ms(500);
    fprintf(BT, "SF, 1\r\n"); // Factory Reset
    delay_ms(1000);

    fprintf(BT, "$$$"); // Command mode
    delay_ms(100);

    fprintf(BT, "SN, BARRERA\r\n");
    delay_ms(500);

    fprintf(BT, "S:, 0000, 1EF2F33980D8\r\n");
    delay_ms(300);

    fprintf(BT, "SS, 20\r\n"); // Servicio: Beacon
    delay_ms(500);

    fprintf(BT, "IB, FF, 4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4\r\n");
    delay_ms(500);

    fprintf(BT, "SC, 2\r\n"); // Enable (0-beacon, 1-conectable, 2-ambos)
    delay_ms(500);

}

```

Figura 23 – “Bitmap” de la instrucción “SS,xx” --- extraído del manual de usuario del RN4870

Posteriormente, en el bucle infinito (“superloop”) dentro del main(), se manda continuamente a través del láser una información concreta que será el comprobante del estado de la barrera. También se lee el estado de la batería a través del conversor analógico-digital y se hacen una serie de comprobaciones a base de “ifs-else” para saber qué baliza se debe activar como respuesta en función del estado de las variables “corte”, “corte_ant” y la recientemente leída “batería”. Para más detalles sobre las posibles combinaciones consultar el programa completo en el anexo.

```
bateria = read_adc();  
fputc(0x5A, laser);
```

Figura 24 – Lectura del ADC y disparo del láser

Por último, del primer programa, queda por analizar la interrupción de la recepción. Esta almacena en la variable “recibido” lo que le llega desde el fotorreceptor y después lo compara con lo que se había mandado originalmente. Si estos coinciden, entonces no ha habido corte pues el mensaje ha llegado sin problemas, pero si no, entonces es que algo lo ha detenido en medio.

```
#int_RDA2  
void RDA2_isr(){  
  
    contador = 10;  
    recibido = fgetc(laser);  
  
    if (recibido == 0x5A){  
        corte = 0;  
    }else{  
        corte = 1;  
    }  
}
```

Figura 25 – int_RDA en el programa maestro

El programa correspondiente a la caja 2 (pareja, esclavo...) consiste únicamente en activar su láser enviando la misma información que ha recibido en el sensor. El programa guarda la información en la variable “c” y la envía después con la instrucción “fputc”. Si no se recibe nada, envía un 1 continuamente sin información. Comparte con la caja 1 la configuración del microcontrolador y la referente a la recepción externa (int_RDA).

```
#int_RDA2  
void RDA2_isr(){  
    recibido = 1;  
    c = fgetc(laser);  
}
```

Figura 26 – int_RDA en el programa pareja

```

for (;;)
{
    if (recibido==1)
    {
        delay_ms(40);
        fputc(c, laser);
        recibido=0;
    }
}

```

Figura 27 – Bucle infinito en el programa pareja

Por último, tenemos el programa para un dispositivo Bluetooth basado en el mismo microcontrolador y Bluetooth utilizados en el proyecto principal que se incluiría en el sistema mayor descrito al principio de la memoria y funcionaría de receptor (caja3).

No obstante, antes de nada, conviene aclarar cómo funciona el sistema Bluetooth 4.0 de bajo consumo (BLE) utilizado y hablar de los roles de cada dispositivo en la comunicación.

En nuestro sistema tenemos 2 Bluetooths, uno que envía y otro que recibe (el número podría ampliarse si fuera necesario cubrir más zonas). Cada uno tiene un rol tipo GAP (*General Access Profile*) definido y que, en este caso, no cambiará a lo largo de su funcionamiento.

El dispositivo que recibe (caja 3) tendrá un rol de “observer” y únicamente escuchará los paquetes de datos que envíe el otro dispositivo sin responder a ellos ni establecer una conexión directa. Para conseguir que no haya otras balizas ajenas al sistema que interfieran en la recopilación de datos, durante la programación, se establecerá una lista blanca de direcciones que puede detectar.

Por otra parte, el otro dispositivo, para hacer ese envío, se definirá como “broadcaster” y estará continuamente enviando paquetes “al aire” que cualquiera puede recibir. Como se ha comentado antes, podrían haber más de uno componiendo el sistema y deberían de incluirse todos en la mencionada lista blanca del “observer”.

Dicho esto, podemos pasar a explicar cada uno de los parámetros modificados en la función InitBT() de este tercer dispositivo.

Como ya se ha comentado, su función consiste en escanear las balizas de los alrededores. Para ello se deben configurar dos parámetros. En primer lugar, debe ignorar los dispositivos con conexiones activas (de manera que solo detecte balizas). Y, por otra parte, se debe deshabilitar el filtro que tiene por defecto de dispositivos repetidos. Así, independientemente de que ya haya detectado una vez la dirección MAC de nuestra baliza, no la omitirá después en sucesivos escaneos, pues es importante denotar que el cambio del “major” y el “minor” en la emisión de la baliza no provoca que sea detectada de nuevo, con lo cual hay que obligar que eso ocurra.

Esto se puede conseguir mediante la instrucción “SR,xxxx” aplicando las características del “bitmap” correspondientes.

```
fprintf(BT, "SR,0C00\r\n"); // NO-filtro; Solo No-conectables
delay_ms(500);
```

TABLE 2-5: BITMAP OF FEATURES

Feature	Bitmap	Description
Enable Flow Control	0x8000	If set, the device enables hardware flow control.
No Prompt	0x4000	If set, device does not send prompt CMD> when RN4870/71 is ready to accept the next command. If cleared, device sends out prompt CMD> when it is ready to take the next command.
Fast Mode	0x2000	If set, no checking of configuration detect character in Transparent UART mode is done. Instead, to enter Command mode, RN4870/71 depends on the pin configured as UART Mode Switch.
No Beacon Scan	0x1000	If set, no non-connectable beacon shows up in the scan result.
No Connect Scan	0x0800	If set, no connectable advertisement shows up in the scan result.
No Duplicate Scan Result Filter	0x0400	If set, RN4870/71 does not filter out duplicate scan results. It is recommended that this bit is set if the RN4870/71 expects a beacon or a peer device which dynamically changes its advertisement.
Passive Scan	0x0200	If set, RN4870/71 performs passive scan instead of default active scan.
UART Transparent without ACK	0x0100	If set, the device uses Write without Response for UART Transparent when communicating with another RN4870/71 module. If cleared, the device uses Write Request for UART Transparent when communicating with another RN4870/71 module.
Reboot after Disconnection	0x0080	If set, the RN4870/71 reboots after disconnection.
Running Script after Power On	0x0040	If set, the RN4870/71 automatically runs the script after powering on, starting with @PW_ON event.

*Figura 28 – Bitmap de la instrucción “SR,xxxx”
– Extraído del manual de usuario del RN4870*

Para la lista blanca se debe utilizar la instrucción “JA,x,xxxx” donde se debe introducir cuál es la dirección deseada y si es pública (0) o privada (1).

```
fprintf(BT, "JA,1,F0A1B4002084\r\n");
delay_ms(500);
```

En nuestro caso la dirección es F0A1B4002084 y es privada.

Después, para empezar el escáner, se usa la instrucción “F” que automáticamente pone el dispositivo en modo “observer” detectando otros dispositivos cercanos.

```
fprintf(BT, "F\r\n");
```

Para discernir entre los dispositivos detectados y recoger la información que contienen las balizas sin establecer conexión con ninguno utilizaremos un vector de caracteres. Este vector se llena durante la interrupción RDA y contiene una serie de caracteres entre los cuales se encuentran los valores del “mayor” y el “minor” que nos permitirán saber el estado de la variable medida (en este caso, el corte o no de la barrera).

La cadena habitual en la emisión de la baliza está compuesta de 80 caracteres con un símbolo “%” al principio y otro al final. Además, se sabe que 3 posiciones antes del último “%” acaba el último carácter del “minor”.

%FOA1B4002084,1,B0,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00030004C4%

DATO	%	F	0	A	1	B	4	0	0	2	0	8	4	,	1	...	F	2	1	E	0	0	0	3	0	0	0	4	C	4	%
VECTOR	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14		65	66	67	68	69	70	71	72	73	74	75	76	77	78	79

El vector se guarda en la variable “trama” y mediante una variable índice “n” se va comprobando cada dato del vector. De esta manera, si hay dos “%” separados 78 caracteres, ahí hay una posible baliza y entonces se pasa a comprobar la posición 76 entre los símbolos.

Con esta técnica se pueden comprobar todos los valores del “mayor” y el “minor” y realizar las acciones pertinentes en función de estos.

En la siguiente imagen podemos ver un ejemplo de las sentencias que estarían dentro de la recogida de datos.

```
#int_rda

void isr_rda(void)

{

    trama[n++]=fgetc(BI);
    if(trama[n-1]=='%')
    {
        if(trama[n-80]=='%')
        {
            if(trama[n-4]=='2')
            {
                ON(PIN_C1);
                OFF(PIN_B3);
            }
            if(trama[n-4]=='4')
            {
                OFF(PIN_C1);
                ON(PIN_B3);
            }
        }
    }
}
```

Aquí solo comprobamos el estado de las balizas. No obstante, se podrían comprobar todos los datos que se necesitaran, como, por ejemplo, el estado de la batería.

Hay que remarcar unos cuantos aspectos de este proceso. En primer lugar, debemos tener en cuenta que la información de estas balizas está llegando al microcontrolador por el canal serie y las está guardando en un buffer de recepción.

Además, también es importante el momento en el que se ha habilitado esta interrupción, pues, para que el buffer esté vacío a la hora de recibir datos, hay que leer también previamente todas las respuestas desde el Bluetooth que le llegan al microcontrolador en la fase de configuración de InitBT(). Para ello basta con habilitar la interrupción antes de la propia ejecución de InitBT().

```
clear_interrupt(int_rda);  
  
enable_interrupts(int_rda);  
  
InitBT();
```

IV. Montaje del conjunto

Habiendo descrito ya de qué partes se forma el dispositivo y cómo está programado internamente, toca ahora comentar el proceso que se ha seguido al unir dichas partes.

Antes de nada, es importante recordar que al principio planteábamos un objetivo: el tamaño reducido del conjunto. Por tanto, hay que remarcar que el factor que más ha condicionado el montaje del proyecto ha sido la caja en la cual se iba a implementar. Usar esta y no otra ha sido una decisión externa y todo se ha tenido que acoplar en mayor o menor medida a ello. Aun así, si hubiera sido totalmente imposible proseguir con esa restricción, cabía la posibilidad de escogerse otro recipiente más grande para el conjunto, pero se pretendía buscar un dispositivo pequeño y manejable.

La caja mide 25 mm de altura, 50 mm de anchura y 72 mm de profundidad y en ella se han tenido que introducir todos los componentes conectados y asegurándose de que no se desmonta ninguno por un golpe.

En dicha caja se han tenido que hacer 3 agujeros en los cuales se han fijado con termocola cada uno de los elementos. Uno para el láser, otro para el receptor y otro para el interruptor de encendido. Los dos primeros se han realizado en la tapa y el otro en un lateral usando con una máquina cortadora por control numérico siguiendo los planos que se pueden consultar al final del presente documento en la parte de anexos.

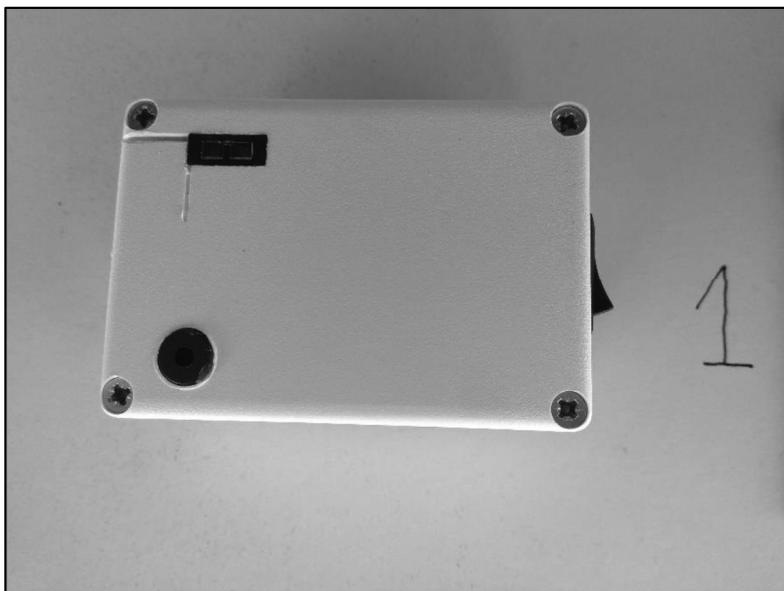


Figura 29 – Caja montada, se pueden ver los 3 agujeros necesarios

Las pilas se han pegado a la caja y entre ellas mediante termocola también. Además, dado el reducido espacio, no se han podido poner conexiones típicas (de muelle) en las paredes o con un soporte estándar, con lo que para unir el circuito de las 3 pilas lo que se ha hecho es soldar sus extremos con un cable. Para tener control del estado (encendido o apagado) del dispositivo, el circuito se ha hecho pasar por el interruptor.

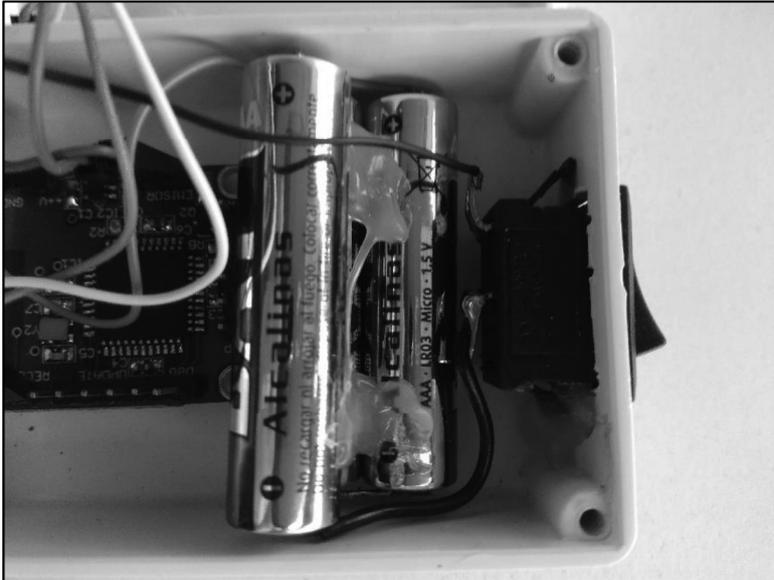


Figura 30 – Pilas pegadas con termocola

Con la placa de circuito impreso se han tenido que emplear dos técnicas diferentes para su montaje en la caja. En un principio, la placa ha sido diseñada con 4 espacios para tornillos de plástico de manera que al atornillarlos contra una tuerca y aplicar calor en su cabeza se podían pegar al fondo de la caja, dejando espacio suficiente para los componentes en ambas caras.

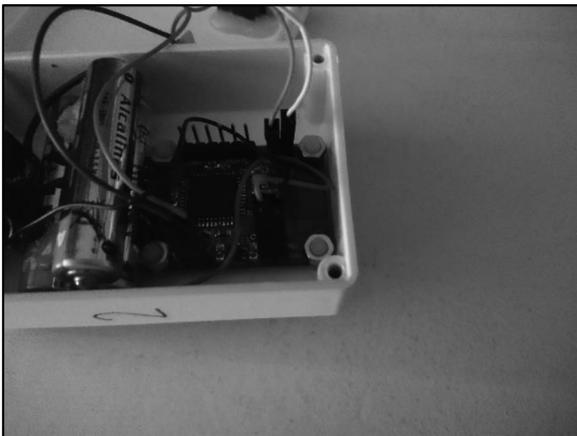


Figura 31 – Tornillos de la placa

No obstante, durante las pruebas de funcionamiento, para la sujeción de la caja en una escuadra de metal se han pegado imanes con cinta de doble cara por fuera. Estos imanes, aunque han cumplido su función, también atraen la placa y sus componentes. Esto no es un problema si seguimos un orden durante el montaje, primero se fijan los tornillos de la placa al fondo, de manera que los imanes no tienen suficiente fuerza para influir en la placa después.

Pero en uno de los dispositivos, este orden se ha invertido y, con la influencia de los imanes, era muy difícil fijar la placa por el tiempo necesario para que los tornillos se enfriaran. En este caso lo que se ha hecho es poner cinta de doble cara en el Bluetooth, el cual tiene superficie suficiente para adherirse bien a la caja y no haya problemas. No se espera ningún fallo o mal funcionamiento del componente por este motivo, pero no era el diseño planteado al principio y denota la importancia del orden durante el montaje si se quieren usar partes imantadas.

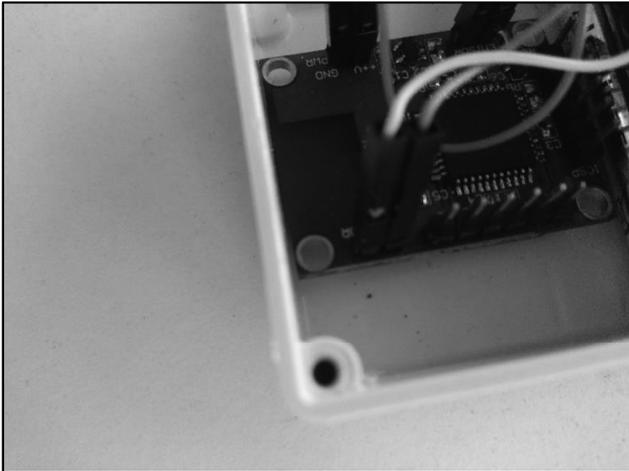


Figura 32 – Placa sin tornillos

4.Pruebas y resultados

Duración de la batería

Cumpliendo con el objetivo que nos hemos propuesto, es importante analizar la duración de la batería para conseguir la mayor autonomía posible en nuestro dispositivo.

Para ello, lo que se hace es tener monitorizado en todo momento el valor de tensión entre ambos extremos de la alimentación.

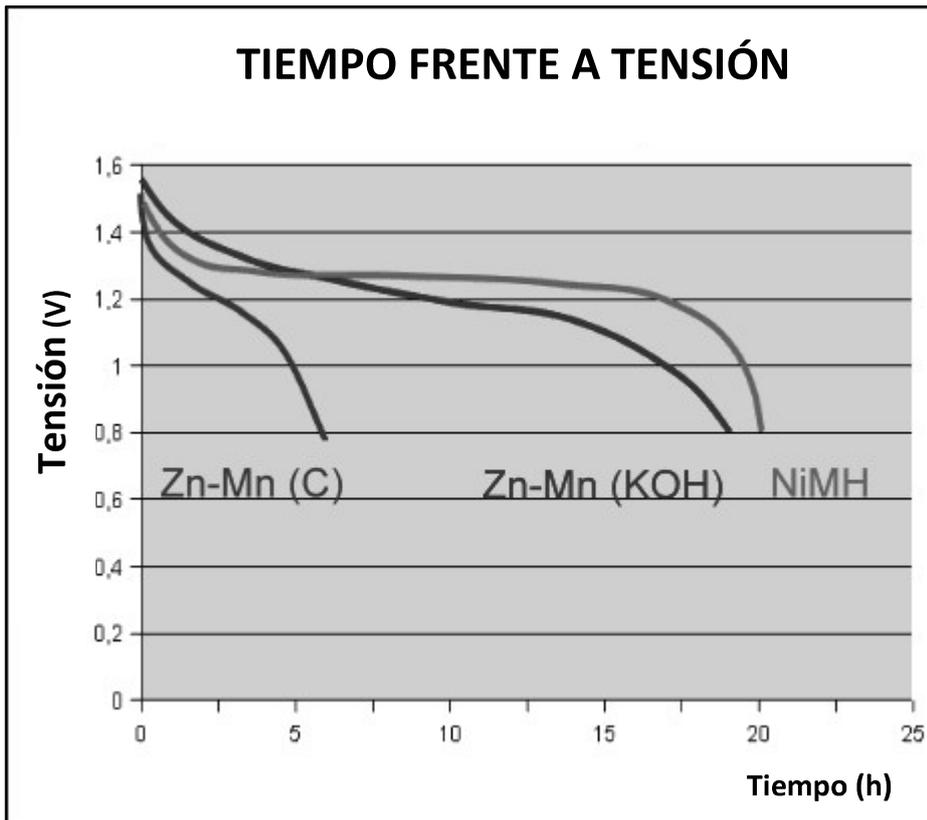
Utilizando unos de los puertos de programación como entrada analógica y mediante el convertidor analógico-digital interno del microcontrolador se tiene esa información disponible en el programa.

Nuestras referencias son: la tensión de 3,3 V que proviene del regulador de tensión y que alimenta tanto al microcontrolador como al Bluetooth, y, masa. Con un divisor resistivo podemos conseguir que la tensión de 4,5 V que ofrecen las pilas esté dentro del rango fijado.

Dado que el convertidor es de 10 bits, el valor máximo de 3,3 V será convertido al número 1023 y la tensión pasado el divisor resistivo, que es de 2,25 V, corresponderá al número 698. A continuación, se muestra una tabla con las conversiones de los valores de carga escogidos.

Porcentaje	Tensión (V)	Tensión (divisor) (V)	Valor convertido
100%	4,5	2,25	698
90%	4,05	2,025	628
80%	3,6	1,8	558
70%	3,15	1,575	488
50%	2,25	1,125	349

La variable “batería” es la que almacena el valor obtenido en el convertidor, sin embargo, dar ese valor directamente en cada envío de información podría generar confusión y no entenderse a primera vista su variación. Por ello, en el programa, en lugar de tener un único valor lo que se tienen son diferentes intervalos (100-90%, 90-80%...), lo cual, aunque le quita precisión, lo hace más fácil de entender.



- Pila de zinc-carbono
- Pila alcalina de manganeso
- Pila de níquel e hidruro metálico

Curva de descarga comparativa de diferentes tipos de pilas extraído de www.wikipedia.com y traducido posteriormente

Cabe denotar que, a partir del 75%, ya puede haber fallos en el dispositivo debido a la baja tensión que proporcionan las pilas.

Monitorización del funcionamiento

Cada dispositivo y cada fase de la programación de este se han realizado y monitorizado aisladamente antes de pasar a la siguiente fase y al conjunto final. Además, se han tenido que hacer varias pruebas de programa antes de llegar al resultado definitivo.

En la programación del dispositivo Bluetooth, la tarea consistió en constatar que las órdenes al mismo se enviaban y recibían correctamente. Por ello, durante la fase de programación del proyecto, se tuvo el dispositivo conectado también al ordenador mediante conexión USB-TTL generando un puerto COM.

Esta conexión se realizó “pinchando” un par de cables en los pines de recepción y transmisión (RX y TX) y de ahí al USB-TTL. Con esto se podían leer las respuestas del Bluetooth a través de programas como el SIOU o el TeraTerm y así comprobar que estaba ocurriendo lo que se esperaba de la programación (entraba correctamente en modo comando/datos, enviaba las balizas adecuadas, etc).

```

Serial Input/Output Monitor
File Edit View Configuration
ASCII HEX Line Status Clear Terminal Columns Display Data Graph Ribbon
ASCII Send HEX Send Input/Output Viewing Options Tools Menu Style

CMD> $$$ %REBOOT%CMD> Y
AOK
CMD> PZ
AOK
CMD> IB,Z
AOK
CMD> IA,Z
AOK
CMD> SF,1
Reboot after Factory Reset
%REBOOT%$$$CMD> SN,CENTRAL
AOK
CMD> SC,2
AOK
CMD> SR,0C00
AOK
CMD> JC
AOK
CMD> JA,1,FOA1B4002084
AOK
CMD> JD
FOA1B4002084,1
END
CMD> F
Scanning
%FOA1B4002084,1,D3,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D2,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D5,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D3,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D5,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D2,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D5,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%

```

Figura 33 – Respuesta a los comandos enviados desde el microcontrolador y comienzo de escaneo

```

%FOA1B4002084,1,D2,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D5,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D2,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D5,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D4,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4%
%FOA1B4002084,1,D6,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00030004C4%
%FOA1B4002084,1,D4,Brcst:1AFF4C0002152F234454CF6D4A0FADF2D88039F3F21E00030004C4%

```

Figura 34 – Cambio de baliza detectado y dirección

En las figuras 33 y 34 de arriba se pueden ver las respuestas a los comandos enviados, así como el cambio en la baliza (de 00010002 a 00030004) y la dirección MAC que tienen estas balizas (la cual se sabe que es privada por el número 1 después de la coma) y que hay que incluir en la lista blanca con el comando “JA,x,xxxx”.

Conociendo estas respuestas se desarrolla un programa de control del conjunto.

Prueba del funcionamiento y montaje

Ya como conjunto completo (los dos dispositivos y el tercero que hace de receptor de balizas) también se ha hecho una prueba final de funcionamiento.

En esta se ha dispuesto la barrera como se haría en un caso más real. Los dispositivos están a una distancia de casi medio metro para comprobar el funcionamiento y, además, la dificultad para encarar los dispositivos. Ambos objetivos son cruciales, no obstante, pensando en el proyecto como un producto que podría estar en el mercado, el segundo objetivo es casi más importante puesto que, aunque funcione, si el usuario final no es capaz de montar la barrera en poco tiempo o con cierta facilidad, desearía esta opción para resolver el problema rápidamente.

Se ha hecho la prueba para comprobar cuánto tiempo tardarían en montarlo varias personas individualmente y, estando el concepto claro en todos los casos, los tiempos varían desde 1 minuto hasta 4 o 5 minutos para montarlo.

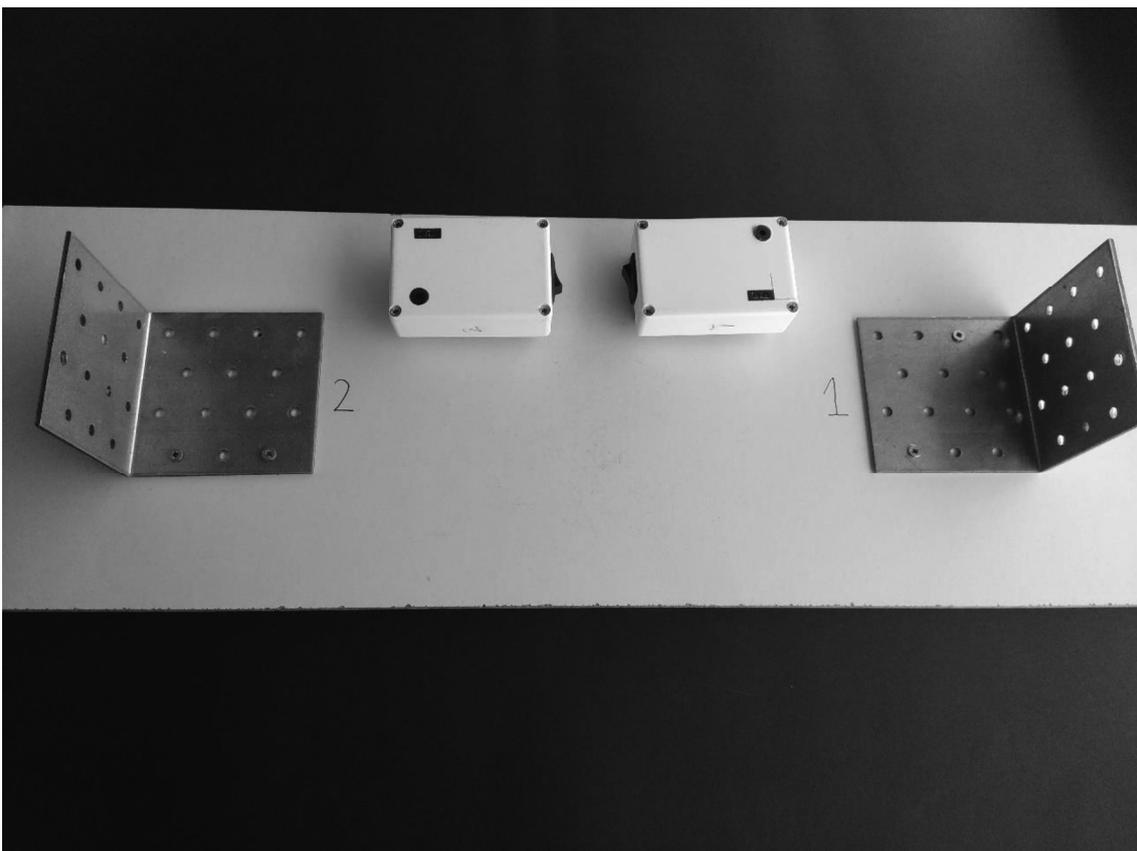


Figura 35 – Montaje

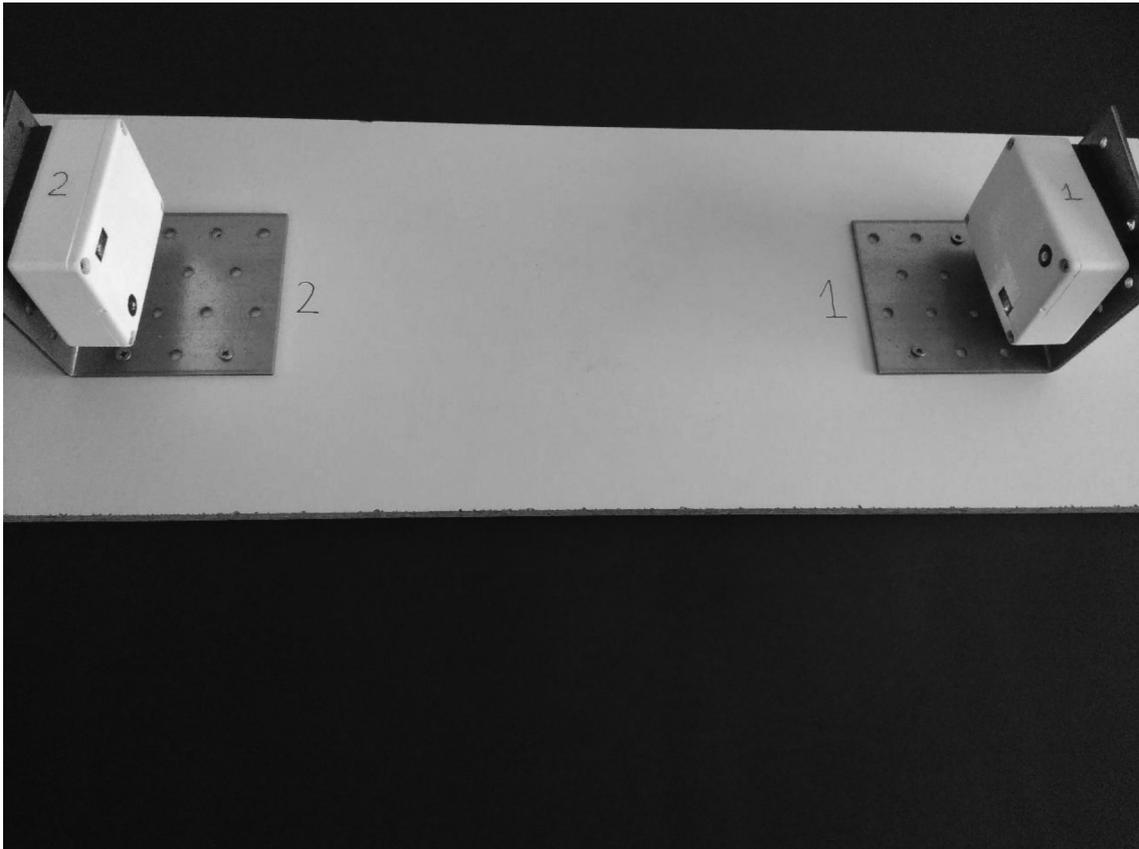


Figura 36 – Montaje encendido

Aparte de la barrera, también necesitamos un tercer dispositivo que haga de receptor y actúe en función del estado de la barrera. Para las pruebas, se han utilizado 3 LEDs montados en la tercera placa: uno azul que indicará que está funcionando el dispositivo, uno rojo que denotará el estado de reposo (barrera sin cortar) y uno verde que se encenderá cuando haya un corte en la barrera.

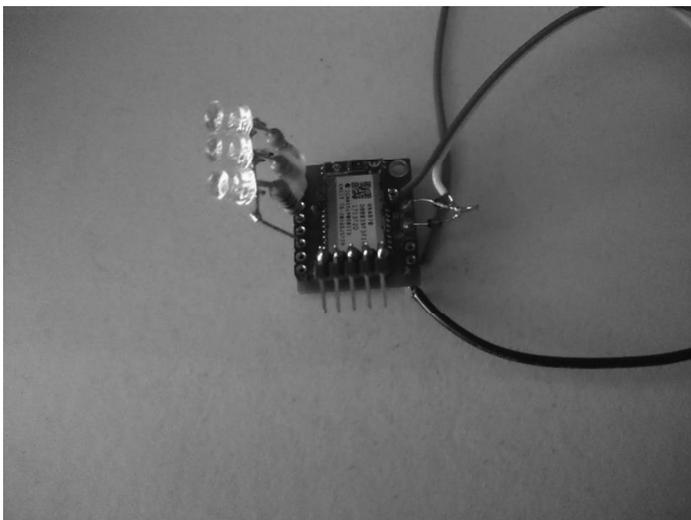


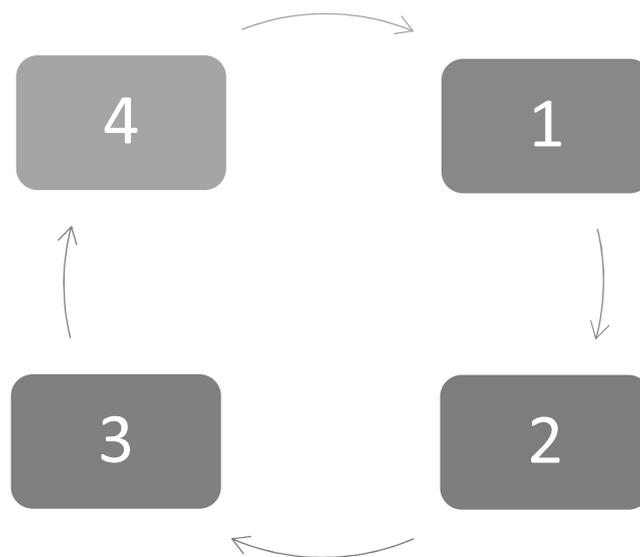
Figura 37 – Tercera placa con LEDs

5. Conclusión y líneas futuras

Como ya se comentó al principio de la memoria en el apartado de antecedentes al proyecto, este sistema tiene cierto parecido con algún otro sistema que realiza funciones similares y que ya existe en la realidad a nivel comercial.

No obstante, por la manera en la que se ha diseñado, este puede ofrecer en el futuro funciones adicionales con cambios no muy grandes.

En primer lugar, sin modificar qué se está midiendo o comprobando y manteniendo el concepto de barrera luminosa, se puede pensar en un perímetro de barreras para detectar cualquier cosa que entre o salga de ese perímetro establecido. Modificando un poco el programa para cada dispositivo y la situación de los láseres y los receptores en las cajas se puede tener que el primer dispositivo emita hacia el segundo, el segundo lo repita al tercero y que así, eventualmente, volviese al primero para la comprobación. Además, dado que podemos enviar información a través de los láseres, podríamos, no solo saber que se ha cortado el perímetro, sino también en qué parte de este ha ocurrido el corte y enviar la información por Bluetooth.



Por otra parte, hay que destacar que nuestro sistema, básicamente, lo que hace es medir una magnitud y después enviar información en función de lo que se haya medido. Si ahora cambiamos la barrera luminosa por una báscula para pesar, la idea del sistema no cambia, seguimos midiendo algo y en función de lo medido podríamos seguir mandando información a través del Bluetooth, pero que tuviera otro significado. En este caso, la modificación al programa sería mínima, pero habría que tener en cuenta cómo se recogen esos datos en la PCB.

Como última modificación propuesta, también tenemos la posibilidad de reducir aún más el consumo del dispositivo.

En la comunicación serie utilizada en el láser, el estado inactivo en la comunicación se codifica, por defecto, con un 1. Por lo que, en lo que se ha propuesto hasta ahora, el láser siempre está encendido, aunque no envíe información útil (Figura39). No obstante, esta lógica se puede cambiar en el programa bien mediante la palabra clave INVERT usada en la directiva “#use rs232” o bien modificando directamente el registro UxSTA (registro de transmisión de la UART). Con esto, el estado inactivo pasa a ser 0 y por lo tanto solo se enciende para enviar información (Figura 40).

REGISTER 20-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

bit 14 **UTXINV:** Transmit Polarity Inversion bit
 If IREN = 0:
 1 = UxTX Idle state is '0'
 0 = UxTX Idle state is '1'

Figura 38 – Bit modificable en el registro de transmisión de la UART

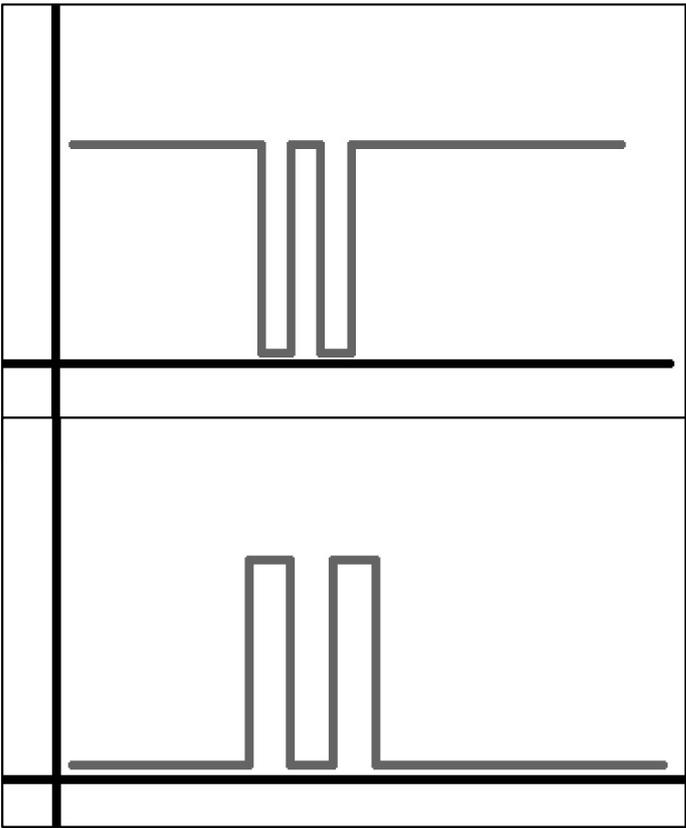


Figura 39 – Envío de información con estado inactivo alto

Figura 40 – Envío de información con estado inactivo bajo

En conclusión, el dispositivo es novedoso en cuanto a que incluye la comunicación vía Bluetooth, sin conexión directa, tiene la posibilidad de ser empleado en distancias muy diferentes (tanto muy reducidas como muy alejadas) y, por tanto, es un dispositivo muy versátil que, además, es pequeño, manejable y consume poca batería, lo que le da una elevada autonomía de funcionamiento.

PRESUPUESTO

Presupuesto

En este apartado se describen todos los costes fundamentales en los que se ha incurrido durante la realización de este proyecto, incluyendo materias primas empleadas, herramientas y máquinas utilizadas y horas de trabajo invertidas.

Materias primas

Para las materias primas se incluye la referencia del distribuidor Farnell.

ELEMENTO	NOMBRE	REF.FARNELL	COSTE P/U	UNIDADES	C.TOTAL (€)
Microcontrolador	DSPIC33FJ128MC804-PT	1576810	5,94	1	5,94
Bluetooth	RN4870-V_RM118	2664513	5,93	1	5,93
Resistencia-- ADC-Divisor resistivo 10 MΩ	Resistencia SMD de Tipo Chip	1469858	0,0386	2	0,0772
Resistencia-- Transistor--Base 250Ω	Resistencia SMD de Tipo Chip, Película Gruesa	2059300	0,0277	1	0,0277
Resistencia-- Transistor-- Colector 130Ω	Resistencia SMD de Tipo Chip, Película Gruesa	2447248	0,0035	1	0,0035
Resistencia-- Micro--Pull-up 10KΩ	Resistencia SMD de Tipo Chip, Cerámica	2072517	0,0076	1	0,0076
Resistencia-- Receptor 10KΩ	Resistencia SMD de Tipo Chip, Cerámica	2072517	0,0076	1	0,0076
Condensador-- Micro 0,1 μF	Condensador de Cerámica, 0201	2526226	0,0472	3	0,1416
Condensador-- Cristal 0,1 μF	Condensador de Cerámica, 0201	2526226	0,0472	2	0,0944
Condensador-- Regulador 0,1 μF	Condensador de Cerámica, 0201	2526226	0,0472	2	0,0944
Condensador-- BT 10 μF	Condensador de Cerámica, 0603	2346889	0,476	1	0,476
Condensador-- MicroVCAP 10 μF	Condensador de Cerámica, 0603	2346889	0,476	1	0,476
Láser/Emisor	OPV330	2860818	4,71	2	9,42
Receptor	OPB732WZ	1678639	5,1	2	10,2
Conector/ Header x06	2211S-06G	1593415	0,000624	1	0,000624
Conector/ Header x02	2211S-02G	1593411	0,001464	6	0,008784
Transistor NPN	BC547B	1574381	0,244	1	0,244
Cristal-Oscilador	RH100-11.0592-18-3030- EXT	2509075	1,12	1	1,12
Regulador de tensión	TPS77001DBVT	8461554	1,35	1	1,35
Placa	PCB	---	14,9	1	14,9
Caja	Carcasa de Plástico, ABS	1277450	5,75	2	11,5
Pilas, cables y tornillos	Materiales de unión y sujeción	---	10	1	10

El coste total de las materias primas asciende a **72,02 €**.

Herramientas y maquinaria

Como única maquinaria empleada en el proyecto tenemos la máquina cortadora por control numérico. En esta se han mecanizado las cajas que conforman la barrera luminosa. Se ha empleado media hora por caja, de manera que para las dos cajas será una hora en total. No se ha contratado personal especializado para realizar el mecanizado.

MÁQUINA	COSTE POR HORA (€/h)	HORAS DE USO (h)	COSTE TOTAL (€)
Cortadora por control numérico	3	1	3

El coste total de las herramientas y maquinaria asciende a **3 €**.

Recursos humanos

En este proyecto sólo es necesaria la actuación y el trabajo de una persona, quien, teniendo los conocimientos necesarios sobre diseño de placas de circuito impreso y programación de microcontroladores, habría dedicado un total de 120 horas. De estas horas, un 40% correspondería a la etapa de diseño de la PCB, otro 40% correspondería a la etapa de programación y un 20% sería de pruebas sobre el conjunto.

No obstante, en total aprendizaje y elaboración han supuesto realmente 300 horas de trabajo.

PERSONAL	COSTE POR HORA (€/h)	HORAS EMPLEADAS (h)	COSTE TOTAL (€)
Etapa de diseño	35	48	1680
Etapa de programación	35	48	1680
Pruebas	35	24	840

El coste total del personal asciende a **4200 €**.

Total

CONCEPTO	COSTE TOTAL (€)
Materias primas	72,02
Herramientas y maquinaria	3,00
Recursos humanos	4200
TOTAL	4275,02

EL COSTE TOTAL DEL PROYECTO ASCIENDE A CUATRO MIL DOS CIENTOS SETENTA Y CINCO EUROS CON DOS CÉNTIMOS.

Pliego de condiciones

En este apartado se recogen las condiciones y supuestos que enmarcan el proyecto, así como las tolerancias de algunos de los materiales para tener un buen funcionamiento.

ELEMENTO	CARÁCTERÍSTICAS			
Condensador 0,1 μ F de capacidad 	<input type="checkbox"/> Encapsulado del Condensador Cerámico: 0201 [Métrica 0603] <input type="checkbox"/> Capacitancia: 0.1 μ F <input type="checkbox"/> Tensión Nominal: 4V <input type="checkbox"/> Tolerancia de Capacitancia: \pm 20% <input type="checkbox"/> Característica de Dieléctrico: X6S <input type="checkbox"/> Rango de Producto: C Series	<input type="checkbox"/> Temperatura de Trabajo Mín.: -55°C <input type="checkbox"/> Temperatura de Funcionamiento Máx.: 105°C <input type="checkbox"/> Estándar de Certificación para Automoción: - <input type="checkbox"/> Conformidad RoHS para Ftalatos: Por confirmar Nivel de Sensibilidad a la Humedad (MSL): MSL 1 - Ilimitado		
Condensador 10 μ F de capacidad 	<input type="checkbox"/> Encapsulado del Condensador Cerámico: 0603 [Métrica 1608] <input type="checkbox"/> Capacitancia: 10 μ F <input type="checkbox"/> Tensión Nominal: 6.3V <input type="checkbox"/> Tolerancia de Capacitancia: \pm 10% <input type="checkbox"/> Característica de Dieléctrico: X5R <input type="checkbox"/> Rango de Producto: Serie C	<input type="checkbox"/> Temperatura de Trabajo Mín.: -55°C <input type="checkbox"/> Temperatura de Funcionamiento Máx.: 85°C <input type="checkbox"/> Estándar de Certificación para Automoción: - <input type="checkbox"/> Conformidad RoHS para Ftalatos: Por confirmar Nivel de Sensibilidad a la Humedad (MSL): -		
Resistencia de 10 k Ω 	<input type="checkbox"/> Resistencia: 10kohm <input type="checkbox"/> Rango de Producto: Serie MCMR <input type="checkbox"/> Tensión Nominal: 50V <input type="checkbox"/> Tipo de Elemento Resistivo: Película Gruesa <input type="checkbox"/> Encapsulado de la Resistencia: 0402 [Métrica 1005]	<input type="checkbox"/> Potencia Nominal: 63mW <input type="checkbox"/> Tolerancia de Resistencia: \pm 1% <input type="checkbox"/> Coeficiente de Temperatura: \pm 100ppm/°C <input type="checkbox"/> Estándar de Certificación para Automoción: AEC-Q200		
Resistencia de 10 M Ω 	<input type="checkbox"/> Resistencia: 10Mohm <input type="checkbox"/> Rango de Producto: Serie CRCW <input type="checkbox"/> Tensión Nominal: 150V <input type="checkbox"/> Tipo de Elemento Resistivo: Película Gruesa <input type="checkbox"/> Encapsulado de la Resistencia: 0805 [Métrica 2012]	<input type="checkbox"/> Potencia Nominal: 125mW <input type="checkbox"/> Tolerancia de Resistencia: \pm 1% <input type="checkbox"/> Coeficiente de Temperatura: \pm 100ppm/K <input type="checkbox"/> Estándar de Certificación para Automoción: AEC-Q200		
Resistencia de 255 Ω 	<input type="checkbox"/> Resistencia: 255ohm <input type="checkbox"/> Rango de Producto: Serie ERJ3E <input type="checkbox"/> Tensión Nominal: 75V <input type="checkbox"/> Tipo de Elemento Resistivo: Película Gruesa <input type="checkbox"/> Encapsulado de la Resistencia: 0603 [Métrica 1608]	<input type="checkbox"/> Potencia Nominal: 100mW <input type="checkbox"/> Tolerancia de Resistencia: \pm 1% <input type="checkbox"/> Coeficiente de Temperatura: \pm 100ppm/°C <input type="checkbox"/> Estándar de Certificación para Automoción: AEC-Q200		

<p>Resistencia de 130 Ω</p> 	<ul style="list-style-type: none"> <input type="checkbox"/> Resistencia: 130ohm <input type="checkbox"/> Rango de Producto: Serie MCWR <input type="checkbox"/> Tensión Nominal: 50V <input type="checkbox"/> Tipo de Elemento Resistivo: Película Gruesa <input type="checkbox"/> Encapsulado de la Resistencia: 0603 [Métrica 1608] 	<ul style="list-style-type: none"> <input type="checkbox"/> Potencia Nominal: 100mW <input type="checkbox"/> Tolerancia de Resistencia: ± 1% <input type="checkbox"/> Coefficiente de Temperatura: ± 100ppm/°C <input type="checkbox"/> Estándar de Certificación para Automoción: -
<p>Regulador de tensión</p> 	<ul style="list-style-type: none"> <input type="checkbox"/> Tipo de Salida: Ajustable <input type="checkbox"/> Tensión de Entrada Mín.: 2.7V <input type="checkbox"/> Tensión de Entrada Máx.: 10V <input type="checkbox"/> Tensión Nominal de Salida Fija: - <input type="checkbox"/> Tensión Mínima de Salida Ajustable: 1.2V <input type="checkbox"/> Tensión Máxima de Salida Ajustable: 5.5V <input type="checkbox"/> Corriente de Salida: 50mA <input type="checkbox"/> Tensión de Caída Vdo: 35mV 	<ul style="list-style-type: none"> <input type="checkbox"/> Encapsulado del Regulador LDO: SOT-23 <input type="checkbox"/> Número de Pines: 5Pines <input type="checkbox"/> Temperatura de Trabajo Mín.: -40°C <input type="checkbox"/> Temperatura de Funcionamiento Máx.: 125°C <input type="checkbox"/> Rango de Producto: - <input type="checkbox"/> Estándar de Certificación para Automoción: - <input type="checkbox"/> Conformidad RoHS para Ftalatos: Sí Nivel de Sensibilidad a la Humedad (MSL): MSL 1 - Ilimitado
<p>Cristal Oscilador</p> 	<ul style="list-style-type: none"> <input type="checkbox"/> Frecuencia Nominal: 11.0592MHz <input type="checkbox"/> Encapsulado del Cristal: SMD, 3.2mm x 2.5mm <input type="checkbox"/> Estabilidad de Frecuencia + / -: 30ppm <input type="checkbox"/> Capacitancia de Carga: 18pF 	<ul style="list-style-type: none"> <input type="checkbox"/> Tolerancia de Frecuencia + / -: 30ppm <input type="checkbox"/> Rango de Producto: RH100 Series <input type="checkbox"/> Temperatura de Trabajo Mín.: -40°C <input type="checkbox"/> Temperatura de Funcionamiento Máx.: 85°C

Bibliografía

- Compilador C CCS y simulador PROTEUS para microcontroladores PIC

García Breijo, Eduardo

- Getting Started with Bluetooth Low Energy---Tools and techniques for low-power networking

Kevin Townsend, Carles Cufí, Akiba &Robert Davidson

Manual de usuario

En vistas al usuario final, aquí detallaremos qué pasos hay que seguir para conseguir un correcto funcionamiento del conjunto.

- Los dispositivos de las cajas blancas tienen un botón de encendido en el lateral.
- Hay que disponerlos en superficies planas firmes, encarados el uno al otro
- El láser se debe apuntar a la parte del receptor situada más en el extremo. Para encararlos se recomienda dejar uno fijo y mover el otro para conseguir que los láseres apunten a los receptores.
- Se recomienda, para poder encontrar la posición ideal para ambos láseres, que se mantenga un láser apuntando al receptor de su gemelo y, sin romper esa conexión, se rote el otro dispositivo para encontrar la posición ideal.
- Aunque no es necesario que ambos láseres apunten exactamente al cuadrado del fotorreceptor por completo, es mejor que no se desvíen demasiado para no perder información.

A partir de este momento, ya se estarán enviando balizas y se podrán detectar con dispositivos Bluetooth. Convendría tener instalado el dispositivo receptor y fijarse en los LEDs para saber si funcionan correctamente. Los LEDs son meramente informativos, pero se pueden añadir funciones extra cuando se detecte una baliza u otra con la electrónica adecuada.

ANEXOS

Anexos

Índice del anexo

1) Circuito electrónico

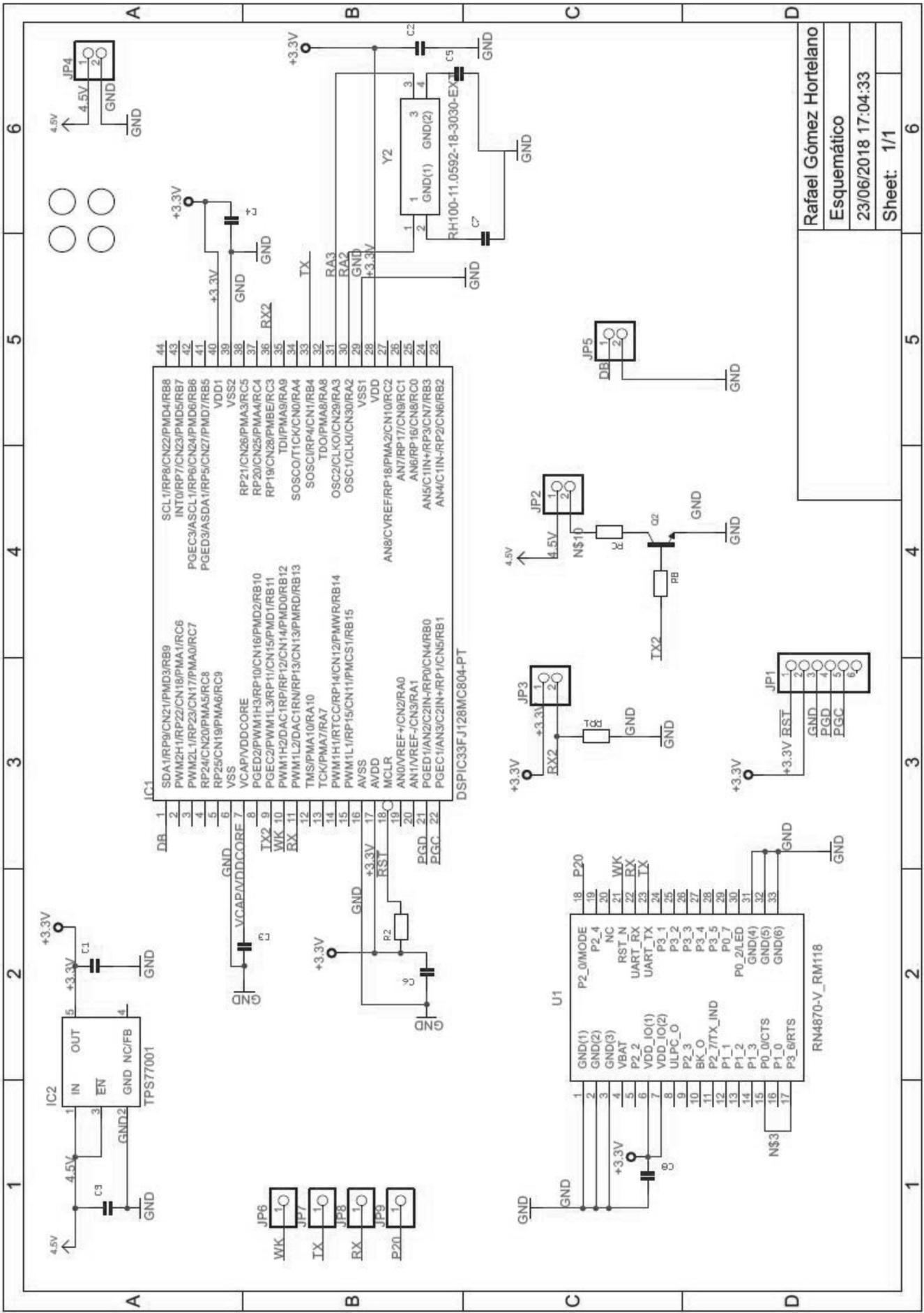
- a. Esquemático
- b. Vista de componentes -- Top
- c. Placa – Top
- d. Vista de componentes -- Bot
- e. Placa – Bot

2) Programas

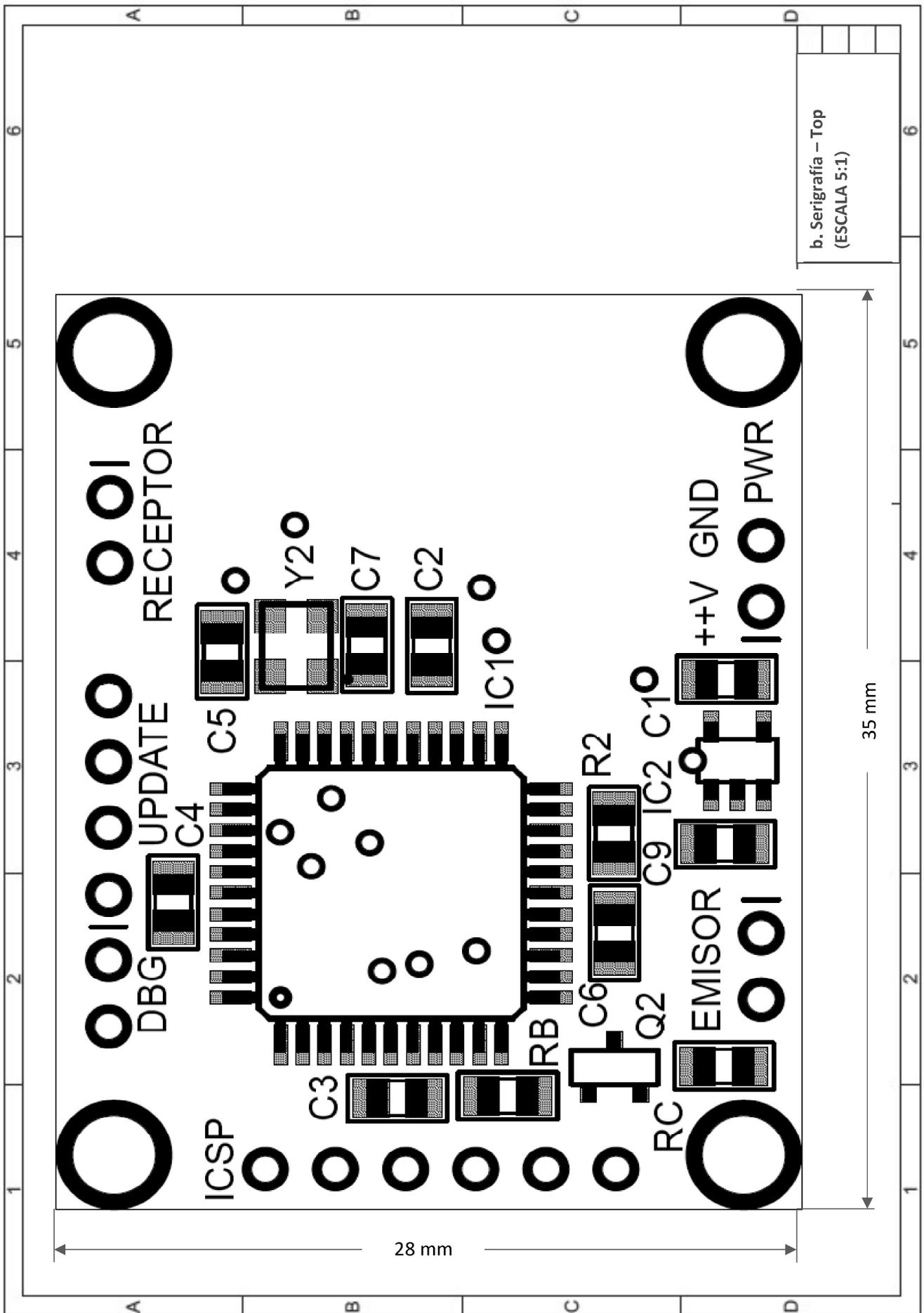
- a. Caja “Maestro/Comunicadora”
- b. Caja “Pareja/Esclavo”
- c. Central

3) Planos de diseño

- a. Caja (Interior)
- b. Caja (Tapa)



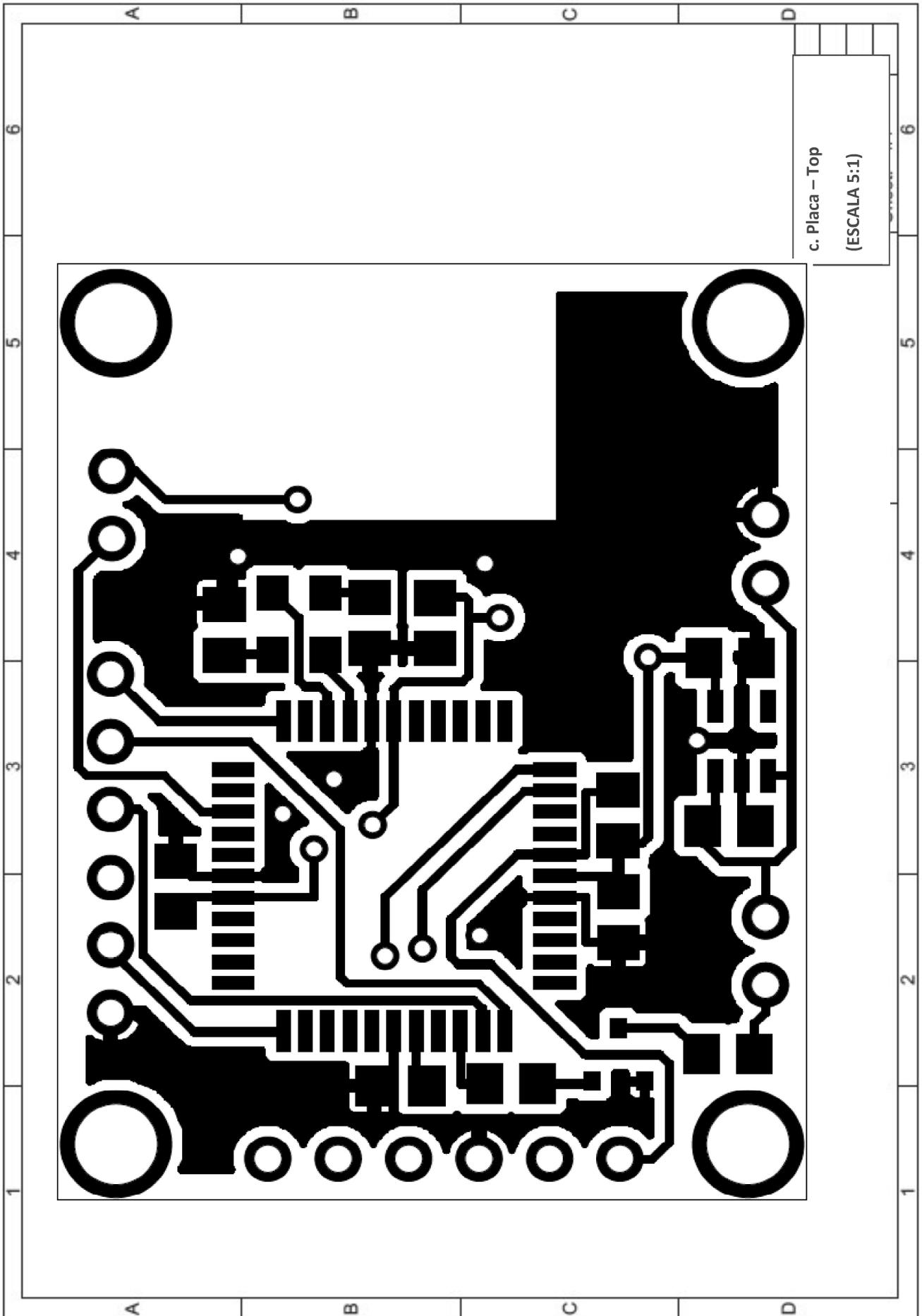
Rafael Gómez Hortelano
 Esquemático
 23/06/2018 17:04:33
 Sheet: 1/1

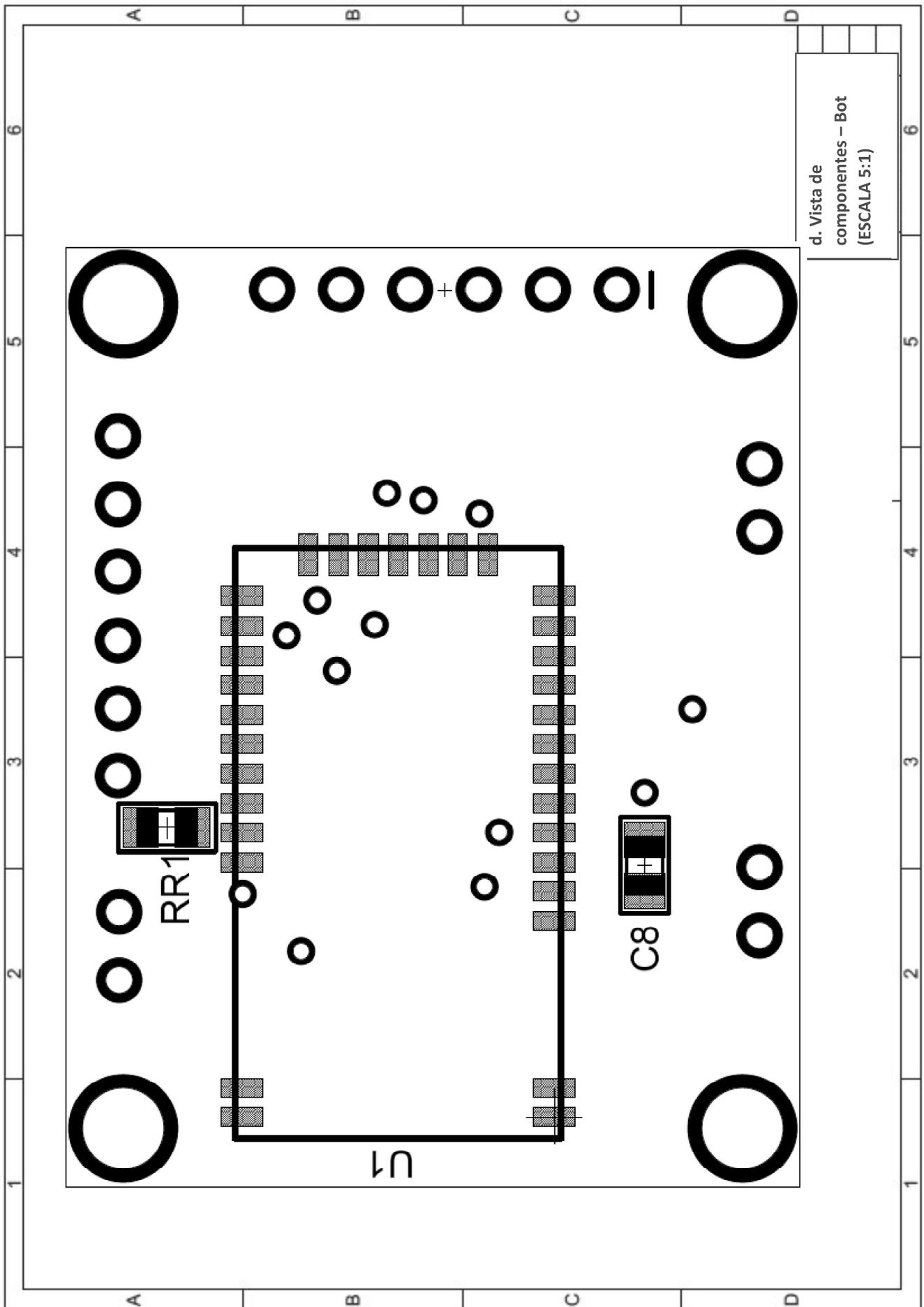


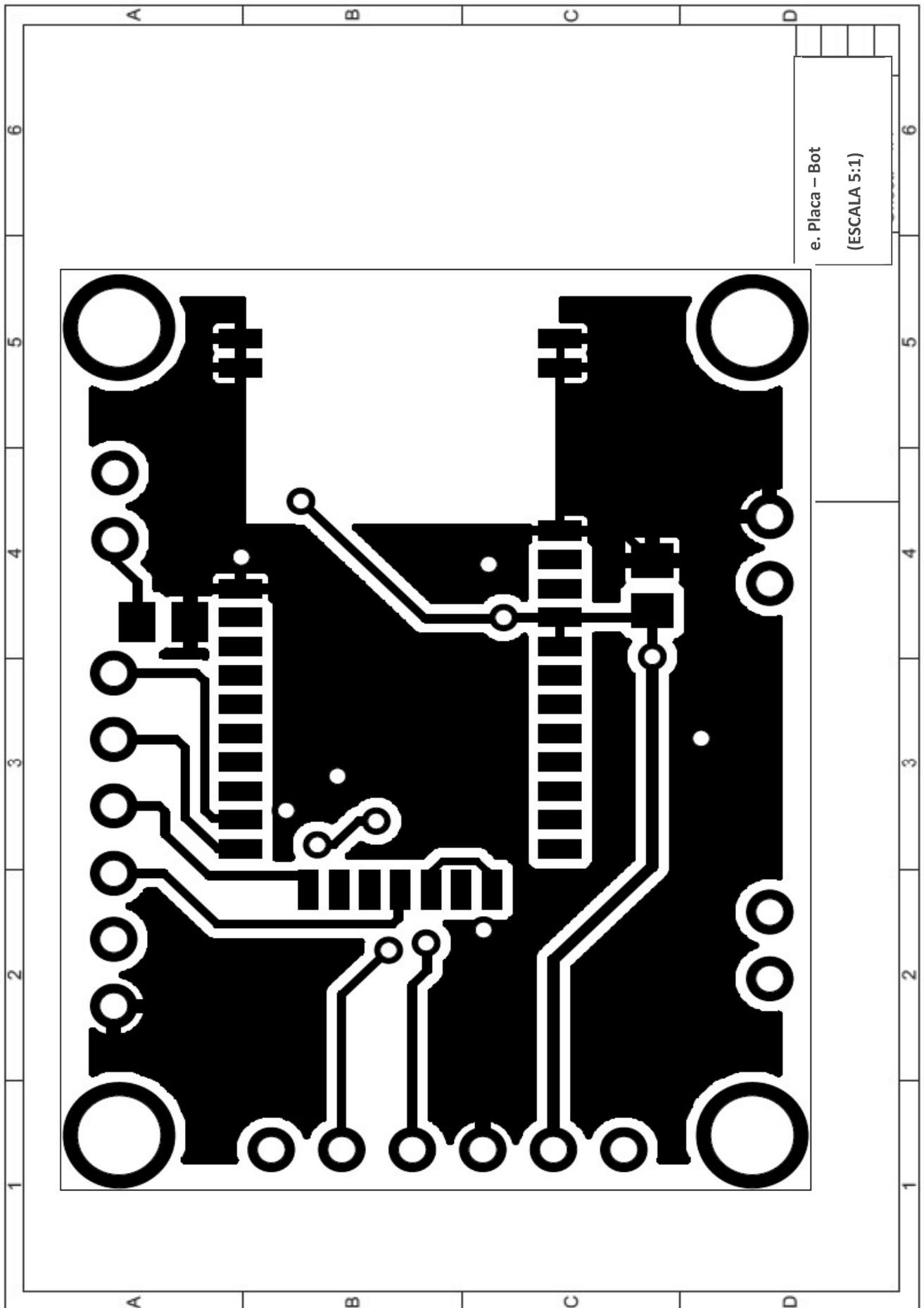
b. Serigrafia - Top
(ESCALA 5:1)

35 mm

28 mm







PROGRAMA CAJA MAESTRO/COMUNICADORA

```
#include "33FJ128MC804.h"
/*****/
#fuses ICSP1 // Puertos de programación
#fuses NOWDT // Sin WDT
#fuses PR // Oscilador primario
#fuses HS // Alta velocidad
/*****/
#use delay (clock = 11059200)

#DEVICE ADC = 10

#define ON(pin) output_high(pin)
#define OFF(pin) output_low(pin)

#pin_select U1TX = PIN_B13
#pin_select U1RX = PIN_B4
#use rs232(UART1, baud=115200, STREAM=BT)

#pin_select U2TX = PIN_B11
#pin_select U2RX = PIN_C3
#use rs232(UART2, baud=300, STREAM=laser)
/*****/
void InitBT(void)
{
    fprintf(BT,"$$"); // Command mode from Data mode
    delay_ms(100);

    fprintf(BT,"Y\r\n"); // Stop advertisement
    delay_ms(500);

    fprintf(BT,"PZ\r\n"); // Stop services
    delay_ms(500);

    fprintf(BT,"IB,Z\r\n"); // Stop Beacon
    delay_ms(500);

    fprintf(BT,"IA,Z\r\n"); // Stop advertising
    delay_ms(500);

    fprintf(BT,"SF,1\r\n"); // Factory Reset
    delay_ms(1000);

    fprintf(BT,"$$"); // Command mode
    delay_ms(100);

    fprintf(BT,"SN,BARRERA\r\n");
    delay_ms(500);

    fprintf(BT,"S.,0000,1EF2F33980D8\r\n");
    delay_ms(300);

    fprintf(BT,"SS,20\r\n"); // Servicio: Beacon
    delay_ms(500);

    fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E00010002C4\r\n");
    delay_ms(500);

    fprintf(BT,"SC,2\r\n"); // Enable (0-beacon, 1-conectable, 2-ambos)
    delay_ms(500);
}
/*****/
int contador = 10;
int corte = 1, corte_ant=0, bateria;
char recibido;
```

```

void main ()
{
  setup_adc(ADC_CLOCK_INTERNAL); // Usa el reloj interno
  setup_adc_ports(sAN2,VSS_VDD); // Configura un convertidor analógico-digital
                                  // en el pin 21 tomando una escala entre 0 y 3,3 V

  set_adc_channel(2); // Selecciona el canal

  InitBT(); // Inicializa el BT

  clear_interrupt(INT_RDA2);

  enable_interrupts(INT_RDA2); // Activación de la recepción

  for(;;)
  {
    delay_ms(100); //10sps
    if(contador>0)
      {
        contador--;
      }
    else
      {
        corte_ant=0; corte = 1;
      }
    bateria = read_adc();

    fputc(0x5A,laser);

    if((corte_ant==1) && (corte==0))
      {
        if(bateria<698 && bateria >628)
          {
            fprintf(BT,"IB,Z\r\n");
            delay_ms(300);
            fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E11110002C4\r\n");
            delay_ms(300);
          }
        if(bateria<628 && bateria >558)
          {
            fprintf(BT,"IB,Z\r\n");
            delay_ms(300);
            fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E90800002C4\r\n");
            delay_ms(300);
          }
        if(bateria<558 && bateria >488)
          {
            fprintf(BT,"IB,Z\r\n");
            delay_ms(300);
            fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E80700002C4\r\n");
            delay_ms(300);
          }
        if(bateria<488 && bateria >349)
          {
            fprintf(BT,"IB,Z\r\n");
            delay_ms(300);
            fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E70500002C4\r\n");
            delay_ms(300);
          }
      }
  }
}

```

```

    }
else if((corte_ant==0) && (corte==1))
{
    if(bateria<698 && bateria >628)
    {
        fprintf(BT,"IB,Z\r\n");
        delay_ms(300);
        fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E11110004C4\r\n");
        delay_ms(300);
    }

    if(bateria<628 && bateria >558)
    {
        fprintf(BT,"IB,Z\r\n");
        delay_ms(300);
        fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E90800004C4\r\n");
        delay_ms(300);
    }
    if(bateria<558 && bateria >488)
    {
        fprintf(BT,"IB,Z\r\n");
        delay_ms(300);
        fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E80700004C4\r\n");
        delay_ms(300);
    }
    if(bateria<488 && bateria >349)
    {
        fprintf(BT,"IB,Z\r\n");
        delay_ms(300);
        fprintf(BT,"IB,FF,4C0002152F234454CF6D4A0FADF2D88039F3F21E70500004C4\r\n");
        delay_ms(300);
    }
    }

    corte_ant=corte;
}

}
/*****/

#int_RDA2
void RDA2_isr()
{
    contador = 10;

    recibido = fgetc(laser);

    if (recibido == 0x5A)
    {
        corte = 0;
    }
    else
    {
        corte = 1;
    }
}

```

PROGRAMA CAJA ESCLAVO/PAREJA

```
#include <33FJ128MC804.h>
/*****/
#fuses NOJTAG // JTAG port is disabled
#fuses NOPROTECT // Code protection is disabled
#fuses NOWRT // Writes to program memory are allowed
#fuses ICSP1 // Emulator functions are shared with PGEC1/PGED1
#fuses NOWDT // Watch Dog Timer OFF
#fuses NOIESO // Internal External Switchover bit
#fuses PR // PR:Primary crystal oscillator
#fuses CKSN0FSM // Clock switching enabled,Fail-Safe Clock Monitor disabled
#fuses OSCIO // OSCO Pin Configuration bit
#fuses HS // Primary Oscillator Configuration bits: Primary Oscillator mode is disabled
/*****/
#use delay(clock=11059200)

#pin_select U2TX = PIN_B11
#pin_select U2RX = PIN_C3
#use rs232(UART2, baud=300, STREAM=laser)

#define ON( pin) output_high(pin)
#define OFF(pin) output_low( pin)

char c, recibido=0;
/*****/
void main()
{
    clear_interrupt(INT_RDA2);
    enable_interrupts(INT_RDA2); // Activación de la recepción

    for(;;)
        {
            if(recibido==1)
                {
                    delay_ms(40);
                    fputc(c,laser);
                    recibido=0;
                }
        }
/*****/
#int_RDA2
void RDA2_isr()
{
    recibido = 1;
    c = fgetc(laser);
}
```

PROGRAMA CAJA CENTRAL

```
#include "33FJ128MC804.h"
/*****/
#fuses NOJTAG // JTAG port is disabled
#fuses NOPROTECT // Code protection is disabled
#fuses NOWRT // Writes to program memory are allowed
#fuses ICSP1 // Emulator functions are shared with PGEC1/PGED1
#fuses NOWDT // Watch Dog Timer OFF
#fuses NOIESO // Internal External Switchover bit
#fuses FRC // PR:Primary crystal oscillator
#fuses CKSNOSFM // Clock switching enabled,Fail-Safe Clock Monitor disabled
#fuses OSCIO // OSCO Pin Configuration bit
#fuses NOPR // Primary Oscillator Configuration bits: Primary Oscillator mode is disabled
/*****/
#use delay (clock = 7370000)

#define ON(pin) output_high(pin)
#define OFF(pin) output_low(pin)

#pin_select U1TX = PIN_B11
#pin_select U1RX = PIN_B10

#use rs232(UART1, baud=115200, STREAM=BT)

#byte U1RXREG = 0x0226

#word U1STA = 0x0222

unsigned int8 trama[256];
/*****/
void InitBT(void)
{
    fprintf(BT,"$$$"); // Command mode from Data mode
    delay_ms(100);

    fprintf(BT,"$$$");
    delay_ms(300);

    fprintf(BT,"Y\r\n"); // Stop advertisement
    delay_ms(500);

    fprintf(BT,"PZ\r\n"); // Stop services
    delay_ms(500);

    fprintf(BT,"IB,Z\r\n"); // Stop Beacon
    delay_ms(500);

    fprintf(BT,"IA,Z\r\n"); // Stop advertising
    delay_ms(500);

    fprintf(BT,"SF,1\r\n"); // Factory Reset
    delay_ms(1000);

    fprintf(BT,"$$$"); // Command mode
    delay_ms(100);

    fprintf(BT,"SN,CENTRAL\r\n");
    delay_ms(500);

    fprintf(BT,"SC,2\r\n"); // Enable (0-beacon, 1-conectable, 2-ambos)
    delay_ms(500);
}
```

```

    fprintf(BT,"SR,0C00\r\n"); // NO-filtro; Solo No-conectables
    delay_ms(500);

    fprintf(BT,"JC\r\n");
    delay_ms(500);

    fprintf(BT,"JA,1,FOA1B4002084\r\n");
    delay_ms(500);

    fprintf(BT,"JD\r\n");
    delay_ms(1000);

}
/*****/

unsigned int16 contador=0;
unsigned int8 n=0;
/*****/

void main ()

{
    ON(PIN_A0);
    ON(PIN_B2);
    ON(PIN_B3);
    ON(PIN_C0);
    ON(PIN_C1);
    ON(PIN_C2);
    ON(PIN_A1);

    memset (trama,'A',256);

    clear_interrupt(int_rda);

    enable_interrupts(int_rda);

    InitBT(); // Inicializa el BT

    fprintf(BT,"F\r\n");

    while (1)
        {
            delay_ms(1);

            if((contador%1000)==0) { output_toggle(PIN_A1); } // Enciende un LED para saber si está
            funcionando

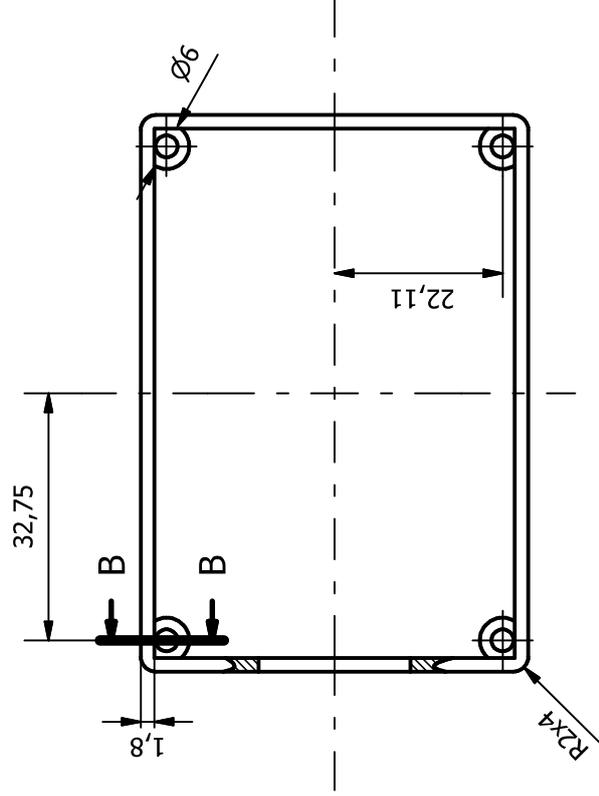
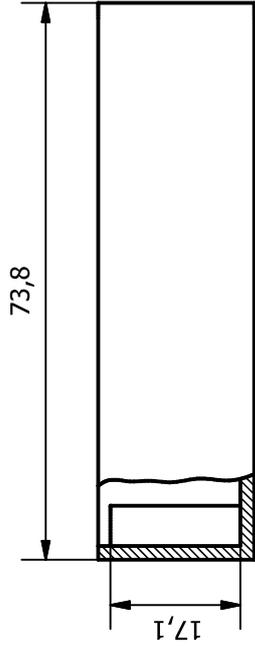
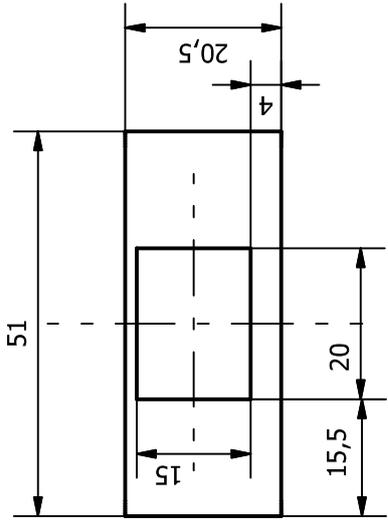
            if(contador++ >10000)
                {
                    contador = 0;
                    fprintf(BT,"F\r\n");
                }
        }
}
/*****/

```

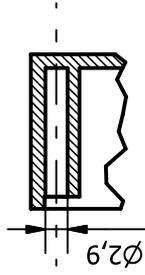
```

#int_rda
void isr_rda(void)
{
  trama[n++]=fgetc(BT);
  if(trama[n-1]!='%')
  {
    if(trama[n-80]!='%')
    {
      if(trama[n-4]=='2')
      {
        ON(PIN_C1); // Rojo, NO hay corte
        OFF(PIN_B3);
      }
      if(trama[n-4]=='4')
      {
        OFF(PIN_C1); // Verde, hay corte
        ON(PIN_B3);
      }
    }
  }
}

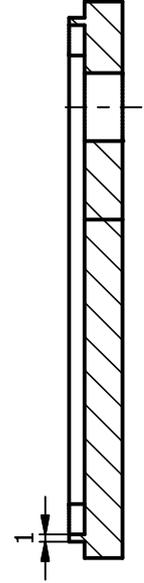
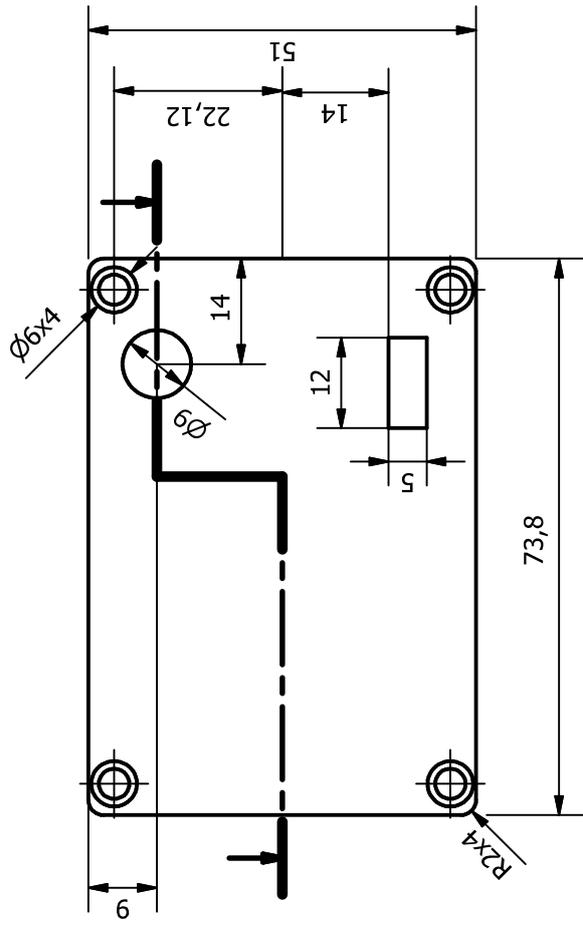
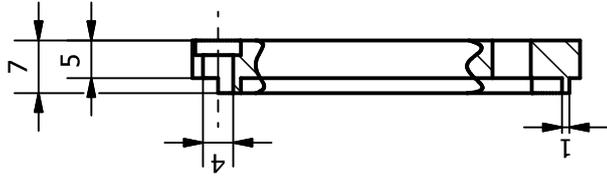
```



B-B (1 : 1)



Diseño de PC-RAFAEL	Revisado por	Aprobado por	Fecha 18/06/2018	Fecha	Edición	Hoja 1 / 1
			Caja (interior)		ESCALA 1:1	



Diseño de Rafael Gómez	Revisado por	Aprobado por	Fecha 21/06/2018	Fecha	Edición	Hoja 1 / 1
			TAPA		ESCALA 1:1	