

UNIVERSIDAD DE OVIEDO

CENTRO INTERNACIONAL DE POSTGRADO

MÁSTER EN INGENIERÍA MECATRÓNICA

TRABAJO FIN DE MÁSTER

Diseño, implementación e integración de equipo para la medida, digitalización, almacenamiento y posterior volcado de señales de navegación interior para robots exploradores.

JULIO 2013

Oscar Rivera Blázquez

Fernando Nuño García

Juan Carlos Álvarez Álvarez

AGRADECIMIENTOS

Un trabajo de la magnitud de un Proyecto de Fin de Máster no solo requiere de una gran cantidad de tiempo y dedicación, sino que también requiere de la ayuda de personas cuyas aportaciones y apoyo han hecho posible el desarrollo de este proyecto.

Me gustaría expresar mi agradecimiento a los tutores de este Trabajo de Fin de Máster, Fernando Nuño García y Juan Carlos Álvarez Álvarez por la dedicación y apoyo mostrados durante el desarrollo del proyecto. Además quiero expresar mi agradecimiento a Daniel Fernández Villanueva, proyectante del Área de Ingeniería de Sistemas y Automática, por su aportación en el presente proyecto.

Agradezco también la colaboración de la empresa TUBECHECK.SL por su aportación con el robot explorador teniendo en cuenta la limitada disponibilidad con la que contaban.

Por último deseo también dar las gracias a mi familia, a mi novia y a mis amigos, dado que durante el transcurso de estos 6 meses de trabajo han sabido darme el apoyo necesario para afrontar los baches y obstáculos que he ido encontrando en el camino.

RESUMEN

Este Trabajo de Fin de Máster (TFM) presenta el diseño, fabricación e integración de un sistema mecatrónico capaz de almacenar los datos de aceleración, velocidad angular y campo magnético a los que está sometido un robot explorador ROVVER 225 de la marca I-PEK durante un ciclo de trabajo. Dicho robot pertenece a la empresa TUBECHECK SL.

El presente proyecto pertenece al Máster en Ingeniería Mecatrónica de la Universidad de Oviedo y forma parte del proyecto “ACRISA: prototipo de equipo registrador de trayectorias”, el cual se desarrolla con la colaboración de dos áreas de conocimiento de la Universidad de Oviedo, ISA (Ingeniería de Sistemas y Automática) y TE (Tecnología Electrónica).

Los datos almacenados procedentes de los sensores inerciales serán utilizados con posterioridad para su procesado en un computador mediante un algoritmo que permitirá determinar la trayectoria seguida por el robot explorador. El equipo desarrollado permitirá la transferencia de esos datos al computador, el procesamiento de los mismos no es objetivo de este trabajo.

El presente proyecto tiene un claro carácter mecatrónico puesto que abarca, en mayor o menor medida, las tres disciplinas: electrónica, mecánica y control.

PALABRAS CLAVE

Robot - Microcontrolador – Unidad de Medida Inercial (IMU) – Sensor - PCB

ÍNDICE GENERAL

1.	INTRODUCCIÓN.....	1
1.1.	PLANTEAMIENTO.....	1
1.2.	OBJETIVO.....	1
1.3.	ALCANCE.....	2
1.4.	PETICIONARIO.....	2
1.5.	REQUISITOS.....	2
2.	ANTECEDENTES Y CONOCIMIENTOS PREVIOS.....	3
2.1.	SENSORES INERCIALES.....	3
2.2.	CÁLCULO DE TRAYECTORIAS.....	4
3.	DISEÑO DEL SISTEMA.....	5
3.1.	DISEÑO ELECTRÓNICO.....	5
3.1.1.	<i>Introducción.....</i>	5
3.1.2.	<i>Esquema general y principales componentes.....</i>	5
3.1.3.	<i>Selección de componentes Comerciales.....</i>	7
3.1.4.	<i>Circuito Esquemático.....</i>	14
3.1.5.	<i>Tarjeta de circuito impreso.....</i>	16
3.2.	DISEÑO MECÁNICO.....	17
3.2.1.	<i>Introducción.....</i>	17
3.2.2.	<i>Estudio del robot explorador.....</i>	17
3.2.3.	<i>Alternativas de diseño.....</i>	18
3.2.4.	<i>Descripción de la solución adoptada.....</i>	21
4.	FABRICACIÓN DEL PROTOTIPO.....	25
4.1.	FABRICACIÓN DEL SISTEMA ELECTRÓNICO.....	25
4.2.	FABRICACIÓN DEL SOPORTE.....	26
5.	CÁLCULOS.....	27
5.1.	CONVERSIÓN DE DATOS.....	27
5.1.1.	<i>Acelerómetro.....</i>	27
5.1.2.	<i>Giróscopo.....</i>	27
5.1.3.	<i>Magnetómetro.....</i>	27
5.2.	TIMER0.....	28
5.3.	TENSIÓN DE REFERENCIA DEL MÓDULO COMPARADOR.....	28
5.3.1.	<i>Tensión de referencia.....</i>	28
5.3.2.	<i>Divisor de tensión.....</i>	28
5.4.	CONSUMO DE CORRIENTE.....	29
6.	COSTES.....	31
7.	ANEXO I: ESPECIFICACIONES TÉCNICAS.....	35
7.1.	ACELERÓMETRO ADXL345.....	35
7.2.	GIRÓSCOPO ITG-3200.....	37
7.3.	MAGNETÓMETRO HMC-5883L.....	38
7.4.	RELOJ TERMÓMETRO DS1629.....	40

7.5.	IMU MT1-28A53G25.....	42
7.5.1.	<i>Sistema de comunicación</i>	42
7.5.2.	<i>Conector</i>	42
7.5.3.	<i>Protocolo de comunicación</i>	43
7.5.4.	<i>Configuración inicial</i>	43
7.6.	TARJETA SD.....	46
7.6.1.	<i>Sistema de comunicación</i>	46
7.6.2.	<i>Conector</i>	46
7.6.3.	<i>Protocolo de comunicación</i>	47
7.6.4.	<i>Configuración inicial</i>	47
7.6.5.	<i>Sistema de archivos FAT</i>	49
7.7.	MICROCONTROLADOR 18F27J13	50
7.7.1.	<i>Tabla de pines del microcontrolador</i>	50
8.	ANEXO II: MANUAL DEL PROGRAMADOR	51
8.1.	INTRODUCCIÓN.....	51
8.2.	ESTRUCTURA Y FUNCIONALIDADES	51
8.3.	MÓDULOS INTERNOS.....	52
8.4.	PROGRAMA PRINCIPAL.....	53
8.5.	FORMATO DE LOS DATOS ALMACENADOS.....	54
8.6.	INTERRUPCIONES.....	55
8.7.	RUTINAS Y FUNCIONES	55
8.8.	DIAGRAMAS DE FLUJO	58
9.	ANEXO III: MANUAL DEL USUARIO	63
9.1.	UBICACIÓN DE LOS COMPONENTES.....	63
9.2.	PUESTA EN MARCHA	64
9.3.	CARGAR BATERÍA.....	66
9.4.	COMUNICACIÓN CON PC	67
9.5.	PANTALLA DE EXPLOTACIÓN ISA.....	69
10.	ANEXO IV: ENSAYOS Y PRUEBAS	73
10.1.	PRUEBA DE ESTANQUEIDAD.....	73
10.2.	ERROR ENTRE LOS SENSORES INERCIALES Y EL IMU	73
10.3.	TRAZADO DE UNA TRAYECTORIA A PARTIR DE DATOS MUESTREADOS.....	77
11.	CONCLUSIONES.....	79
12.	BIBLIOGRAFÍA.....	81
13.	PLANOS	82
13.1.	ÍNDICE DE PLANOS.....	82

1. INTRODUCCIÓN

A continuación se presentan los apartados iniciales que explicarán y concretarán las características principales del proyecto.

1.1. Planteamiento

La empresa TUBECHECK SL (Oviedo) se dedica a la exploración e inspección de tuberías de saneamiento subterráneas. Para ello cuentan con una serie de robots exploradores que son capaces de moverse por el interior de dichas tuberías grabando una señal de video en tiempo real. Cuando se detecta una anomalía es necesario alcanzar la tubería en el punto defectuoso y repararla. Para ello es necesario cavar una zanja en el punto correcto, para lo cual solo se cuenta con la medida de la distancia recorrida por el robot para localizar dicho punto, lo que da lugar a una incertidumbre a la hora de realizar la excavación.

Dado que la zona de trabajo se encuentra bajo tierra, no es posible realizar un seguimiento del robot vía GPS. Será necesario por tanto realizar una medida en tiempo real de los datos más relevantes a los que está sometido el robot durante su ciclo de trabajo, y almacenarlos en un soporte móvil para procesarlos posteriormente.

Gracias a estos datos se puede estimar un mapa de la red de conductos y canalizaciones por las que circula el robot explorador. De esta manera se podría determinar un esquema cartográfico más preciso del que se puede observar en los planos teóricos de la red de canalizaciones, detectando así cualquier posible anomalía o cambio con respecto a estos.

Una vez estimado el mapa de la red, el proceso de excavación puede realizarse de manera más precisa y fiable, reduciendo tiempo y costes.

1.2. Objetivo

El objetivo de este proyecto será diseñar un sistema electrónico capaz de desarrollar las funciones mencionadas en el apartado anterior y fabricar una caja de protección que incluya dicho sistema para integrarlo y acoplarlo al robot explorador.

El sistema electrónico deberá ser totalmente independiente del sistema electrónico del robot explorador y el soporte de integración no deberá influir o entorpecer en el correcto funcionamiento del robot en su rutina de trabajo.

En el presente proyecto se exponen todos los aspectos técnicos relacionados con el diseño de la parte electrónica y mecánica del sistema, cálculos justificados y costes del sistema.

1.3. Alcance

Este proyecto abarcará todos los puntos relacionados con el diseño, fabricación e integración del sistema y su puesta en funcionamiento. Se limitará a guardar los datos obtenidos en un soporte de almacenamiento removible en el formato que más se ajuste a nuestras necesidades. El posterior volcado, estudio e interpretación de los datos obtenidos será desarrollado por el Área de Ingeniería de Sistemas y Automática.

1.4. Peticionario

Máster en Ingeniería Mecatrónica y empresa TUBECHECK SL.

1.5. Requisitos

A continuación se detallarán los requisitos mínimos que debe reunir el prototipo diseñado. Dichos requisitos se numerarán con un identificador único e inequívoco para poder referirse a ellos si fuera necesario.

Nº	DESCRIPCIÓN
1	El prototipo debe ser capaz de almacenar los datos de aceleración, velocidad angular y campo magnético durante un ciclo de trabajo del robot explorador.
2	El sistema debe tener una capacidad de almacenamiento de datos de al menos 1 hora .
3	El sistema electrónico del prototipo será totalmente independiente del robot explorador.
4	El sistema de sujeción de la caja será totalmente independiente del robot explorador.
5	El sistema debe contar con una fuente de alimentación que empleará como energía primaria la procedente de una batería recargable que permita una autonomía de al menos 1 hora .
6	El prototipo debe poder integrarse en la carcasa exterior del robot explorador sin entorpecer o dificultar el movimiento del mismo.
7	La carcasa del prototipo debe ser totalmente estanca debido a las condiciones de trabajo.

Tabla 1.1. Tabla de requisitos

2. ANTECEDENTES Y CONOCIMIENTOS PREVIOS

Para poder abordar el diseño electrónico del sistema es necesario adquirir una serie de conocimientos previos relacionados con los sensores inerciales que se van a utilizar.

También es conveniente, aunque no está recogido en el alcance del proyecto, hacer una breve introducción al cálculo de trayectorias a partir de los datos de los sensores inerciales.

2.1. Sensores inerciales

Un sensor inercial es un dispositivo capaz de medir una magnitud en función de la inercia a la que esté sometido. Típicamente cuando se habla de un sensor de inercia suele referirse a un acelerómetro o a un giróscopo.

Un acelerómetro es un sensor capaz de medir aceleración o un movimiento de translación, mientras que un giróscopo es un sensor que mide velocidad angular. En la actualidad podemos encontrar en el mercado acelerómetros y giróscopos de tipo MEMS (Micro Electro Mechanical Systems) con múltiples características y funcionalidades (salidas digitales, interrupciones, rango de medida configurable...).

Se pueden diferenciar cinco tipos de movimiento que estos dos sensores pueden medir. Translación, vibración, impacto, ladeo y rotación. Salvo la rotación, los cuatro tipos de movimientos restantes son diferentes manifestaciones de la aceleración.

Un movimiento de translación implica un cambio en la posición de un objeto que se producirá a una determinada velocidad, un cambio de velocidad define una aceleración, a una aceleración negativa se le denomina desaceleración. Si se observa una aceleración y desaceleración que se producen a un periodo concreto podemos hablar de vibración, mientras que si se produce en un único momento determinado y en un corto intervalo de tiempo hablamos de impacto. El ladeo se puede detectar cuando un movimiento implica un cambio respecto a la fuerza de gravedad a la que está sometido un objeto respecto a un eje. Todos estos movimientos se miden en Gs y el propio sensor los convierte en voltios (analógico) o en bits (digital).

El quinto tipo de movimiento (rotación) no puede medirse con un acelerómetro, puesto que un objeto que gira a una velocidad angular constante respecto a un eje no modifica su aceleración.

Este tipo de sensores pueden proporcionar medidas erróneas debido a problemas de offset y ruido. El offset puede deberse a dos problemas principalmente: la temperatura y la propia constitución mecánica del sensor. La primera es una fuente de error variable dependiente de la temperatura, mientras que la segunda es fija y depende del propio sensor. Ambos se pueden compensar con mayor o menor dificultad. El ruido suele ser atenuado por los filtros que incorpora el propio sensor.

2.2. Cálculo de trayectorias

A partir de los datos que proporcionan los sensores inerciales es posible realizar una estimación de la trayectoria de un objeto. Mediante la odometría, estudio de la estimación de posición de un objeto a partir de la integración de información incremental en función del tiempo, es posible calcular la posición final de un objeto y su orientación a partir de la integración de los datos del acelerómetro y del giróscopo. Sin embargo, la odometría cuenta con un problema inherente a su definición ya que la integración de los datos inerciales conlleva a la acumulación de un error que aumenta cuantas más muestras se integren.

Para evitar o disminuir este problema, es necesario incluir referencias externas que nos permitan comparar los datos medidos con una referencia real. De esta manera se podría corregir la trayectoria cada cierto número de muestras reseteando el error acumulado.

3. DISEÑO DEL SISTEMA

En este capítulo se describen los pasos acometidos en las tres partes del diseño del sistema: Electrónico, mecánico y de software. Para su desarrollo se han utilizado programas de diseño electrónico y mecánico para dimensionar el prototipo antes de pasar a la etapa de fabricación.

El diseño comenzó con la parte electrónica, con la fabricación y puesta en marcha de un prototipo de tarjeta electrónica sobredimensionada para probar los componentes seleccionados y realizar el programa de control.

El diseño mecánico del soporte y la tarjeta electrónica final no pudieron ser dimensionadas hasta que no se dispuso del robot explorador. Dada la falta de información por parte del fabricante de las especificaciones técnicas, fue necesario solicitar a la empresa TUBECHECK.SL el robot explorador para proceder al dimensionamiento del prototipo final.

3.1. Diseño electrónico

3.1.1. INTRODUCCIÓN

Para cumplir los requisitos del proyecto, desarrollaremos una placa de circuito impreso donde se integrarán todos los componentes necesarios. Dicha placa se ubicará en el interior del soporte para fijar el sistema al robot.

Tras disponer del robot explorador y una vez evaluadas las opciones, se opta por realizar dos placas que irán montadas una encima de la otra, conectadas a través de cables. Esta opción permitirá una considerable reducción de la superficie de la caja

El presente trabajo da soporte a un proyecto de investigación y desarrollo, por lo que se incluirá un componente de medida inercial comercial (IMU) junto a un componente de medida inercial compuesto por sensores individuales. De esta manera se contará con dos fuentes de información, pudiendo comparar los datos de un sistema comercial totalmente funcional con uno en desarrollo y evaluar así las prestaciones que se pueden alcanzar.

3.1.2. ESQUEMA GENERAL Y PRINCIPALES COMPONENTES

A continuación se detallan los principales componentes que se integrarán en la placa de circuito impreso:

- **Microcontrolador:** Será necesario para llevar a cabo la rutina de recogida y almacenamiento de datos procedentes de los sensores inerciales a la tarjeta SD. Se alimentará a 3.3v para simplificar la conexión con la tarjeta SD.
- **Sensores inerciales:** Componentes encargados de obtener las medidas inerciales correspondientes y enviárselas al microcontrolador a través del protocolo de comunicación I2C. Finalmente se opta por una única tarjeta que integra los sensores inerciales necesarios.

- Reloj/Termómetro: Sensor encargado de iniciar una cuenta en segundos desde el inicio del ciclo de trabajo para sincronizar la tabla de datos con la señal de video. Además se ocupará de medir la temperatura a la que está sometido el sistema. Se comunican con el microcontrolador a través del protocolo de comunicación I2C.
- IMU: Componente comercial adicional que incluye sensores inerciales. Con este dispositivo se podrá comparar los datos obtenidos con los sensores inerciales individuales. Se comunican con el microcontrolador a través del protocolo de comunicación asíncrono RS232.
- Tarjeta SD: Soporte de almacenamiento removible para poder transmitir los datos desde el sistema de medida inercial a un PC.
- Puerto serie: Componente con conexión USB para conectar el microcontrolador a un PC. De esta manera se podrá manejar el sistema a través de un interfaz de usuario desde el PC.
- Batería recargable: Fuente de energía primaria del sistema. Debe ser capaz de suministrar la corriente eléctrica que demandan todos los componentes del circuito.
- Regulador de tensión: Componente que adapta la tensión de la batería a 3.3v y es capaz de suministrar la corriente que se demanda. Necesario para alimentar el microcontrolador, la tarjeta SD y reloj/termómetro
- Convertidor DC/DC: Componente que adapta la tensión de la batería a 5v y es capaz de suministrar la corriente que se demanda. Necesario para alimentar el IMU y la tarjeta de sensores.

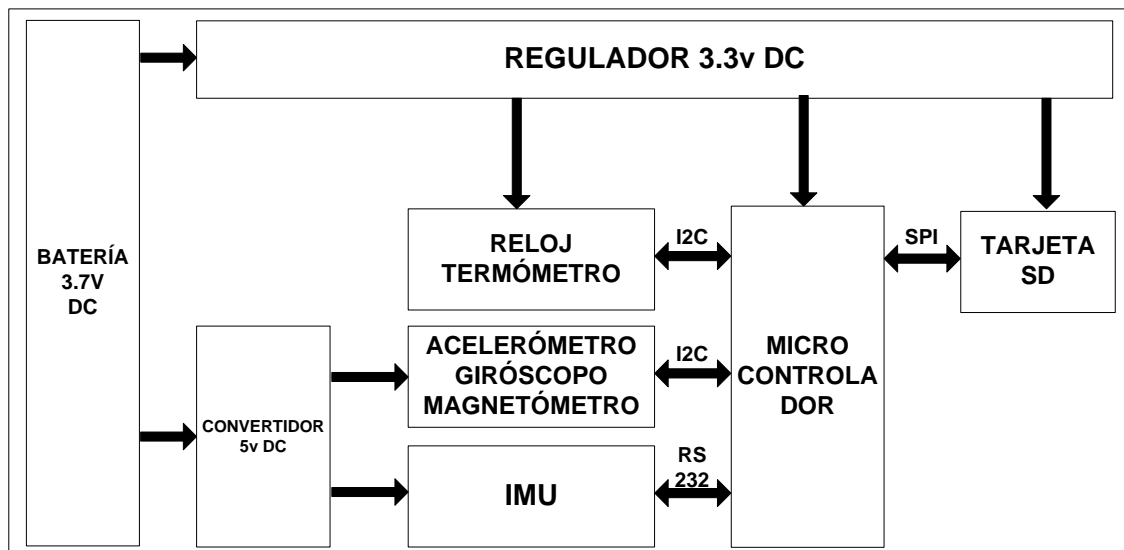


Figura 3.1. Diagrama de bloques del circuito electrónico

3.1.3. SELECCIÓN DE COMPONENTES COMERCIALES

A continuación se detallan los principales componentes comerciales incluidos en el diseño.

Sensores

Una unidad de medida inercial está equipada con un acelerómetro, un magnetómetro y un giróscopo. Para simplificar el diseño, se buscarán sensores cuya salida sea digital y que se comuniquen con un protocolo admitido por un microcontrolador.

Seleccionamos una tarjeta comercial que incluye los tres sensores que necesitamos con las características requeridas. Dichos sensores trabajan a una tensión de 3.3v y se comunican a través del protocolo síncrono I2C.

Se trata del modelo SEN-10724 de la marca SPARKFUN ELECTRONICS. Dicha tarjeta cuenta con 4 pines, VCC, GND, SDA y SCL. Además lleva integrado un regulador de tensión para alimentar la tarjeta a 5V y las resistencias de PULL-UP necesarias para la comunicación I2C.

Cada sensor puede ser configurado para cambiar su rango de medida, resolución (en algún caso) y frecuencia de muestreo. La configuración de éste último no marcará realmente la frecuencia de muestreo del sistema, puesto que esta la marcará el microcontrolador cuando recoja valores a través de la comunicación I2C.

Algunos sensores tienen dos modos de medida: modo de medida continuo o simple. Para esta aplicación se utilizará el modo continuo, tomando así muestras continuamente a la frecuencia configurada; pero sólo se recogerán las que coincidan con la frecuencia de muestreo que establecerá el microcontrolador.

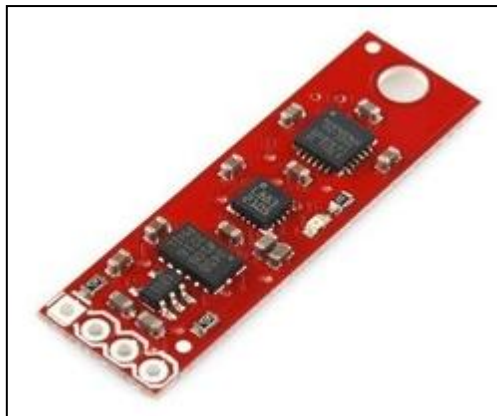


Figura 3.2. Tarjeta SEN-10724 de SPARKFUN

Para ubicar en el tiempo cada muestra y así poder comparar los datos almacenados con la señal de video del robot explorador será necesario incluir un reloj en la tarjeta de circuito impreso.

Se incluirá un termómetro para detectar la proximidad de conductos cuya temperatura sea superior o inferior a la temperatura ambiente; de esta manera si se prevé que la trayectoria del robot pasará cerca de uno de estos conductos, el cambio de temperatura se almacenará dando como resultado un punto de referencia a la hora de comprobar los resultados obtenidos.

Esta tarjeta no incorpora ningún reloj o termómetro, por lo que será necesario buscar un componente que cumpla estas funciones. Seleccionamos el reloj DS1629 de la marca DALLAS SEMICONDUCTOR. Este componente se incorporará en la placa principal.

Las características técnicas, descripción y configuración de los sensores están detalladas en el capítulo ANEXO I del presente documento.

IMU

Para realizar una comparación con los datos obtenidos de los sensores y así tener una segunda fuente de información contamos con una unidad de medida inercial (IMU) de la marca XSENS. Concretamente el modelo MTi-28A53G25.

Se trata de un componente que cuenta en su interior con un acelerómetro, un giróscopo y un magnetómetro. Además gracias a un procesador digital de señal dsPIC que incorpora es capaz de analizar la información obtenida de sus sensores para calcular los ángulos de navegación (roll, pitch y yaw) del IMU.

Este dispositivo trabaja con una alimentación de 5V y un consumo de 70mA en régimen permanente.



Figura 3.3. IMU modelo MTi-28A53G25

Las características técnicas, tipo de conexión, configuración y protocolo de comunicación del IMU están detalladas en el capítulo ANEXO I del presente documento.

TARJETA SD

Para almacenar los datos obtenidos será necesario un soporte removible con capacidad suficiente y compatible con un microcontrolador. Las tarjetas SD cumplen las características que se necesitan para este proyecto. Cuentan con un gran rango de capacidades, en este caso una capacidad de 2Gb será suficiente, y se puede comunicar a través del protocolo síncrono SPI, el cual podemos implementar en un microcontrolador.

Existen varias clases de tarjetas SD dependiendo de su tamaño. Para este caso seleccionaremos el tipo microSD puesto que son muy comunes y compatibles con los móviles de última generación, pudiendo, si fuera necesario, volcar los datos obtenidos en uno de estos dispositivos.

Se selecciona una tarjeta micro SD de la marca Transcend de 2 GB por cuestión de accesibilidad.



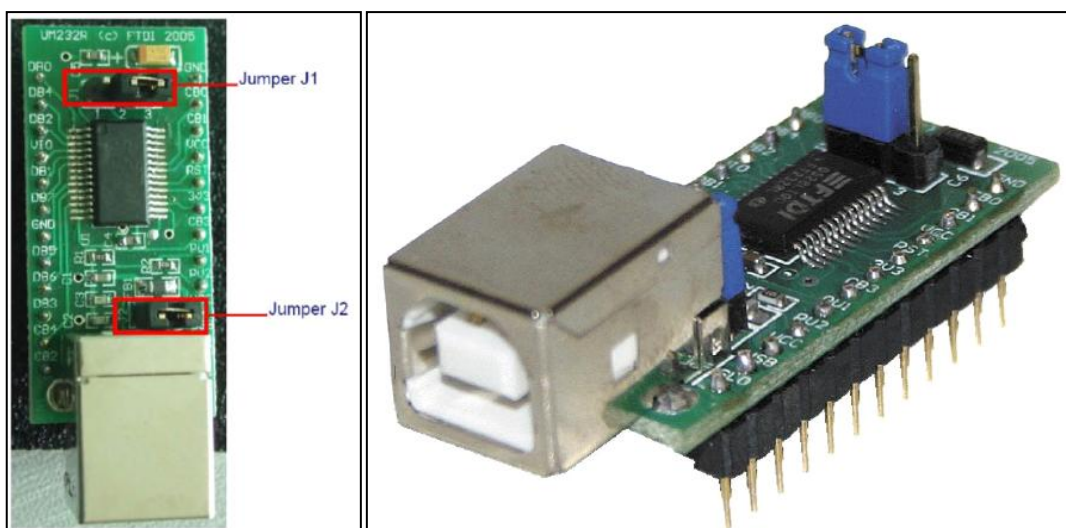
Figura 3.4. Tarjeta micro SD

Las características técnicas, tipo de conexión, configuración y protocolo de comunicación de la tarjeta SD están detalladas en el capítulo ANEXO I del presente documento.

PUERTO SERIE USB

Se incluye en el diseño un sistema capaz de conectar el microcontrolador a un PC. A través de una pantalla de explotación será posible sincronizar el reloj del sistema y volcar los datos de la tarjeta SD directamente, sin necesidad de sacar la tarjeta. Será posible además crear y borrar archivos en la tarjeta SD.

La comunicación entre el PC y el microcontrolador será asíncrona. Se utilizará el dispositivo UM232R USB de la marca FTDI CHIP. Este componente cuenta con una clavija hembra USB tipo-B en un extremo y una base de 24 pines. Sólo se usarán 4 pines, 1(TX) ,5(RX) y 7,24(GND). Gracias a dos JUMPERS (J1 y J2) que tiene en la parte superior puede configurarse para que la comunicación con el microcontrolador se realice a niveles TTL y para que se alimente a través del PC (5v). La velocidad de transmisión será de 9600 baudios.



Figuras 3.5. Y Figura 3.6. Puerto serie UMR232R

MICROCONTROLADOR

Seleccionaremos un microcontrolador que realizará las funciones de adquisición de datos de los sensores y su almacenamiento en la tarjeta SD. Para ello necesitaremos que cumpla las siguientes características:

- 2 módulos MSSP, para poder implementar la comunicación con los sensores (I2C), y con la tarjeta SD (SPI).
- Tensión de alimentación de 3.3V, puesto que es la tensión de trabajo de la tarjeta SD y de la mayoría de los componentes de la tarjeta.
- Capaz de admitir un oscilador externo de al menos 40MHz, para agilizar las operaciones del microcontrolador.
- Encapsulado pequeño, ya que no necesitaremos un gran número de pines y así minimizamos el tamaño de la tarjeta de circuito impreso.

Seleccionamos un microcontrolador de la marca MICROCHIP que se ajusta a estas características (PIC18F27J13-I/SP). Dicho micro se alimenta a 3.3V, cuenta con 2 módulos MSSP, admite un oscilador de 40MHz a 3.3V, y tiene un encapsulado DIP de 28 pines.



Figura 3.7. PIC 18F27j13

Necesitaremos implementar 4 tipos diferentes de comunicación con el microcontrolador, 2 síncronas (SPI y I2C) para comunicarnos con los sensores y la tarjeta SD, y 2 asíncronas para comunicarnos con el IMU y el PC.

Las características técnicas y conexión de cada pin del microcontrolador se incluyen en el capítulo ANEXO I del presente documento.

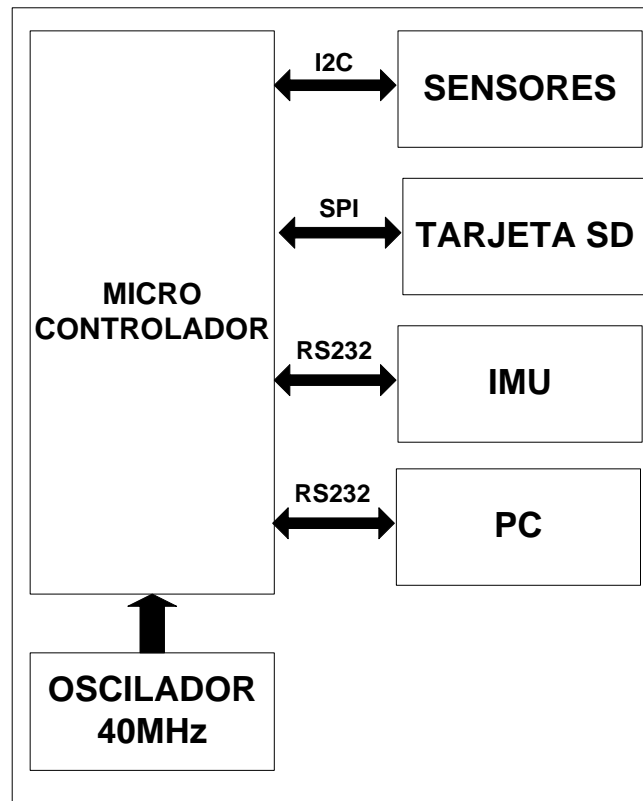


Figura 3.8. Diagrama de bloques de las comunicaciones

BATERÍA RECARGABLE

Conocidas las demandas de tensión y corriente, podemos seleccionar una fuente de alimentación adecuada. El cálculo de la demanda de corriente del circuito se calcula en el capítulo CÁLCULOS del presente documento.

Para alimentar el circuito se selecciona una batería recargable del catálogo de FARNELL (PA-L2.K05.R003). Dicha batería es capaz de entregar 3.7V y tiene una capacidad de 1950mAh. Su reducido tamaño (34.5, 56, 11 mm) y sus características eléctricas se ajustan a las necesidades del sistema. Sus parámetros de carga son 4.2V de tensión máxima de carga y 1300mA de intensidad máxima de carga.



Figura 3.9. Batería ion-litio

CARGADOR DE BATERÍAS DE ION-LITIO

Para recargar la batería es necesario un dispositivo capaz de mantener una corriente estable y constante a medida que la tensión de la batería va aumentando. Para ello se incluirá un integrado LTC4054LES5-4.2 de la marca LINEAR TECHNOLOGY capaz de cargar la batería bajo sus condiciones máximas.



Figura 3.10. Cargador LTC4054LES5-4.2

Este integrado admite una tensión de entrada de entre 4.5V y 6.5V para cargar una batería de ion-Litio hasta una tensión máxima de 4.2V. La corriente de carga viene determinada por una resistencia colocada en su pin PROG, y será de 70mA. El integrado cuenta además con un pin CHRG en el cual podemos colocar un diodo led que nos indicará el final del ciclo de carga.

ADAPTADOR DE TENSIÓN CA/CC

Para alimentar el integrado de carga se selecciona un adaptador CA/CC para así poder usar una toma de corriente común. Encontramos un adaptador en el catálogo de RS (APS1500 Euro) que se ajusta a las condiciones de carga de la batería, 6V y 1500mA de carga. El adaptador cuenta con múltiples clavijas adaptables, seleccionamos el JACK de 3,5mm puesto que contamos con una hembra para esta clavija para PCB.



Figura 3.11. Adaptador de tensión

REGULADOR DE TENSIÓN

La corriente nominal de la batería es de 3,7V, sin embargo es necesario adaptar esta tensión a 3,3v para poder alimentar el microcontrolador, la tarjeta SD y el reloj/termómetro.

Seleccionamos un regulador en el catálogo de RS (MCP1826S-3302E/AB) de la marca MICROCHIP. Dicho regulador cumple los requisitos que necesitamos para adaptar la señal de 3.7V DC a 3.3V DC. Se trata de un regulador LDO (Low Dropout) ya que es necesario dada la pequeña diferencia entre las tensiones de entrada y salida. Su tensión de salida es de 3.3V y es capaz de entregar una corriente máxima de 1A.



Figura 3.12. Regulador de tensión

CONVERTIDOR ELEVADOR

La corriente nominal de la batería es de 3,7v, sin embargo es necesario adaptar esta tensión a 5V para poder alimentar el IMU y la tarjeta de sensores.

Seleccionamos un convertidor elevador en el catálogo de RS (MAX1797 EUA) de la marca MAXIM. Dicho convertidor tiene una tensión de salida de 5v y es capaz de entrega hasta 370mA, suficiente para las necesidades del sistema.



Figura 3.13. Convertidor elevador

ESQUEMA DE ALIMENTACIÓN

A continuación se presenta un esquema general de la etapa de alimentación del sistema electrónico.

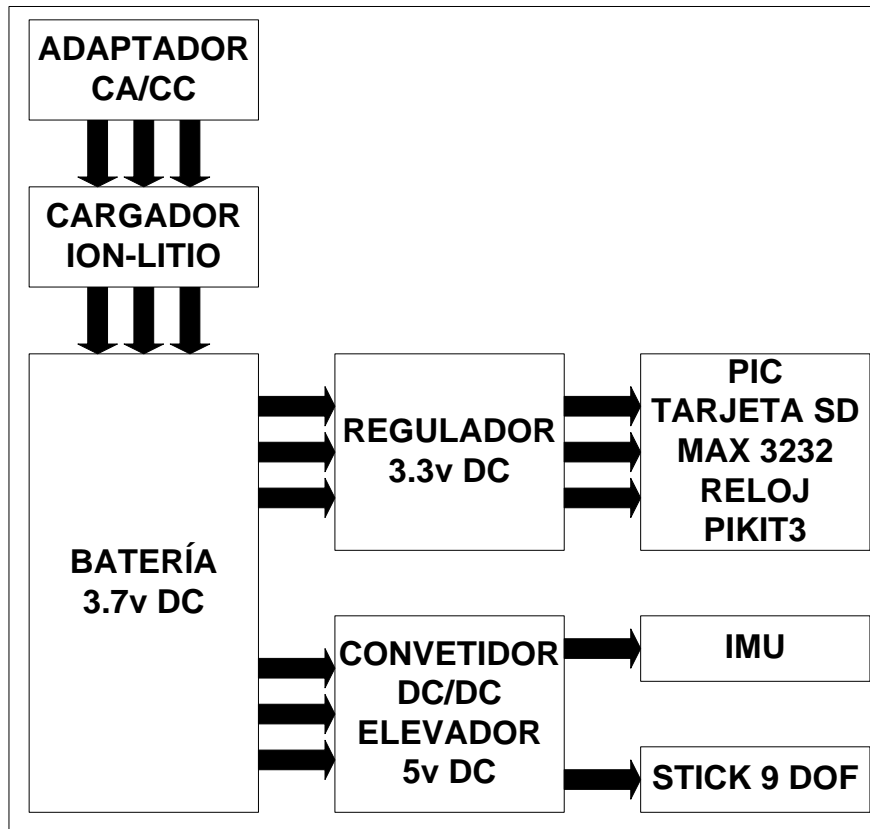
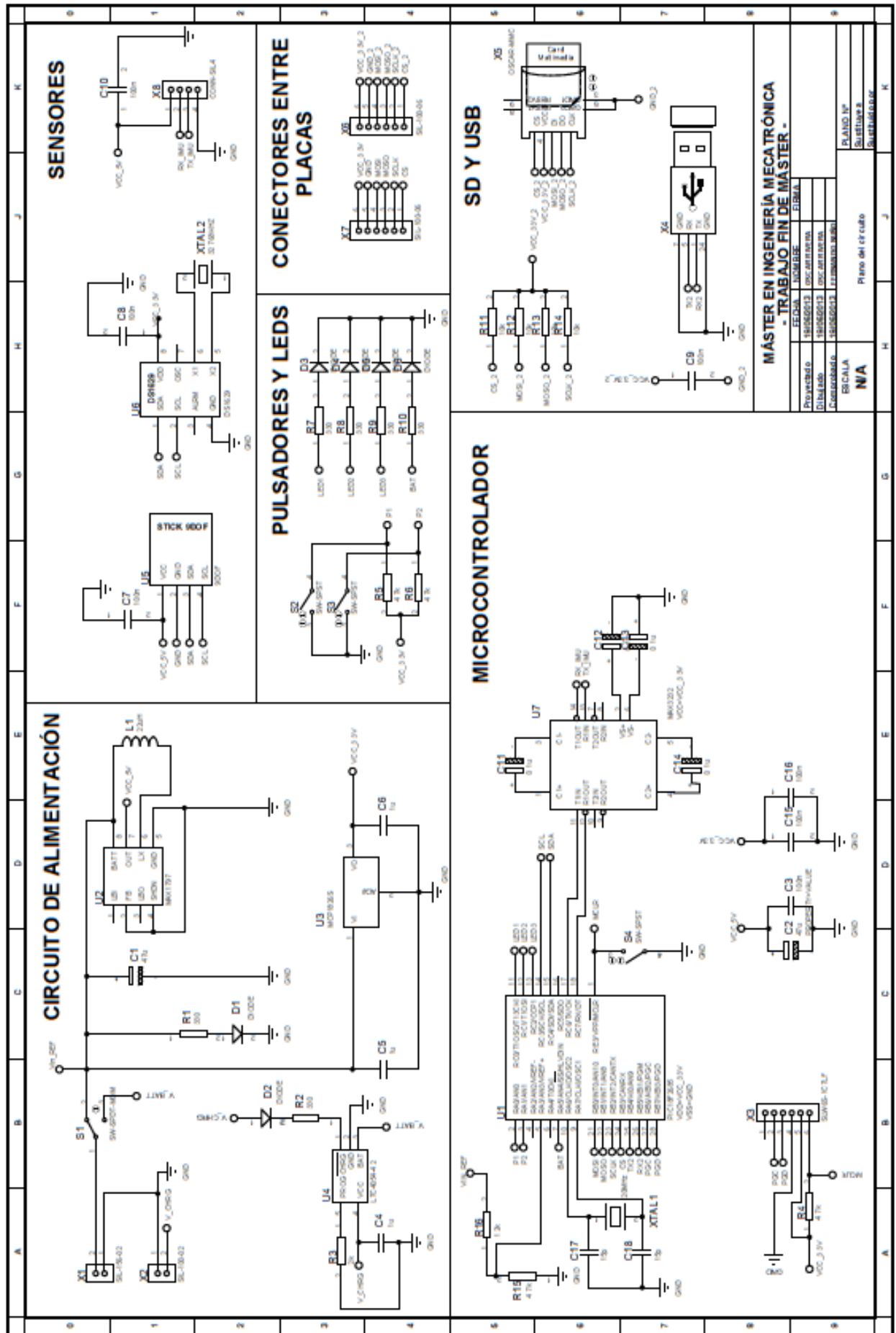


Figura 3.14. Diagrama de bloques de alimentación

3.1.4. CIRCUITO ESQUEMÁTICO

Una vez seleccionados todos los componentes se procede a diseñar el circuito. Para ello se utilizó la aplicación ISIS del programa PROTEUS.



3.1.5. TARJETA DE CIRCUITO IMPRESO

El sistema electrónico se dividirá en dos tarjetas por motivos de espacio. El diseño de dichas PCBs se realizará en base al circuito esquemático del apartado anterior con la aplicación ARES del programa PROTEUS. En el capítulo PLANOS se muestran los planos a escala.

Una vez diseñado se crea una modelo 3D junto con el IMU y la batería en el programa SOLIDWORKS para dimensionar el soporte donde irán ubicadas las tarjetas.

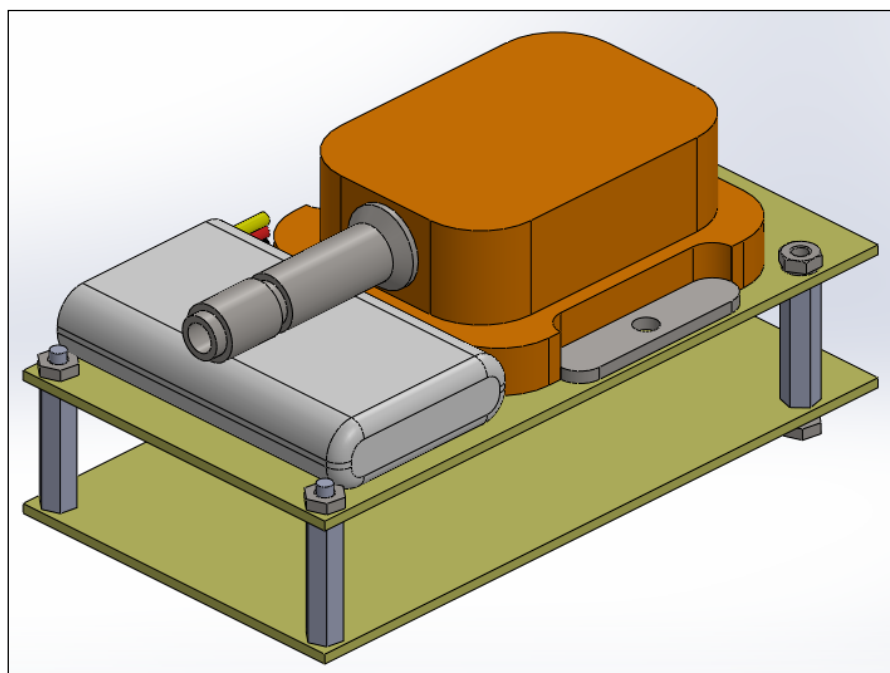


Figura 3.15. Modelo 3D del sistema electrónico

3.2. Diseño mecánico

3.2.1. INTRODUCCIÓN

Para albergar todo el sistema electrónico de adquisición y almacenamiento de datos será necesario diseñar y fabricar un alojamiento que cumpla dicho objetivo y que proteja al prototipo electrónico en las adversas condiciones de trabajo en las que va a operar. Los componentes electrónicos deberán estar sujetos en su interior de manera que actúen como un solo cuerpo, recibiendo así el sistema electrónico las mismas solicitaciones a las que estará sometido el soporte. Dicho soporte deberá contar además con un sistema de sujeción que garantice la fijación del prototipo a la estructura del robot explorador. Atendiendo a las especificaciones del proyecto el sistema de sujeción deberá ser ‘no agresivo’, por lo que será totalmente independiente del robot sin posibilidad de aprovechar ningún elemento de sujeción del propio robot.

Teniendo en cuenta las condiciones de trabajo en las que se mueve el robot explorador será necesario que el soporte sea totalmente estanco, robusto e inmune a agentes corrosivos.

Dado que se trata de un prototipo de pruebas con el propósito de probar el sistema electrónico y que ya se prevé varias actualizaciones y mejoras en él, se considera oportuno realizar un diseño orientado a su fabricación con sistemas de fabricación rápidos (Impresora 3D) aunque el material no sea el más adecuado para esta aplicación (Plástico ABS).

El prototipo se diseñará con el software SOLIDWORKS 2012.

3.2.2. ESTUDIO DEL ROBOT EXPLORADOR

Antes de comenzar con el diseño de la carcasa que aloja el prototipo electrónico, será necesario realizar un estudio de la estructura del robot explorador para localizar el punto óptimo donde se pueda integrar dicho soporte.

Para la elección de este punto hay que tener en cuenta varias cuestiones:

- El robot contiene elementos móviles en su estructura.
- Para su inserción/extracción, el robot cuenta con un sistema de gancho que no se puede obstaculizar.
- Durante la maniobra de inserción/extracción del robot en la zona de trabajo algunos elementos del sistema reducen la zona de posible integración del soporte.
- La parte frontal y posterior del robot resultan inviables ya que el soporte estaría expuesto a los elementos y escombros del trayecto.
- Dependiendo de la zona, el soporte podría recibir más o menos impactos.
- Es necesaria una zona donde se pueda implementar un sistema de sujeción independiente al robot para el soporte.

Teniendo en cuenta todas estas cuestiones se llega a la conclusión de que el único punto que reúne las características mínimas necesarias para la integración del soporte es la ‘espalda’ del robot explorador. Se realiza un CAD de la zona en cuestión lo suficientemente acotada para utilizarlo de referencia a la hora de diseñar el soporte.

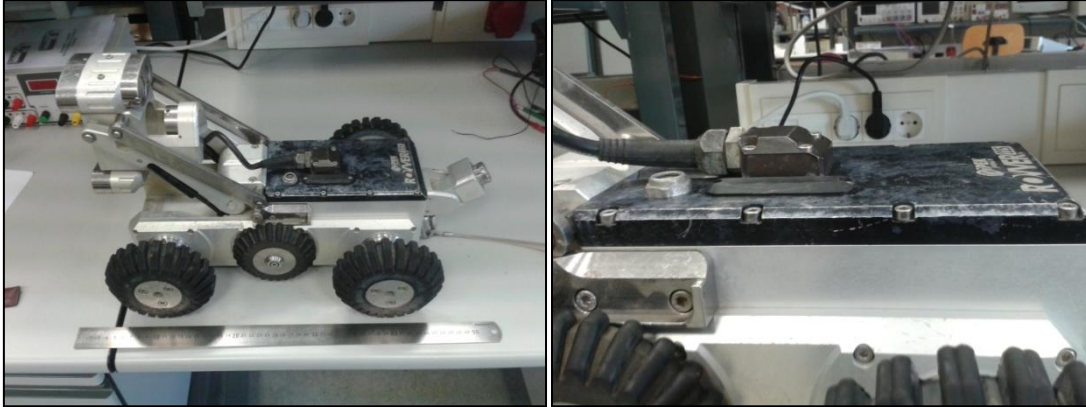


Figura 3.16. Y Figura 3.17. Robot explorador

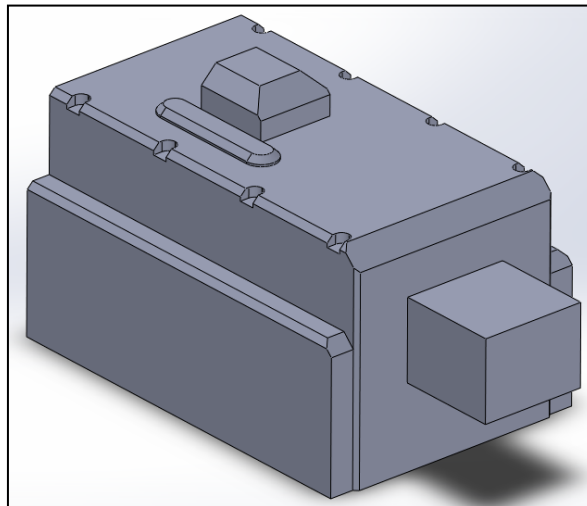


Figura 3.18. Modelo 3D del robot explorador

3.2.3. ALTERNATIVAS DE DISEÑO

El diseño del soporte se puede dividir en dos partes, la caja, donde irá ubicado el sistema electrónico; y el sistema de sujeción, que se encargará de fijar la caja al robot explorador.

CAJA

Se barajan dos posibles soluciones para el diseño de la caja:

1. Partir de una caja estanca comercial y realizar modificaciones sobre la misma para poder integrar el sistema electrónico y el sistema de sujeción.
2. Diseñar y fabricar un prototipo en una impresora 3D de plástico ABS con todas las sujeciones necesarias.

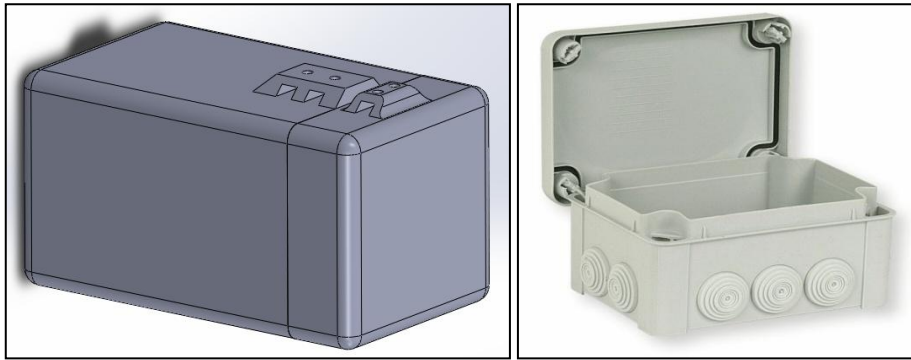


Figura 3.19. Y Figura 3.20. Opciones 1 y 2 de caja

Para poder realizar una elección objetiva del diseño de la caja se establecen una serie de criterios para evaluar las dos posibilidades. Estos criterios son:

- a) Adecuación al objetivo (Peso: 25): Mide el grado de satisfacción con el que cumple los objetivos generales del proyecto.
- b) Estanqueidad (Peso: 25): Mide la capacidad de evitar que agentes externos se introduzcan en el interior de la caja.
- c) Robustez (Peso: 20): Mide la capacidad para absorber impactos.
- d) Manejabilidad (Peso: 10): Facilidad a la hora de abrir y cerrar el soporte.
- e) Precio (Peso: 20).

En base a estos criterios se seleccionará una de las dos opciones. En la siguiente tabla puede observarse la valoración asignada a cada alternativa. (5-Muy bueno→0-Muy malo).

Alternativa/ Criterio	a	b	c	d	e	TOTAL (100)
1	5	3	4	2	5	80
2	5	5	4	4	2	82

Tabla 3.1. Tabla de selección de caja

Criterio a) Ambas alternativas cumplirían adecuadamente los requisitos y objetivos que se necesitan para albergar el sistema electrónico.

Criterio b) Pese a que en principio el 1 es estanco, será necesario realizar modificaciones en él para incluir el sistema de fijación a la caja y el sistema de fijación interno del sistema electrónico, por lo que su estanqueidad se verá mermada considerablemente. El 2 en cambio está diseñado para que la integración del sistema de fijación no suponga realizar orificios pasantes en la caja, no comprometiendo así la estanqueidad del conjunto.

Criterio c) Ambas alternativas están formadas por el mismo material ABS, por lo que ofrecen la misma robustez ante impactos.

Criterio d) Para la opción 1 es necesario usar una herramienta para cerrar la caja, mientras que la opción 2 cuenta con un sistema de cierre que permite anclar la tapa con las manos.

Criterio e) En precio la opción 1 es claramente mejor que la opción 2, sin embargo como se trata de un prototipo y no se prevé su fabricación en cadena, se le ha asignado poco peso en la tabla.

Finalmente se selecciona la opción 2 dado que ha obtenido una mayor puntuación en la tabla de criterios.

SISTEMA DE FIJACIÓN

Se barajan dos posibles soluciones para el diseño del sistema de sujeción al robot:

1. Sistema de palanca.
2. Sistema de pinza.

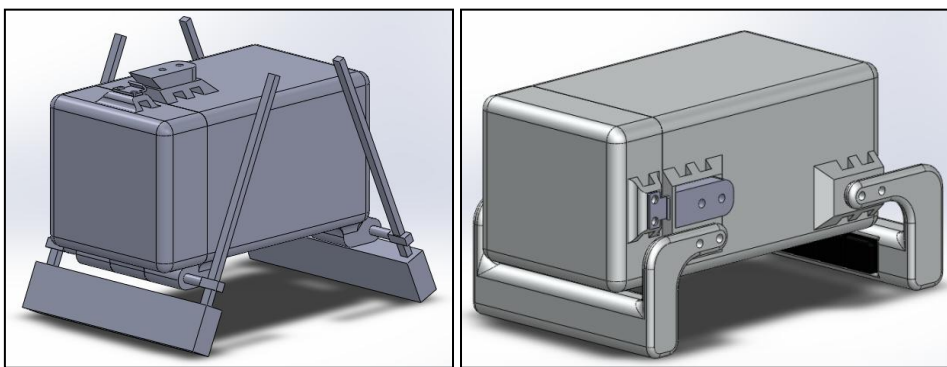


Figura 3.21. Y Figura 3.22. Opciones 1 y 2 de sistema de sujeción

Para poder realizar una elección objetiva del diseño del sistema de sujeción se establecen una serie de criterios para evaluar las dos posibilidades. Estos criterios son:

- a) Adecuación al objetivo (Peso: 50): Mide el grado de satisfacción con el que cumple los objetivos generales del proyecto.
- b) Robustez (Peso: 25): Mide la capacidad para absorber impactos.
- c) Manejabilidad y sencillez (Peso: 25): Facilidad a la hora de fijar el soporte.

En base a estos criterios se seleccionará una de las dos opciones. En la siguiente tabla puede observarse la valoración asignada a cada alternativa. (5-Muy bueno→0-Muy malo).

Alternativa/ Criterio	a	b	c	TOTAL (100)
1	4	3	3	70
2	5	4	4	90

Tabla 3.2. Tabla de selección de sistema de sujeción

Criterio a) El mecanismo de la opción 1 no permite una sujeción tan firme como el de la opción 2.

Criterio b) El mecanismo 2 es más sencillo, con dos piezas unidas a la base con tornillos, lo que la hace menos vulnerable ante un impacto. La opción 1 en cambio necesita de un sistema para empujar las palancas que, en caso de impacto, podría mermar su funcionamiento.

Criterio c) El mecanismo 2 es más fácil de fabricar y de utilizar, basta con apretar tres tornillos; mientras que la alternativa 2 resulta en un diseño más complejo y costoso de fabricar.

Finalmente se selecciona la opción 2 dado que ha obtenido una mayor puntuación en la tabla de criterios.

3.2.4. DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA

Una vez seleccionado el tipo de caja y el sistema de sujeción, se procede a describir las principales características del conjunto. Como se detalla en el apartado anterior se opta por diseñar una caja que se fabricará en una impresora 3D de plástico ABS y se utilizará un mecanismo de pinza para sujetarla al robot.

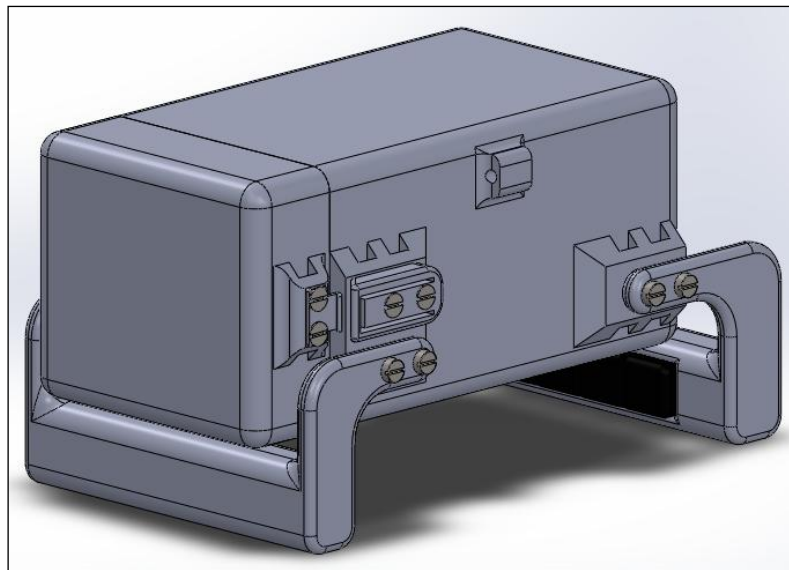


Figura 3.23. Modelo final del prototipo

El soporte estará constituido por 4 piezas: caja, tapa, pieza de sujeción y pieza de sujeción con sistema de presión.

La caja albergará el sistema electrónico. Sus dimensiones son 60x70x111 mm, con 3mm de espesor, suficiente para introducir la tarjeta electrónica con una holgura de 2mm a cada lado. En sus laterales cuenta con unos salientes para atornillar las piezas de sujeción sin necesidad de perforar la estructura principal de la caja, evitando así problemas de estanquidad. En su interior cuenta con un sistema de sujeción para el sistema electrónico. Al fondo una guía y al frente dos apoyos para atornillar la placa a la caja. Todos los agujeros de la caja son para tornillos M3.

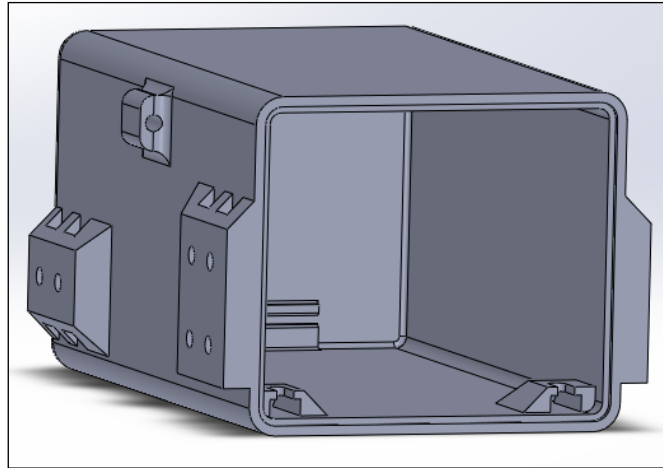


Figura 3.24. Modelo final de caja

La tapa de la caja se fabricará de la misma manera que la caja. Sus dimensiones son 60x70x25 mm, con 3mm de espesor. A sus lados cuenta con dos apoyos para el cierre de palanca que mantendrá unidos la caja y la tapa. Todos los agujeros de la tapa son para tornillos M3.

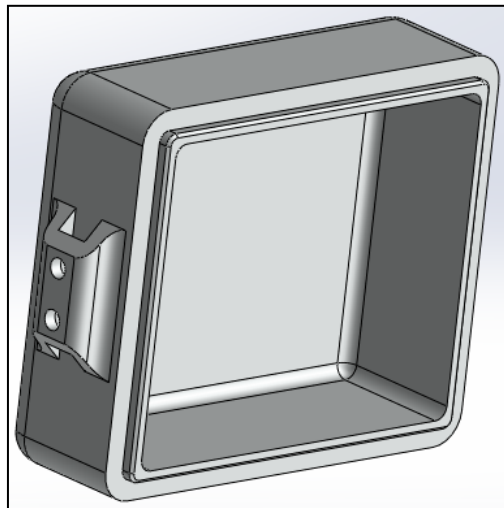


Figura 3.25. Modelo final de tapa

Para asegurar la estanquidad del soporte se incluirá una junta tórica de 1,6mm de diámetro entre la tapa y la caja.

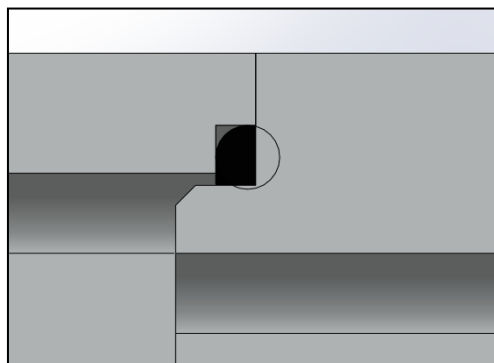


Figura 3.26. Vista de perfil de unión entre caja y tapa

Se diseña la caja con dos salientes, uno a cada lado, con un orificio pasante de 3mm de diámetro como elemento de fijación auxiliar ante un posible fallo del sistema de sujeción al robot.

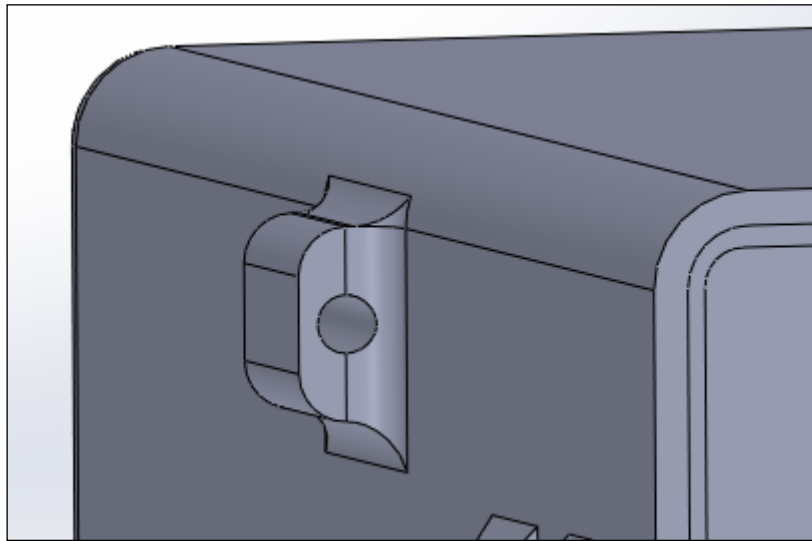


Figura 3.27. Orificios para sistema de fijación auxiliar

Las piezas de sujeción se fijan a la base a través de cuatro tornillos M3 cada una. En su parte inferior-interior cuentan con unas zapatas que serán las encargadas de hacer contacto con el robot explorador. La pieza de sujeción con sistema de presión cuenta con tres orificios a través de los cuales se introducen tres tornillos M3 que presionan la zapata contra el robot, fijando así todo el sistema.

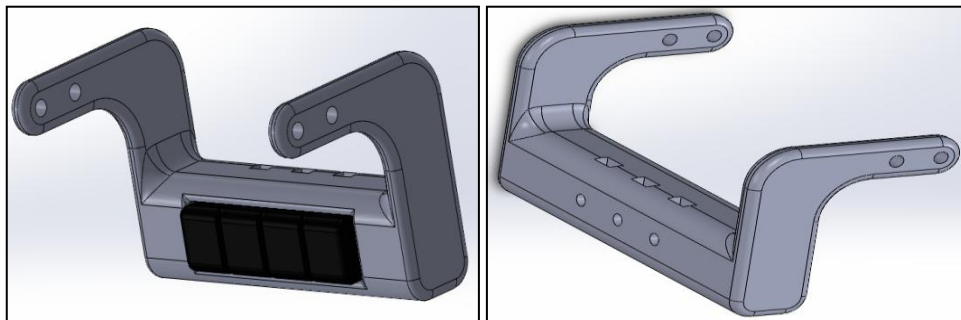


Figura 3.28. Y Figura 3.29. Piezas del sistema de sujeción al robot

4. FABRICACIÓN DEL PROTOTIPO

En este capítulo se detallan los procesos de fabricación empleados para construir el prototipo, tanto el sistema electrónico como el soporte.

4.1. Fabricación del sistema electrónico

La fabricación del sistema electrónico engloba la fabricación, soldadura de componentes y ensamble de las dos placas de circuito impreso.

Ambas placas se han realizado a doble cara, por motivos de diseño y número de componentes; con un grosor de pista de 40 centésimas de pulgada para las pistas de potencia y 30 centésimas de pulgada para las de señal.

Por seguridad se dejó un margen de 10 centésimas de pulgada entre pista-pista y pista-pad.

Será necesario imprimir el diseño creado en ARES para realizar el fotolito, el cual se incluye en el capítulo PLANOS a escala 1/1. Con el fotolito de las dos caras de la tarjeta, se procede a insolar, revelar, y atacar con ácido la tarjeta para obtener el circuito impreso deseado.

Antes de taladrar la placa y soldar los componentes, es necesario limpiar la tarjeta con alcohol para deshacer la capa de barniz que todavía protege la tarjeta.

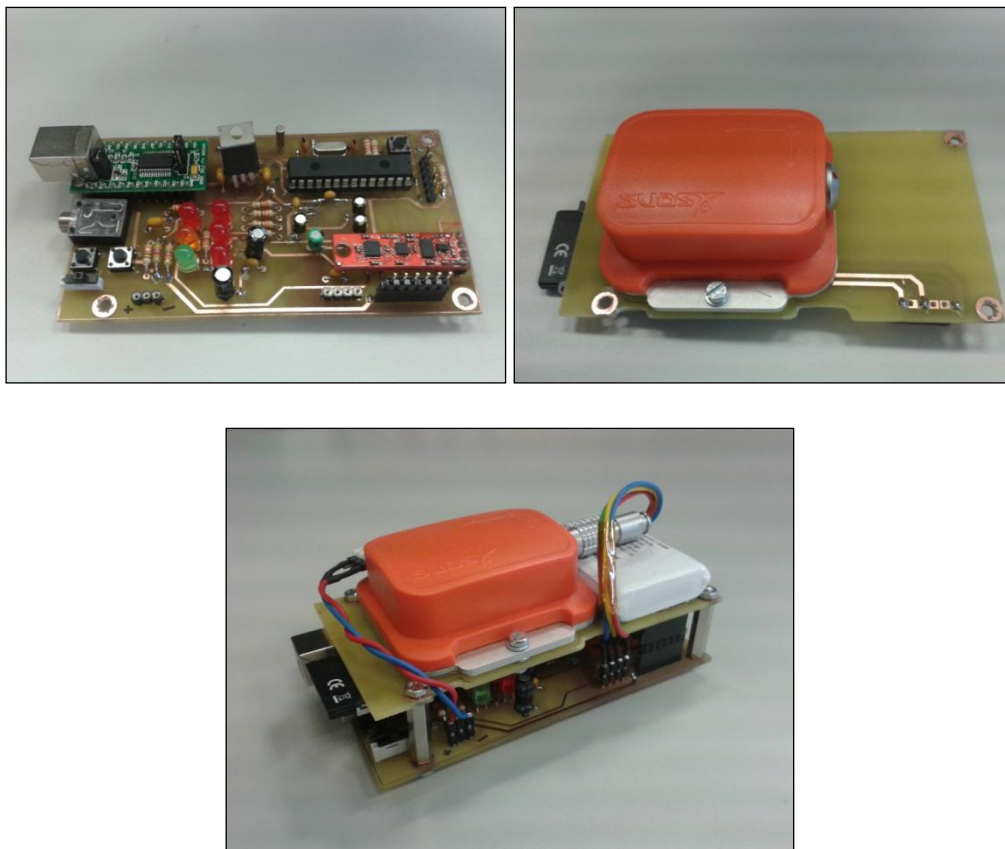


Figura 4.1. Figura 4.2. Figura 4.3. Sistema electrónico

4.2. Fabricación del soporte

La fabricación del soporte de fijación se realizará con la impresora de prototipado rápido HP Designjet 3D. Para poder imprimir piezas en esta impresora es necesario guardar los archivos diseñados en SOLIDWORKS con formato.stl.

Esta impresora necesita de dos tipos de material para fabricar las piezas, el principal (ABS), que forma el cuerpo principal de cada pieza, y el de soporte, que sirve de apoyo para construir partes de la pieza en voladizo.

El proceso de fabricación de la caja, la tapa, y las dos piezas de fijación duró 20 horas, con un consumo de 187 cm³ de material principal y 50 cm³ de soporte.

Una vez finalizado el proceso se procede a eliminar el material de soporte.

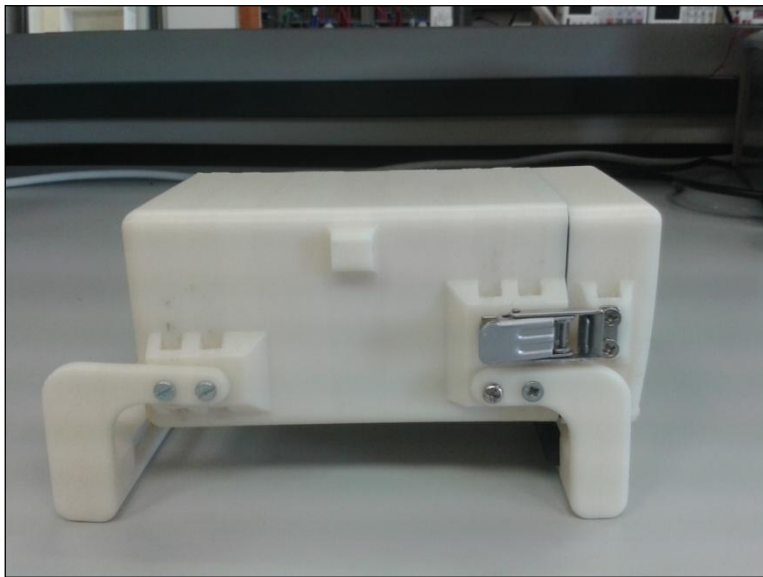


Figura 4.4. Soporte del prototipo

5. CÁLCULOS

En este capítulo se detallan los cálculos y operaciones necesarias para el diseño del prototipo.

5.1. Conversión de datos

Los datos procedentes de los sensores inerciales individuales son recogidos en bits, es necesario adaptarlos a las unidades correspondientes.

5.1.1. ACELERÓMETRO

El acelerómetro está configurado con una resolución de 10 bits y un rango de medida de $\pm 4G$.

$$Factor\ de\ escala = \frac{Rango\ de\ medida}{Resoluci\ on} = \frac{8G}{2^{10}\ bits} = 7,8mG/bit \quad (5.1)$$

$$Dato(m/s^2) = Dato(bits) * Factor\ de\ escala * \frac{9,8\ m/s^2}{1000} \quad (5.2)$$

5.1.2. GIRÓSCOPO

El giróscopo está configurado con una resolución de 16 bits y un rango de medida de $\pm 2000^\circ/s$.

$$Factor\ de\ escala = \frac{Rango\ de\ medida}{Resoluci\ on} = \frac{4000^\circ/s}{2^{16}\ bit} = 0,061 \frac{^\circ/s}{bit} \quad (5.3)$$

$$Dato(rad/s) = Dato(bits) * Factor\ de\ escala * \frac{\pi\ rad}{180^\circ} \quad (5.4)$$

5.1.3. MAGNETÓMETRO

El magnetómetro está configurado con una resolución de 12 bits y un rango de medida de $\pm 4,7Ga$.

$$Factor\ de\ escala = \frac{Rango\ de\ medida}{Resoluci\ on} = \frac{9,4\ Ga}{2^{12}\ bit} = 0,0022 \frac{Ga}{bit} \quad (5.5)$$

$$Dato(Ga) = Dato(bits) * Factor\ de\ escala \quad (5.6)$$

5.2. TIMERO

Se programará el TIMER0 del microcontrolador para realizar la rutina de muestreo a una frecuencia de 3Hz. Hay que tener en cuenta que el microcontrolador trabaja con un reloj de 40MHz y que el TIMER tiene una resolución de 16 bits.

Se calculará la precarga necesaria para que el TIMER se desborde cada 0.333s.

$$Precarga = 2^{16} - \frac{T_{TIMER0} * f_{osc}}{4 * Prescaler} = 65536 - \frac{0.33 * 40 * 10^6}{4 * 256} = 52645 \text{ bit} \quad (5.7)$$

5.3. Tensión de referencia del módulo comparador

Para monitorizar la tensión de la batería es necesario compararla con una referencia que se calculará a continuación. Además será necesario adaptar la tensión de la batería a un rango de tensión admisible por el microcontrolador con un divisor de tensión.

5.3.1. TENSIÓN DE REFERENCIA

El microcontrolador es capaz de generar una tensión de referencia interna para compararla con una tensión externa, dicha referencia depende de la tensión de alimentación (3,3v) y no puede ser superior a ésta.

Se configurará el módulo que genera la tensión de referencia para que su valor sea el máximo posible. Para ello CVR=15 (Máximo valor).

$$V_{ref} = \left(\frac{CVR}{24}\right) * V_{dd} = \frac{15}{24} * 3,3 = 2,06V \quad (5.8)$$

5.3.2. DIVISOR DE TENSIÓN

La tensión máxima de la batería es de 4,2V y el prototipo se apaga cuando ésta tensión baja de los 3,3V. Se pretende configurar el sistema para que la señal de aviso de carga aparezca cuando la batería se encuentra por debajo de los 3,5V (22.25% de la carga máxima). Se diseñará un divisor de tensión que adapte la señal de tal manera que cuando la batería tenga 3,5v, la tensión que reciba el microcontrolador sea de 2,06v.

$$V_{RA3} = \frac{R2}{R1+R2} * V_{BAT} = \frac{4,7K}{4,7K+3,3K} * 3,5 = 2.06V \quad (5.9)$$

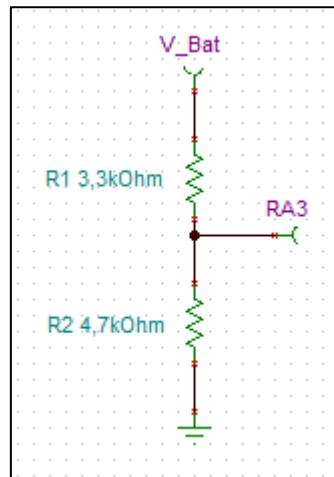


Figura 5.1. Divisor de tensión

5.4. Consumo de corriente

Para seleccionar una fuente de energía es necesario conocer primero las demandas de corriente que exige el sistema electrónico. Se realizará una estimación a partir de los consumos teóricos de intensidad facilitados por el fabricante.

COMPONENTE	Intensidad
IMU	200mA
Acelerómetro ADXL 345	140uA
Magnetómetro HMC5883L	100uA
Giróscopo ITG-3200	6,5mA
Reloj DS1629	1mA
PIC 18F27J13	20mA*
Tarjeta SD	10uA
MAX 3232	1mA
TOTAL	228,75mA

Tabla 5.1. Tabla de consumos de corriente

*Consumo estimado

6. COSTES

En este capítulo se detallan los costes de los materiales necesarios para realizar el prototipo. No se incluye el IMU Xsens MTi-28A53G25 en la suma total de los costes de los componentes electrónicos porque se trata de un componente facilitado por el Área de Ingeniería de Sistemas y Automática que no supuso un coste directo en el presente proyecto. La única finalidad de este dispositivo es comparar sus datos con los de un IMU de bajo coste desarrollado con componentes discretos.

MÁSTER DE INGENIERÍA MECATRÓNICA		Pag. 1 COMPONENTES ELECTRÓNICOS			
Nº ORDEN	CONCEPTOS	Nº UNIDADES	FABRICANTE	PRECIO UNITARIO MATERIAL	TOTAL
1	Placa PCB 2 caras 300x200x0,8mm	2	CIF	8,78 €	17,5 €
2	Separadores M3 25mm	4	RS	0,19 €	0,76 €
3	Batería recargable PANASONIC - PA-L2	1	PANASONIC	34,44 €	34,4 €
4	Adaptador de tensión 230v-3v	1	ANSMANN	18,52 €	18,5 €
5	Tarjeta IMU 9 DOF	1	SPARKFUN	21,43 €	21,4 €
6	Sensor T ^a /reloj	1	MAXIM	5,08 €	5,08 €
7	Conector IMU	1	LEMO	41,44 €	41,44 €
8	Microcontrolador	1	MICROCHIP	3,61 €	3,61 €
9	Adaptador RS232	1	MAXIM	4,21 €	4,21 €
10	Regulador de tensión 3,3v 1A LDO	1	MICROCHIP	1,09 €	1,09 €
11	Convertidor elevador 5v 350mA	1	MAXIM	1,67 €	1,67 €
12	Conector Jack hembra 3,5mm STER	1	Decelect Forgos	1,25 €	1,25 €

MÁSTER DE INGENIERÍA MECATRÓNICA		Pag. 2 COMPONENTES ELECTRÓNICOS			
Nº ORDEN	CONCEPTOS	Nº UNIDADES	FABRICANTE	PRECIO UNITARIO MATERIAL	TOTAL
13	Adaptador microSD-SD	1	AMPHENOL	1,33 €	1,33 €
14	Socket SD montaje superficial	1	Hirose	3,52 €	3,52 €
15	Tarjeta de memoria microSD 2GB	1	Transcend	6,52 €	6,52 €
16	Interruptor de encendido	1	EOZ	1,79 €	1,79 €
17	Cargador ion-lítio 4,2v	1	LINEAR	3,79 €	3,79 €
18	Cristal oscilador 32.768Hz	1	IQD	0,54 €	0,54 €
19	Cristal oscilador 20MHz	1	Euroquartz	0,63 €	0,63 €
20	Diodo LED	6	Kingbright	0,16 €	0,96 €
21	Pulsador	3	TE Connectivity	0,107 €	0,32 €
22	Resistencias 1/2W 5%	16	RS	0,007 €	0,11 €
23	Condensador electrolítico 47uF 25v	2	Panasonic	2,93 €	5,86 €
24	Condensador cerámico 1uF/50v	4	Kemet	0,90 €	3,60 €
25	Condensador cerámico 100nF/50v	7	AVX	0,14 €	0,98 €

MÁSTER DE INGENIERÍA MECATRÓNICA		Pag. 3 COMPONENTES ELECTRÓNICOS			
Nº ORDEN	CONCEPTOS	Nº UNIDADES	FABRICANTE	PRECIO UNITARIO MATERIAL	TOTAL
26	Condensador electrolítico 1uF/50v	4	Panasonic	0,22 €	0,88 €
27	Condensador cerámico 15pF/50v	2	Vishay	0,276 €	0,55 €
28	Bobina 22uH	1	Colicraft	2,02 €	2,02 €
29	Tira de pines	5	Preci-Dip	3,94 €	19,7 €
30	IMU	1	XSENS	1.020,00 €	1.020 €*
SUBTOTAL					204,01 €

*Componente no incluido en la suma total

MÁSTER DE INGENIERÍA MECATRÓNICA		Pag. 1 COMPONENTES MECÁNICOS			
Nº ORDEN	CONCEPTOS	Nº UNIDADES	FABRICANTE	PRECIO UNITARIO MATERIAL	TOTAL
1	Cordón nitrilo p/junta tórica, diámetro 1.6mm	1	RS	11,40 €	11,40 €
2	Cierre de palanca superior	2	RS	1,15 €	2,3 €
3	Cierre de palanca inferior	2	RS	8,96 €	17,92 €
4	Kit 2 Zapatás	1	btwin	5,95 €	5,95 €
5	Material ABS principal	186.8 cm3	HP	112.08 €	112.08 €
6	Material ABS soporte	48.94 cm3	HP	29.36 €	29.36 €
7	Estuche de tornillos y tuercas M3	1	CEBEK	12,45 €	12,45 €
SUBTOTAL					191,46 €

Subtotal componentes electrónicos	204,01 €
Subtotal componentes mecánicos	191,46 €
TOTAL	395,47 €

7. ANEXO I: ESPECIFICACIONES TÉCNICAS

Se detallan las características, conexiones, y configuraciones de los componentes que lo requieren.

Las hojas de características de todos los componentes se pueden encontrar en el CD adjunto.

7.1. Acelerómetro ADXL345

El acelerómetro que incorpora la tarjeta es el modelo ADXL345 de la marca ANALOG DEVICES. Se trata de un acelerómetro digital con 3 grados de libertad con conexión para comunicación I2C y de montaje superficial.

Este sensor permitirá almacenar los datos de aceleración a los que estará sometido el robot explorador a lo largo de un ciclo de trabajo. Dichos datos corresponden a la aceleración a la que están sometidos cada uno de los tres ejes x, y, z del sensor y cuyo sentido y orientación se detallan al final de este apartado.

Gracias a estos datos se puede obtener, mediante el algoritmo correspondiente, la posición del robot explorador en cada instante respecto al punto inicial, pudiendo calcular así la trayectoria descrita por el robot.

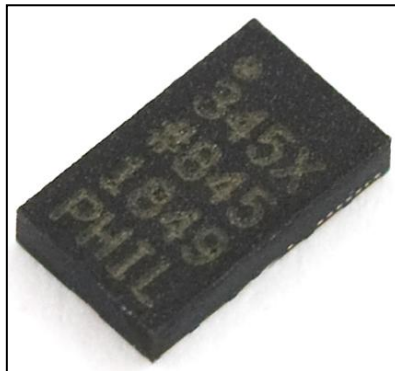


Figura 7.1. Acelerómetro ADXL345

El rango de medida del acelerómetro y su resolución se pueden modificar accediendo al correspondiente registro. De esta manera se puede elegir entre 4 rangos de medida (± 2 , ± 4 , ± 8 , $\pm 16g$) y 4 rangos de resolución (10, 11, 12 y 13 bits) dependiendo del rango de medida.

La frecuencia de muestreo también se puede configurar, pudiendo variar su valor entre los valores 6,25 y 3200 Hz.

Para poder escribir y leer los registros de este sensor habrá que usar la dirección 0xA6 para escribir y 0xA7 para leer en el BUS I2C. Los registros que usaremos son: BW_RATE, DATA_FORMAT Y POWER_CTL.

BW_RATE se encuentra en la posición 0x2C de la memoria del sensor. En este registro podremos configurar la frecuencia de muestreo. Con el valor 1010 introducido en los cuatro bits menos significativos del registro configuramos el sensor con una frecuencia de muestreo de 100Hz.

Register 0x2C—BW_RATE (Read/Write)							
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	LOW_POWER	Rate			

Tabla 7.1. Tabla del registro BW_RATE del acelerómetro

El siguiente registro a configurar es el DATA_FORMAT, situado en la posición 0x31, en el cual podemos configurar el rango de medida, la resolución y el formato de los datos, entre otras cosas. Si cargamos este registro 0x03 se obtiene un rango de medida de $\pm 4g$ con 10 bits de resolución.

Register 0x31—DATA_FORMAT (Read/Write)							
D7	D6	D5	D4	D3	D2	D1	D0
SELF_TEST	SPI	INT_INVERT	0	FULL_RES	Justify	Range	

Tabla 7.2. Tabla del registro DATA_FORMAT del acelerómetro

Por último habrá que configurar el registro POWER_CTL, situado en la posición 0x2D, para comenzar la medida de datos. Poniendo a 1 el bit Measure el sensor comenzará a tomar datos a la frecuencia configurada.

Register 0x2D—POWER_CTL (Read/Write)							
D7	D6	D5	D4	D3	D2	D1	D0
0	0	Link	AUTO_SLEEP	Measure	Sleep	Wakeup	

Tabla 7.3. Tabla del registro POWER_CTL del acelerómetro

Los datos de aceleración se encuentran en 6 registros de la memoria; DATA0, DATA1, DATAY0, DATAY1, DATAZ0 Y DATAZ1 entre las posiciones 0x32 y 0x37 de la memoria del sensor. Cada dato es de 16 bits en complemento a 2, siendo DATAx0 la parte baja del dato, y DATAx1 la parte alta del dato. El sentido positivo de cada eje se muestra en la siguiente imagen.

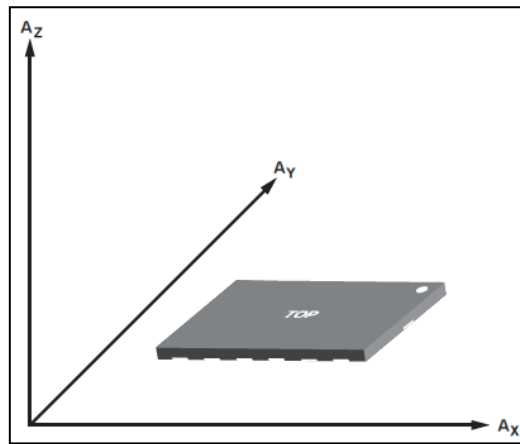


Figura 7.2. Esquema del sentido y dirección de los ejes del acelerómetro

7.2. Giróscopo ITG-3200

El giróscopo que incorpora la tarjeta es el modelo ITG-3200 de la marca INVENSENSE. Se trata de un giróscopo digital con 3 grados de libertad con conexión para comunicación I2C y de montaje superficial.

Este sensor permitirá almacenar los datos de velocidad angular que desarrolla el robot explorador en cada instante. Dichos datos se obtienen respecto a tres ejes x, y, z del sensor y cuyo sentido y orientación se detallan al final de este apartado.

Gracias a estos datos se puede obtener los ángulos de navegación y saber así en qué sentido y dirección se encuentra el robot explorador en cada instante.

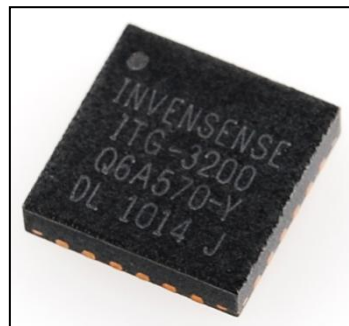


Figura 7.3. Giróscopo ITG-3200

El rango de medida del sensor es de ± 2000 °/s con una resolución de 16 bits. La frecuencia de muestreo puede variar desde 3,9 Hz hasta 8kHz.

Para poder escribir y leer los registros de este sensor habrá que usar la dirección 0xD0 para escribir y 0xD1 para leer en el BUS I2C. Los registros que usaremos son: DLPG y SMPLRT.

El registro DLPG se encuentra en la posición 0x16 y en él podemos configurar el rango de medida, la frecuencia de corte del filtro digital pasa-bajos incluido en el sensor y la frecuencia de muestreo interna del sensor. Configuramos este registro con el valor 0x19 para tener un rango de medida de ± 2000 °/s, una frecuencia de corte de 188Hz y una frecuencia de muestreo interna de 1kHz.

Register 22 – DLPF, Full Scale										
Type: Read/Write										
Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value
16	22	-			FS SEL		DLPF CFG			00h

Tabla 7.4. Tabla del registro DPLG del giróscopo

El registro SMPLRT se encuentra en la posición 0x15 y en él podemos configurar la frecuencia de muestreo de salida del sensor. Configuramos este registro con el valor 0xFF para obtener un divisor de frecuencia de 255. De esta manera la frecuencia de muestreo de salida será de 3,9 Hz.

Register 21 – Sample Rate Divider											
Type: Read/Write											
Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default Value	
15	21	SMPLRT DIV									00h

Tabla 7.5. Tabla del registro SMPLRT del giróscopo

Los datos de orientación se encuentran en 6 registros de la memoria; GYRO_XOUT_H, GYRO_XOUT_L, GYRO_YOUT_H, GYRO_YOUT_L, GYRO_ZOUT_H Y GYRO_ZOUT_L entre las posiciones 0x1D y 0x22 de la memoria del sensor. Cada dato es de 16 bits en complemento a 2, siendo GYRO_xOUT_H la parte alta del dato, y GYRO_xOUT_L la parte baja del dato. El sentido positivo de cada eje se muestra en la siguiente imagen.

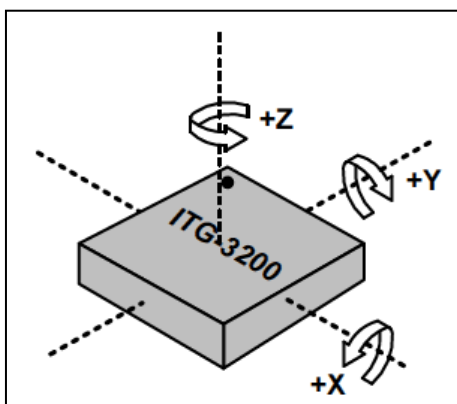


Figura 7.4. Esquema del sentido y dirección de los ejes del giróscopo

7.3. Magnetómetro HMC-5883L

El magnetómetro que incorpora la tarjeta es el modelo HMC-5883L de la marca HONEYWELL. Se trata de un magnetómetro digital con 3 grados de libertad con conexión para comunicación I2C y de montaje superficial.

Este sensor permitirá almacenar los datos de campo magnético a los que estará sometido el robot explorador en cada instante. Dichos datos corresponden al campo magnético al que están sometidos cada uno de los tres ejes x, y, z del sensor y cuyo sentido y orientación se detallan al final de este apartado.

Gracias a estos datos se puede obtener los ángulos de navegación y saber así en qué sentido y dirección se encuentra el robot explorador en cada instante.

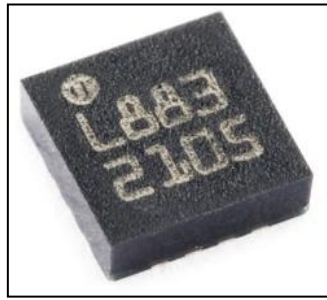


Figura 7.5. Magnetómetro HMC-5883L

El rango de medida del magnetómetro se puede modificar accediendo al correspondiente registro. De esta manera se puede elegir entre 8 rangos de medida (± 0.88 , ± 1.3 , ± 1.9 , ± 2.5 , ± 4.0 , ± 4.7 , ± 5.6 , ± 8.1 Ga) con una resolución de 12bits.

La frecuencia de muestreo también se puede configurar, pudiendo variar su valor entre los valores 0.75 y 75 Hz.

Para poder escribir y leer los registros de este sensor habrá que usar la dirección 0x3C para escribir y 0x3D para leer en el BUS I2C. Los registros que usaremos son: CONF_A, CONF_B y MODE.

CONF_A se encuentra en la posición 0x00 de la memoria del sensor. En este registro podremos configurar la frecuencia de muestreo y el número de muestras que tomará para hacer la media por cada dato de salida. Cargamos este registro con el valor 0x70 para obtener 8 muestras por dato de salida y una frecuencia de muestreo de 15Hz.

CRA7	CRA6	CRA5	CRA4	CRA3	CRA2	CRA1	CRA0
(0)	MA1(0)	MA0(0)	DO2 (1)	DO1 (0)	DO0 (0)	MS1 (0)	MS0 (0)

Table 3: Configuration Register A

Tabla 7.6. Tabla del registro CONF_A del magnetómetro

CONF_B se encuentra en la posición 0x01 de la memoria del sensor. En este registro podremos configurar el rango de medida del sensor y a su vez la ganancia del mismo. Cargamos este registro con el valor 0xA0 para un rango de medida de ± 4.7 Ga y una ganancia de 390LSB/Ga.

CRB7	CRB6	CRB5	CRB4	CRB3	CRB2	CRB1	CRB0
GN2 (0)	GN1 (0)	GN0 (1)	(0)	(0)	(0)	(0)	(0)

Table 7: Configuration B Register

Tabla 7.7. Tabla del registro CONF_B del magnetómetro

Por último, accedemos al registro MODE situado en la posición 0x02 para configurar el modo de funcionamiento del sensor. Cargaremos dicho registro con el valor 0x00 para configurarlo en modo medida continua.

MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
(0)	(0)	(0)	(0)	(0)	(0)	MD1 (0)	MD0 (1)

Table 10: Mode Register

Tabla 7.8. Tabla del registro MODE del magnetómetro

Los datos de campo magnético se encuentran en 6 registros de la memoria; DXRA, DXRB, DYRA, DYRB, DZRA, y DZRB, entre las posiciones 0x03 y 0x08 de la memoria del sensor. Cada dato es de 16 bits en complemento a 2, siendo; DXRA la parte alta del dato, y; DXRB la parte baja del dato. El sentido positivo de cada eje se muestra en la siguiente imagen.

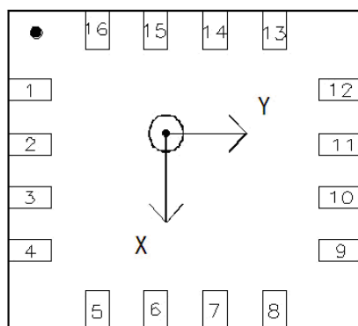


Figura 7.6. Esquema del sentido y dirección de los ejes del magnetómetro

7.4. Reloj termómetro DS1629

El sensor DS1629 incorpora un reloj y un termómetro, con el primero podremos situar cada muestra en el tiempo y gracias al segundo obtendremos la temperatura de la placa.

Incorporando un oscilador externo de 32.768 Hz el reloj será capaz de contar segundos, minutos, horas, semanas y años. Para este caso con contar hasta horas bastará dado el ciclo de trabajo del robot explorador.

Los datos serán enviados por I2C, sin embargo, no incorporaremos resistencias de PULL-UP en la PCB puesto que la tarjeta de SPARKFUN, donde están incorporados los otros tres sensores, ya cuenta con dichas resistencias. La dirección de escritura de este sensor será 0x9E mientras que la de lectura será 0x9F.

Para configurar el reloj y el termómetro será necesario acceder al registro ACCESS_CONFIGURATION en la posición 0xAC. Se trata de un registro de 2 bytes en el que sólo el byte más significativo es de escritura. Cargaremos este registro con 0x00 para que el sensor realice medidas de temperatura constantemente, desactivar las alarmas y utilizar la frecuencia del oscilador sin divisores (32.768 Hz).

Configuration/Status Register Figure 5								
OS1	OS0	A1	A0	0	CNV	POL	1SH	MSB
							LSb	
CAF	TAF	CAL	TAL	0	0	0	0	LSB

Tabla 7.9. Tabla del registro ACCESS_CONFIGB del reloj

El reloj empieza a contar desde 0 desde el momento en el que el sensor recibe alimentación. Por tanto necesitaremos resetearlo en el momento en el que consideremos que queremos que empiece el almacenamiento de datos. Para ello se accederá al registro CLOCK_REGISTER_FORMAT situado en la posición 0xC0. Éste está a su vez dividido en 7 registros, de los cuales 0x00, 0x01y 0x02 deben ser cargados con 0x00 para que el contador empiece a contar desde 0 horas, 0 minutos, 0 segundos. Es posible también iniciar la cuenta en una hora determinada, basta con cargar dicho registro con los valores deseados teniendo en cuenta la estructura en la que se muestran los datos.

BYTE ADDRESS	BIT 7 MSb	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 Lsb	BYTE RANGE
00h	CH	10 SECONDS			SECONDS				00-59
01h	0	10 MINUTES			MINUTES				00-59
02h	0	12 MODE	AM/PM	10 HOURS	HOURS				01-12 00-23
		24 MODE	10 HOURS						
03h	0	0	0	0	0	DAY			01-07
04h	0	0	10 DATE		DATE				01-31*
05h	0	0	0	10 MONTH	MONTH				01-12
06h	10 YEAR				YEAR				00-99

* DATE BYTE MAXIMUM VALUE RANGES FROM 28 TO 31, DEPENDING ON MONTH AND YEAR

Tabla 7.10. Tabla del registro CLOCK_REG_FORMAT del reloj

Como se puede observar en la tabla, los datos referentes a los segundo y los minutos se encuentran en las posiciones 0x00 y 0x01 del registro 0xC0. El formato tanto de las unidades como de las decenas es BCD. El dato referente a las horas se encuentra en la posición 0x03 también en formato BCD. A diferencia de los segundos y los minutos, en las horas las decenas y las unidades se guardan en la misma posición.

Los valores de temperatura obtenidos se encuentran en el registro TEMPERATURE_DATA situado en la posición 0xAA. Se trata de un registro de 2 bytes en el cual el byte más significativo corresponde al valor entero, con signo, de la temperatura y el bit más significativo del byte menos significativo representa una resolución de ±0.5°C.

S	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	MSB
(unit = °C)								LSb
2 ⁻¹	0	0	0	0	0	0	0	LSB

Tabla 7.11. Tabla del registro TEMPERATURE_DATA del reloj

7.5. IMU MTi-28A53G25

7.5.1. SISTEMA DE COMUNICACIÓN

Este modelo en concreto, MTi-28A53G25, transmite los datos obtenidos con el sistema de comunicación asíncrono RS-232. Para que la señal sea compatible con el microcontrolador (niveles TTL) será necesario incluir un adaptador entre ambos. Seleccionamos el chip MAX 3232 de la marca MAXIM que permitirá la correcta comunicación entre estos dos dispositivos alimentados a dos tensiones diferentes (3.3v y 5v).

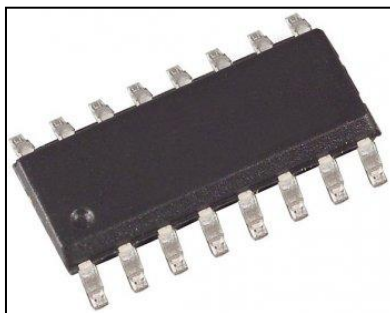


Figura 7.7. MAX 3232

7.5.2. CONECTOR

Para realizar la conexión física con el IMU necesitamos un conector compatible con el mismo. En el catálogo de LEMO CONNECTORS encontramos un conector que reúne las características necesarias para realizar la conexión (FGG-0B-307-CLAD52). De los 7 pines con los que cuenta el conector solo usaremos 4, los correspondientes a VCC, GND, TX y RX.

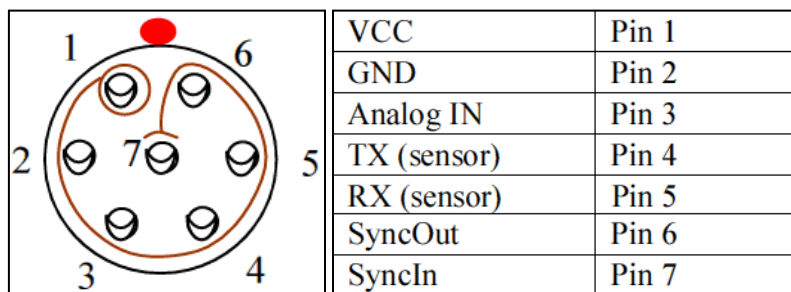


Figura 7.8. Pines conector IMU

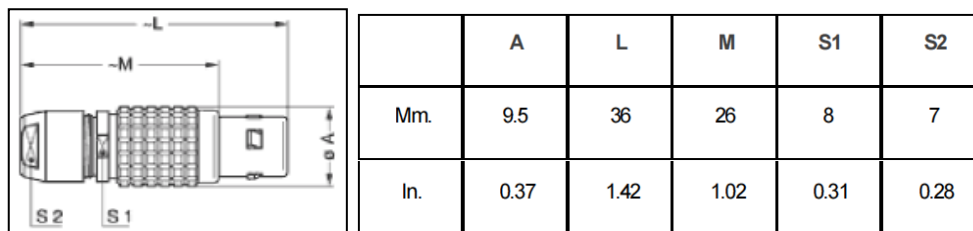


Figura 7.9. Conector IMU

7.5.3. PROTOCOLO DE COMUNICACIÓN

La comunicación con el IMU debe realizarse de acuerdo a un protocolo establecido por el fabricante del dispositivo. Dicho protocolo consiste en una trama de datos organizada en campos que se define a continuación:

PREAMBLE	BID	MID	LEN	DATA	CHECKSUM
----------	-----	-----	-----	------	----------

Tabla 7.12. Tabla de formato de protocolo del IMU

- **PREAMBLE:** Se trata de un byte que indica el inicio del paquete de datos. Siempre será 0xFA.
- **BID:** Representa la dirección del BUS de comunicación. Está compuesto por un byte que varía entre 0xFF ó 0x01, dependiendo del comando.
- **MID:** Un byte que representa el comando deseado. La lista de todos comandos válidos se encuentra en el documento MT “Low-Level Communication Protocol Documentation”.
- **LEN:** Tamaño del campo DATA representado en un byte.
- **DATA:** Campo en el que se representan los datos solicitados. Su tamaño varía dependiendo del modo que esté configurado el IMU (Máximo 254 bytes).
- **CHECKSUM:** Byte de confirmación que se usa para la detección de errores en la transmisión. Si el byte menos significativo de la suma de BID, MID, LEN, DATA y CHECKSUM es igual a 0, la transmisión fue correcta.

En la mayoría de los casos el IMU responde a cada envío con otro paquete de datos que confirma la transmisión. Esta respuesta es idéntica al envío salvo en dos campos, el MID y el CHECKSUM. El MID se incrementará en uno en la respuesta dando lugar al cálculo de un nuevo CHECKSUM.

7.5.4. CONFIGURACIÓN INICIAL

El IMU cuenta con una gran variedad de opciones configurables para obtener los datos requeridos con el formato y las características que más se ajusten a nuestras necesidades. En su interior cuenta con una memoria no volátil donde se almacena la última configuración actualizada. Gracias al software MT Manager de XSENS podremos cambiar dichas opciones.

Para conectar el IMU a un PC será necesario el cable RS-232 MTi cable (CA-USB2) suministrado con el propio IMU.

Una vez alimentado, el IMU puede estar en dos estados, modo configuración o modo medida. Si en 500ms no se establece lo contrario, el IMU pasará a modo medida, en el cuál comenzará a transmitir datos con el formato y velocidad que estipula la última configuración escrita en la memoria no volátil del dispositivo. Los parámetros por defecto se muestran en la siguiente tabla.

Output mode	Orientation output
Output settings	Orientation in quaternion mode Sample counter enabled
Sample frequency	100 Hz
Baudrate	115k2 bps
Output skip factor	0
SyncIn	Disabled
SyncOut	Disabled

Tabla 7.13. Tabla configuración del IMU

Para cambiar algún término de la configuración se recomienda pasar primero al modo configuración, enviando el correspondiente comando. Una vez realizado el cambio se puede pasar de nuevo al modo medida, en el cuál la nueva configuración se aplicará al formato y velocidad de los datos transmitidos.

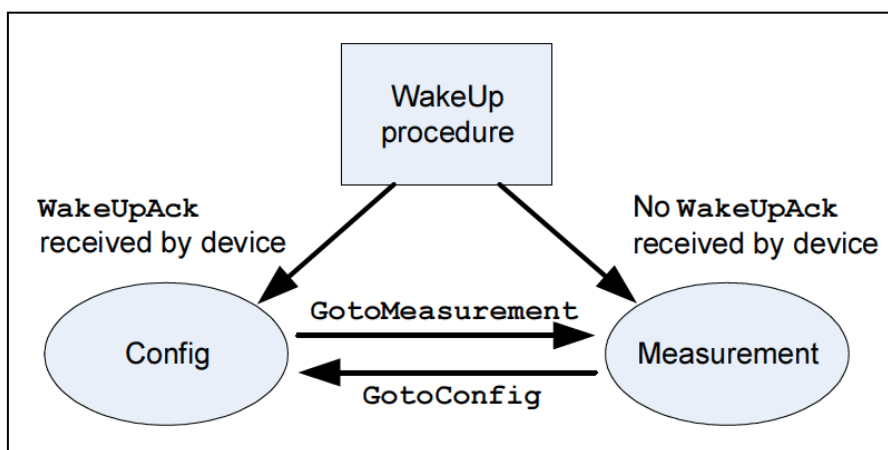


Figura 7.10. Esquema de inicio del IMU

De todas las opciones configurables, necesitaremos establecer 3 para que el IMU trabaje correctamente con el microcontrolador.

En primer lugar configuramos el IMU para que transmita y reciba datos a 9600 baudios por el puerto serie. Es la velocidad más baja a la que puede transmitir, sin embargo una velocidad superior supondría pérdida de datos en la comunicación asíncrona con el microcontrolador. Para cambiar la velocidad a 9600 baudios será necesario enviar el comando "Setbaudrate" 0xFA 0x01 0x18 0x01 0x09 0xDB.

En segundo lugar estableceremos qué datos queremos que el IMU transmita. Para este caso seleccionamos “calibrated data output mode”, en el cuál obtenemos los datos de aceleración, orientación y campo magnético calibrados de acuerdo a un algoritmo que tiene en cuenta entre otras cosas la temperatura del dispositivo. Para cambiar a este modo será necesario enviar el comando “SetOutputMode” 0xFA 0x01 0xD0 0x02 0x00 0x02 0x2B.

En este formato cada campo es un dato de 4 bytes de tipo float (IEEE 754 standard).

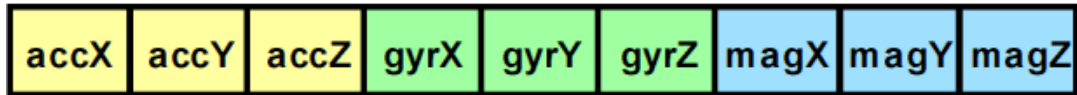
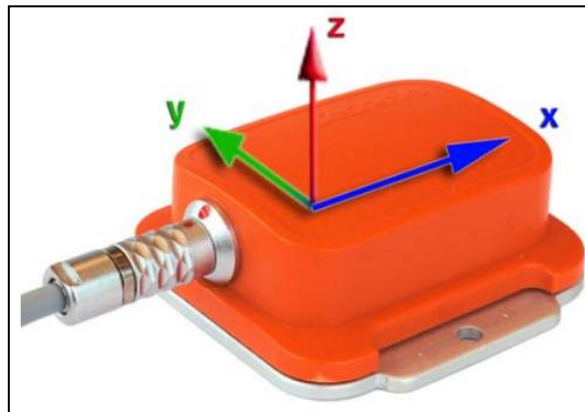


Tabla 7.14. Tabla del formato de datos IMU

El sentido de los ejes y las unidades de los datos se pueden observar en las siguientes imágenes.



Acceleration	m/s ²
Angular velocity (rate of turn)	rad/s
Magnetic field	a.u. (arbitrary units) normalized to earth field strength

Figura 7.11. Unidades dirección y sentido de datos IMU

Por último será necesario modificar el SKIPFACTOR. Este parámetro actúa como un prescaler para obtener diferentes frecuencias de salida de datos. La frecuencia mínima a la que el IMU muestrea los datos es 100Hz, si se desea una frecuencia de salida menor se modifica el SKIPFACTOR para que no todos los datos muestreados se transmitan. En este caso estableceremos el máximo valor, 0xFFFF, para que en el modo medida no se transmita ningún dato. Para configurar el SKIPFACTOR será necesario enviar el comando “SetOutputSkipFactor” 0xFA 0x01 0xD4 0x02 0xFF 0xFF 0x2B.

Cuando sea necesaria la transmisión de un paquete de datos le enviaremos el comando “MTData” 0xFA 0x01 0x34 0x00 0xCB y el dispositivo devolverá un solo paquete de datos de manera puntual en el formato y velocidad en el que lo hayamos configurado.

7.6. Tarjeta SD

7.6.1. SISTEMA DE COMUNICACIÓN

Como se comenta en el apartado anterior, la comunicación entre el microcontrolador y la tarjeta SD se realizará a través del protocolo de comunicación síncrono SPI. Se trata de un tipo de comunicación FULL-DUPLEX, por lo que podemos transmitir y recibir datos a la vez, ahorrando tiempo en la comunicación. Esta característica es posible ya que existe un pin para recibir datos, MISO, y otro para transmitir datos, MOSI. Además, este tipo de comunicación incluye un pin de reloj, SCLK, para sincronizar la comunicación; y un pin de selección de dispositivo, CS, de nivel activo bajo. Por último, cuenta con un pines de alimentación, 3,3v, y dos de masa, GND. Los dos pines sobrantes se usan en el modo de comunicación SD.



Figura 7.12. Pines tarjeta SD

Existen 4 tipos de comunicación SPI dependiendo de la señal de reloj SCLK y del modo de transmisión. Según el estándar SD se debe utilizar el tipo 3 e incluir una resistencia de PULL-UP en cada patilla de conexión.

7.6.2. CONECTOR

Para realizar la conexión física con el microcontrolador utilizaremos un adaptador microSD-SD y un conector SD para PCB.

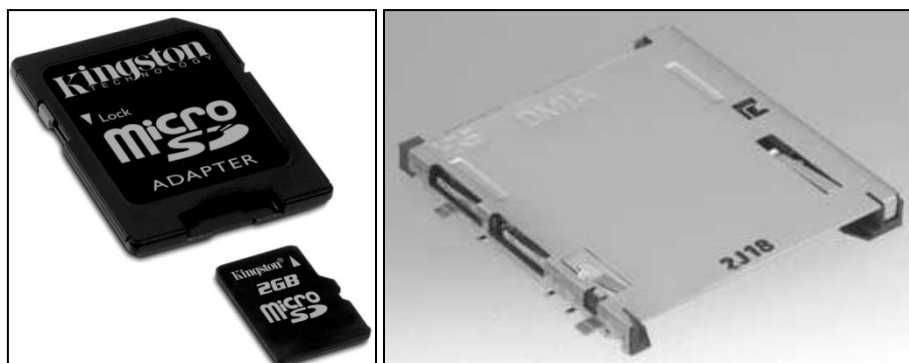


Figura 7.13. Conector tarjeta SD

7.6.3. PROTOCOLO DE COMUNICACIÓN

La comunicación con la tarjeta SD debe realizarse de acuerdo a un protocolo establecido por el estándar SD. Dicho protocolo consiste en una trama de datos organizada en campos que se define a continuación:

Byte 1				Byte 2 - 5				Byte 6	
7	6	5	0	31	0	7	0		
0	1	Comand		Comand		CRC	1		

Tabla 7.15. Tabla del protocolo tarjeta SD

Cada paquete de datos se organiza en 6 bytes.

El primer byte corresponde al número del comando del mensaje, que deberá escribirse en los 5 bits menos significativos del byte. Los dos bits más significativos siempre serán '0' y '1', ya que indican el inicio de la transmisión.

Los siguientes 4 bytes corresponden al argumento, su significado dependiendo del comando (ej. Dirección de lectura).

El último byte es un byte de verificación. Solo se usa en modo de comunicación SD y se calcula de acuerdo a un algoritmo establecido en el estándar SD.

Cada vez que se envía un paquete a la tarjeta, ésta responde con un mensaje de confirmación. Dependiendo del comando enviado la respuesta puede variar entre R1, R2 o R3. Las diferencia entre las repuesta radica en el tamaño y en la información que suministran. En nuestro caso la respuesta recibida siempre será R1, la cual nos informará de si hubo algún fallo en la transmisión y de qué tipo fue.

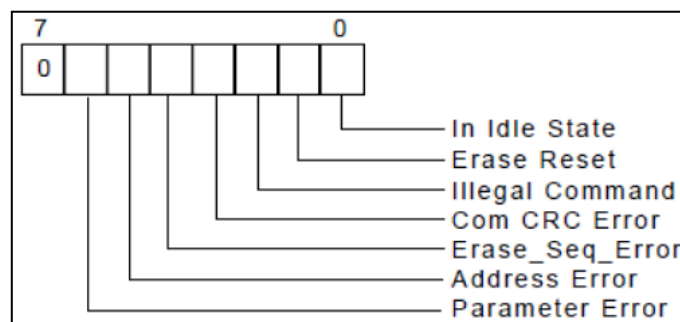


Figura 7.14. Protocolo de tarjeta SD

7.6.4. CONFIGURACIÓN INICIAL

Para poder almacenar datos en la tarjeta SD será necesario inicializarla y configurarla primero. Para ello se seguirán los siguientes pasos:

- Alimentamos la tarjeta a 3.3v y esperamos al menos 1ms para que se estabilice. A continuación le mandamos 74 pulsos de reloj para que la memoria se inicialice.
- Mandamos el comando 0 y esperamos su respuesta, que puede tardar algunos ciclos de reloj. Si la respuesta R1 es 0x01 significa que la memoria se a reseteado en modo SPI y está en modo IDLE. Si la respuesta no es la deseada puede ser un problema de compatibilidad.
- A continuación mandamos el comando 8 con el argumento 0x1AA para indicarle a la tarjeta que la tensión de trabajo es de 3.3v. Si la tarjeta responde con la repuesta R7 y el mismo argumento que le mandamos, la tarjeta admite la tensión de trabajo y está lista para ser inicializada. En caso contrario puede tratarse de otra versión de tarjeta SD que no admite la tensión de trabajo de 3.3v.
- Lo siguiente que se debe enviar es el comando ACM41, que corresponde a los comandos 55 y 41. Con esta trama la tarjeta sale del modo IDLE y se puede iniciar la lectura/escritura de la misma. Es probable que la tarjeta no admita este comando a la primera, por lo que será necesario iniciar un bucle de al menos 1s.
- Por último mandamos el comando 16 para indicarle a la tarjeta que organice la memoria con sectores de 512 bytes.

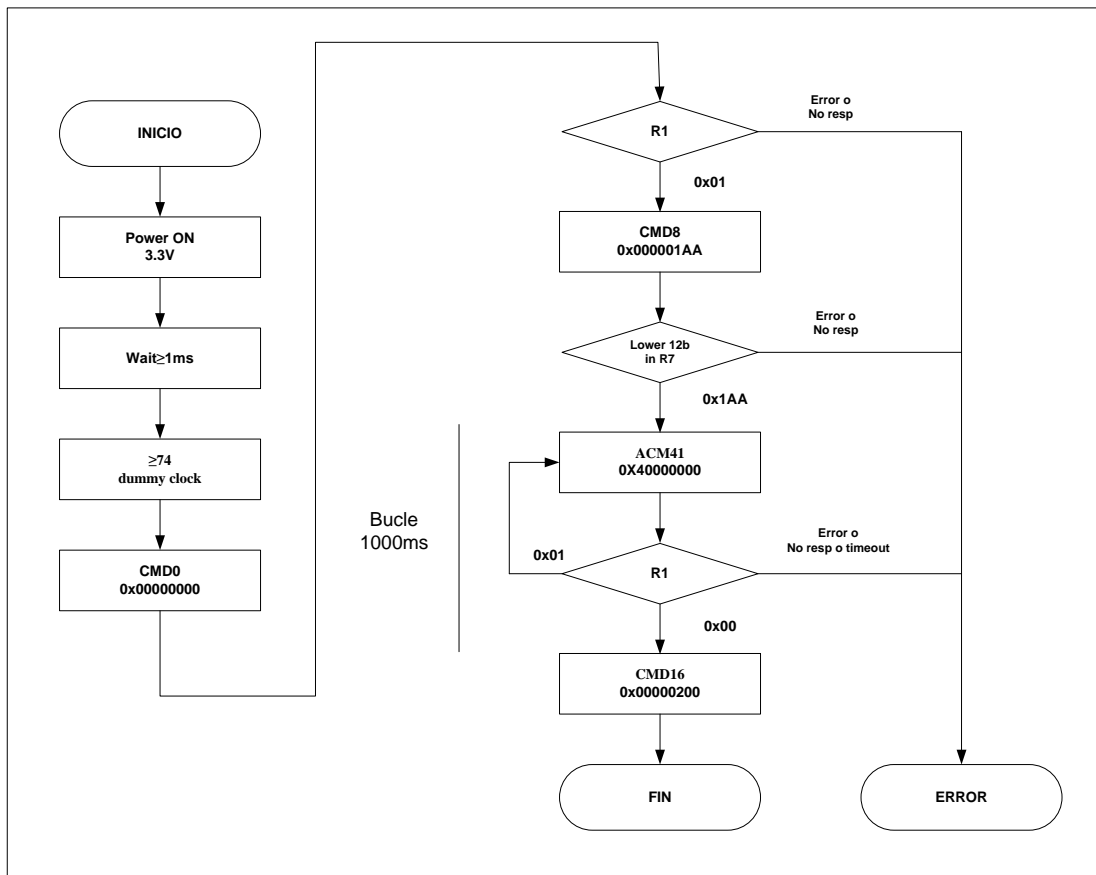


Figura 7.15. Diagrama de flujo de inicialización tarjeta SD

7.6.5. SISTEMA DE ARCHIVOS FAT

Una vez inicializada la tarjeta se pueden guardar datos en los sectores de la misma, sin embargo no resulta funcional puesto que sería necesario un microcontrolador para leer la información guardada. Con el sistema de archivo FAT (File Allocation Table) podremos crear un archivo .txt y guardar la información en formato de carácter. De esta manera el volcado de datos a un PC resulta mucho más sencillo y efectivo.

Cuando se formatea una unidad removible se crea una partición en el primer sector físico de la memoria, a esta partición se la conoce por sector de arranque. Aquí podemos encontrar 512 bytes con información relativa a la distribución y estructura de la memoria de la unidad. Leer e interpretar esta información será necesario para saber dónde podemos crear un archivo dentro de la memoria.

Para llevar a cabo este proceso el compilado de CCS cuenta con una librería que se encarga de inicializar la tarjeta en modo SPI y leer la FAT para poder crear y borrar archivos.

7.7. Microcontrolador 18F27J13

7.7.1. TABLA DE PINES DEL MICROCONTROLADOR

A continuación se muestran una tabla donde se presentan y describen los pines utilizados en el microcontrolador:

PIN	PUERTO	ETIQUETA	DESCRIPCIÓN
2	RA0	P1	Pulsador de cambio de estado, sentido horario.
3	RA1	P2	Pulsador de cambio de estado, sentido anti horario.
4	RA2	-	-
5	RA3	VIN_REF	Tensión de referencia de la batería.
-	RA4	-	-
7	RA5	BAT	LED de aviso de carga.
10	RA6	OSC1	Entrada para el oscilador.
9	RA7	OSC2	Entrada para el oscilador.
21	RB0	MOSI	Entrada de datos del BUS SPI.
22	RB1	MOSO	Salida de datos del BUS SPI.
23	RB2	SCLK	Señal de reloj del BUS SPI.
24	RB3	CS	Selector de dispositivo esclavo del BUS SPI.
25	RB4	TX2	Salida de datos BUS RS232 con el dispositivo USB.
26	RB5	RX2	Entrada de datos BUS RS232 con el dispositivo USB.
27	RB6	PGC	Entrada de programación.
28	RB7	PGD	Entrada de programación.
11	RC0	LED1	LED de cambio de estado INACTIVO.
12	RC1	LED2	LED de cambio de estado COMUNICACIÓN.
13	RC2	LED3	LED de cambio de estado REGISTRO.
14	RC3	SCL	Señal de reloj BUS I2C.
15	RC4	SDA	Entrada y salida de datos BUS I2C.
16	RC5	-	-
17	RC6	TX1	Salida de datos BUS RS232 con el IMU.
18	RC7	RX1	Entrada de datos BUS RS232 con el IMU.

Tabla 7.16. Tabla de pines del microcontrolador

8. ANEXO II: MANUAL DEL PROGRAMADOR

Se detallan las características principales, funciones, rutinas e interrupciones utilizadas en el programa de control. Además en este capítulo se detalla el formato de los datos muestreados y los diagramas de flujo del programa.

8.1. Introducción

El software a realizar consiste en un programa que el microcontrolador ejecutará para realizar una tarea cíclica y autónoma. El lenguaje que se empleará para programar el microcontrolador será C, ya que facilitará en gran medida la implementación de instrucciones para desarrollar las comunicaciones y el programa en general. Será necesario el uso de un compilador que traduzca el programa a código máquina que será grabado en la memoria interna de programa, no volátil, para que el microcontrolador pueda ejecutarlo.

El entorno de trabajo que se usará para desarrollar el código será MPLAB v8.80 de MICROCHIP, mientras que el compilador será CCS v4.130. Para depurar y programar el código en el microcontrolador se usará el programador PICKIT3 de la marca MICROCHIP.

El código del programa está adjunto en el CD.

8.2. Estructura y Funcionalidades

El programa de control deberá ser capaz de implementar las siguientes funcionalidades a desarrollar por el microcontrolador:

- Configurar e inicializar las comunicaciones serie: síncrona y asíncrona.
- Configurar e inicializar las características de la toma de medidas y las comunicaciones de los sensores.
- Recibir y enviar datos a los sensores e IMU.
- Enviar datos a la tarjeta SD.
- Recibir y enviar datos a un PC.

Para ello el programa consistirá en un archivo principal, main.c, donde se desarrolla todas las tareas de manera cíclica, y una serie de librerías que dotarán a dicho programa de las funciones necesarias para implementar todas las funcionalidades antes mencionadas. El propio nombre de las librerías indica la utilidad de las funciones que incluyen. Las librerías fat.c y mmcsd.c pertenecen al compilador CCS.

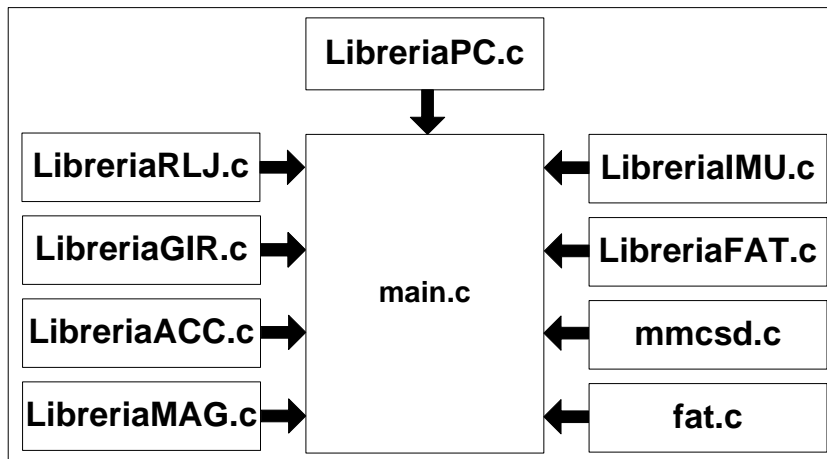


Figura 8.1. Esquema de librerías

Las librerías RLJ, GIR, ACC, MAG, IMU, PC y FAT contiene cada una las funciones y constantes necesarias para desarrollar las características antes mencionadas, de tal manera que se puede hacer uso de ellas desde el “main” a través de una llamada a función. De esta manera el programa está estructurado por módulos, lo que simplifica en gran medida el programa principal y resulta más asequible para su comprensión, depuración en caso de error y posible modificación.

Las librerías fat y mmcsd forman parte del compilador CCS, en ellas podemos encontrar todas las funciones necesaria para comunicarnos con una tarjeta SD o MMC, y todo lo necesario para desarrollar un sistema de archivos FAT ellas.

8.3. Módulos Internos

A continuación se resumen los módulos internos del microcontrolador utilizados:

- Contador/temporizador TIMER0: Se utiliza para solicitar los datos correspondientes de cada sensor a una frecuencia fija a través de su interrupción.
- Comunicación serie síncrona MSSP (MASTER SYNCHRONOUS SERIAL PORT): Se utiliza para implementar las comunicaciones serie síncronas I2C y SPI para comunicarse con los sensores inerciales y la tarjeta SD.
- Comunicación serie asíncrona EUSART (ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER): Se utilizará el EUSART1 para implementar las comunicaciones con el IMU, mientras que el EUSART2 se utilizará para implementar las comunicaciones con un PC a través del módulo UMR232.
- Mapeo de pines PPS (Peripheral Pin Select): Este microcontrolador incorpora un sistema a través del cual podemos configurar los pines de algunos puertos para que desarrollen las funciones de los módulos antes mencionados. De esta manera configuramos los pines RB0, RB1 y RB2 como MOSO, MOSI, y SCLK respectivamente para implementar la comunicación SPI; RA0 y RA1 como INT1 e IN2 respectivamente para activar las interrupciones externas de los pulsadores; y RA5 como C1OUT para configurarlo como la salida del módulo comparador.

- **Módulo comparador:** Se utilizará este módulo para monitorizar el voltaje de la batería y lanzar una señal de aviso cuando su nivel de tensión sea bajo. La tensión de la batería se comparará con una referencia interna de tensión generada por el propio microcontrolador.
- **Tensión de referencia interna:** Este módulo generará una señal de referencia para compararla con la tensión de la batería a través del módulo comparador. Cuando el voltaje de la batería sea inferior al de referencia se encenderá un LED.
- **Interrupciones:** Será necesario programar 5 interrupciones, la generada por el TIMER0 (INT_T0) para muestrear a una frecuencia fija, 2 interrupciones por recepción de dato en el puerto serie (INT_RDA e INT_RDA2) para la comunicación con el IMU y el PC; y dos interrupciones externas (INT_EXT1) y (INT_EXT2) para detectar el cambio de estado del sistema a través de dos pulsadores. Para activarlas será necesario habilitar la máscara global de interrupción. Será necesario también priorizar las 2 interrupciones del puerto serie para que puedan interrumpir a otras interrupciones.

8.4. Programa Principal

Se trata del archivo principal, donde está el main(). Lo primero que se incluye es el archivo main.h, donde se introduce la cabecera del microcontrolador, se establecen los bits de configuración a través de las directivas #FUSES, se mapean los pines que se van a emplear con la directiva #PIN_SELECT y se inicializan las comunicaciones a través de las instrucciones #USE.

A continuación se establece el tipo de FAT a utilizar (FAT 16) y se incluyen todas las librerías que se van a usar. Por último antes del main() se da nombre a los registros necesarios para poder cargarlos con el valor adecuado en la inicialización.

Una vez en el main(), se realizan todas las configuraciones iniciales. A partir de este punto el programa entra en un bucle de ejecución infinita donde se evalúa en qué estado se encuentra el sistema (Inactivo, registrando o comunicación).

En modo inactivo, por defecto, el programa se encuentra a la espera de que se produzca un cambio en el estado del sistema, dicho cambio se detectará a través de una de las dos interrupciones externas programadas asociadas a dos pulsadores (P1 Y P2). Si se pasa al estado comunicación, se habilita la comunicación serie con el PC para mandar y recibir datos, mientras que si se pasa al estado registrando, se encenderá el timer0 para que comience el muestreo de datos. Para que el usuario pueda conocer el estado actual del sistema, se asociarán tres diodos LEDs a cada estado.

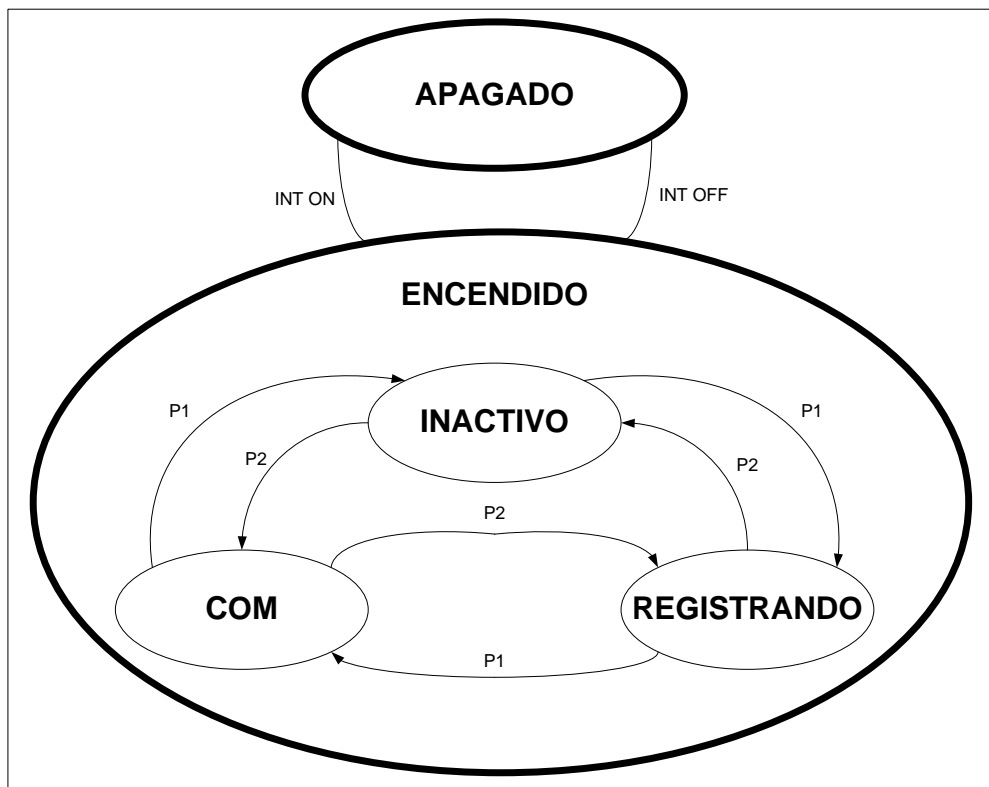


Figura 8.2. Esquema de estados

8.5. Formato de los datos almacenados

Se creará un archivo.txt dentro de la tarjeta SD donde se almacenarán los datos obtenidos por los sensores. La distribución de la información dentro del archivo de texto se detalla a continuación:

SENSORES INERCIALES INDIVIDUALES													
	horas	minutos	segundos	temperatura	acc X	acc Y	acc Z	v ang X	v ang Y	v ang Z	c mag X	c mag Y	c mag Z
	8bits	8bits	8bits	32 bits	10 bits			16 bits			12 bits		
	entero			flotante	entero Complemento a 2								
	24h	59min	59s	-55°C/125°C	±4G			±2000°/s			±8 Gauss		
Muestra1	horas	minutos	segundos	°C	m/s ²			°/s			mGauss		

IMU								
acc X	acc Y	acc Z	v ang X	v ang Y	v ang Z	c mag X	c mag Y	c mag Z
32 bits								
flotante								
±5G			±300°/s			±7.5 Gauss		
m/s ²			°/s			mGauss		

Tabla 8.1. Tabla de formato original de los datos

Tanto el tamaño en bits como el tipo de dato de la tabla son el formato original que se recoge de cada sensor. Una vez adquiridos, los datos procedentes de los sensores inerciales individuales son convertidos a sus respectivas unidades teniendo en cuenta su resolución y rango de medida. Finalmente y para poder guardar todos los datos en la tarjeta SD, se introducen todos los datos en un vector de caracteres.

Cada muestra, teniendo en cuenta espacios, tabulaciones y datos, ocupa entre 107 y 125 bytes en la tarjeta, este margen se debe a que algunos datos pueden ser negativos o no, lo que implica un byte para el signo '-' en cada dato.

8.6. Interrupciones

A continuación se resumen las interrupciones implementadas. Todas las rutinas de interrupción se encuentran en el archivo main.c:

- **Interrupción por recepción de dato por el puerto serie:** Se activa cada vez que se reciben 8 bits por el puerto serie. Se utiliza en dos estados diferentes. En el estado 'comunicación', cada vez que enviamos información del PC al microcontrolador; y en el estado 'registrando', cada vez que el IMU manda información al microcontrolador. En ambos casos la comunicación se realizará a 9600 baudios y sin bit de paridad.
- **Interrupción del TIMER0:** Se activa cada vez que se desborda el contador del TIMER0. De esta manera controlamos la frecuencia de muestreo de los datos almacenados. Esta interrupción sólo se produce en el estado 'registrando'. Cada vez que el programa entra en esta interrupción, se solicita un bloque de datos para posteriormente almacenarlos en la tarjeta SD.
- **Interrupciones externas:** Se programan dos interrupciones externas, que se activan cada vez que uno de los pulsadores asociados a los correspondientes pines del microcontrolador es pulsado. Dicha activación se produce por flanco de bajada y en su rutina se gestionará el cambio de estado a través de una variable global.

8.7. Rutinas y funciones

El programa principal hace uso de una serie de funciones que se alojan en las diferentes librerías que se incluyen en el proyecto. Dichas funciones están organizadas por bloques y su propio nombre indica a que bloque pertenecen y qué función desempeñan. A continuación se realiza una descripción de dichas funciones:

- **LibreríaACC.c (Librería creada por el programador)**
 - void escribe_sensor_ACC(int dir, int dato), escribe un dato de 8bits a partir de la dirección deseada. La dirección del acelerómetro está definida por defecto.
 - signed long leer_dato_ACC(int direccion), devuelve un dato de 16 bits con signo, siendo la dirección enviada la ubicación de la parte menos significativa del dato.
 - void pide_datos_aceleracion_ACC(float acc[]), esta función llama a su vez tres veces a la función leer_dato_ACC() para almacenar los datos de

aceleración en una tabla de tres elementos flotantes, cuya dirección de inicio se manda en el argumento.

- void inicializa_sensor_ACC(), esta función llama a su vez a la función escribe_sensor_ACC para configurar los correspondientes registros del sensor.

- **LibreríaGIR.c(Librería creada por el programador)**

- void escribe_sensor_GIR(int dir, int dato), escribe un dato de 8bits a partir de la dirección deseada. La dirección del giróscopo está definida por defecto.
- signed long leer_dato_GIR(int direccion), devuelve un dato de 16 bits con signo, siendo la dirección enviada la ubicación de la parte más significativa del dato.
- void pide_datos_giro_GIR(float gir[]), esta función llama a su vez tres veces a la función leer_dato_GIR() para almacenar los datos de velocidad angular en una tabla de tres elementos flotantes, cuya dirección de inicio se manda en el argumento.
- void inicializa_sensor_GIR(), esta función llama a su vez a la función escribe_sensor_GIR para configurar los correspondientes registros del sensor.

- **LibreríaMAG.c(Librería creada por el programador)**

- void escribe_sensor_MAG (int dir, int dato), escribe un dato de 8bits a partir de la dirección deseada. La dirección del magnetómetro está definida por defecto.
- signed long leer_dato_MAG (int direccion), devuelve un dato de 16 bits con signo, siendo la dirección enviada la ubicación de la parte más significativa del dato.
- void pide_datos_campomag_MAG(float mag[]), esta función llama a su vez tres veces a la función leer_dato_MAG () para almacenar los datos de velocidad angular en una tabla de tres elementos flotantes, cuya dirección de inicio se manda en el argumento.
- void inicializa_sensor_MAG (), esta función llama a su vez a la función escribe_sensor_MAG para configurar los correspondientes registros del sensor.

- **LibreríaRLJ.c (Librería creada por el programador)**

- void escribe_sensor_RLJ (int dir, int dato), escribe un dato de 8bits a partir de la dirección deseada. La dirección del reloj está definida por defecto.
- void escribe_tiempo_RLJ(int dir_1,int dir_2,int dato_1), inicializa la cuenta del reloj escribiendo un 0 en las dos direcciones que se mandan en el argumento.

- signed long leer_dato_RLJ (int direccion), devuelve un dato de 16 bits con signo correspondiente a la temperatura, siendo la dirección enviada la ubicación de la parte más significativa del dato.
- int leer_tiempo_RLJ(int dir_1,int dir_2), devuelve un dato de 8bits correspondiente al tiempo almacenado en una dirección de 16 bits.
- float pide_datos_temperatura_RLJ(),esta función llama a su vez a la función leer_dato_RLJ () para procesar y devolver el datos flotante de temperatura.
- void pide_datos_tiempo_RLJ(float tiempo[]), esta función llama a su vez tres veces a la función leer_tiempo_RLJ () para almacenar los datos de tiempo en una tabla de tres elementos flotantes, cuya dirección de inicio se manda en el argumento.
- void inicializa_sensor_RLJ (), esta función llama a su vez a la función escribe_sensor_RLJ y escribe_tiempo_RLJ para configurar los correspondientes registros del sensor.

- **LibreríaIMU.c(Librería creada por el programador)**

- void genera_datos_IMU(), manda la trama de datos necesaria para que el IMU responda con los datos solicitados a través del puerto serie.
- void recibe_dato_IMU(float datos_IMU[]), esta función es llamada cada vez que se activa la interrupción por recepción de dato por el puerto serie. En ella se van recogiendo los datos recibidos en la trama enviada por el IMU y se almacenan en la tabla cuya dirección de inicio se envía en el argumento.

- **LibreríaFAT.c(Librería creada por el programador)**

- void crear_directorio_archivo_FAT(), se llama a esta función al inicio del programa para crear una carpeta y un archivo de texto en la tarjeta SD. En esta función se usan a su vez funciones alojadas en la librería fat.c.
- void escribe_cadena_FAT(char cadena[]), escribe una cadena de caracteres en el archivo de texto previamente creado. En esta función se usan a su vez funciones alojadas en la librería fat.c.
- void borrar_archivo_FAT(), borra un archivo de la tarjeta SD previamente creado.
- void nuevo_archivo_FAT(), crea un archivo nuevo en la tarjeta SD.
- void volcar_datos_FAT(), inicia una descarga de datos de la tarjeta SD al PC, dicha descarga se puede anular pulsando uno de los botones de cambio de estado.

- **LibreríaPC.c(Librería creada por el programador)**

- void procesa_tecla_PC(), gestiona y procesa los caracteres introducidos a través del teclado del PC. Cuando se pulsa la tecla ENTER, se llama a la función procesa_texto_PC para evaluar el comando.

- void procesa_texto_PC(char texto[]), compara el comando recibido con una lista de comandos registrados y actúa en consecuencia.

- **fat.c(Librería propia del compilador CCS)**

- fat_init(), inicializa la tarjeta SD en modo SPI y accede al sector 0 de la tarjeta para recoger información necesaria para poder crear archivos y carpetas dentro de ella. Devuelve 0 si todo ha sido correcto.
- mk_dir(path), crea una carpeta con el nombre incluido en la variable path. Devuelve 0 si todo ha sido correcto.
- mk_file(path), crea un archivo cuya ubicación y extensión se indican en la variable path. Devuelve 0 si todo ha sido correcto.
- fopen (char *name, char *mode, FILE *fstream), abre un archivo ya creado y le atribuye los permisos pertinentes.
- fclose(FILE *fstream) , cierra un archivo previamente abierto.
- fputs(char* str, FILE *fstream), escribe una cadena de caracteres en un archivo previamente creado.

8.8. Diagramas de Flujo

A continuación se presentan los diagramas de flujo correspondientes a las diferentes partes del programa de control, comenzando por los diagramas más generales.

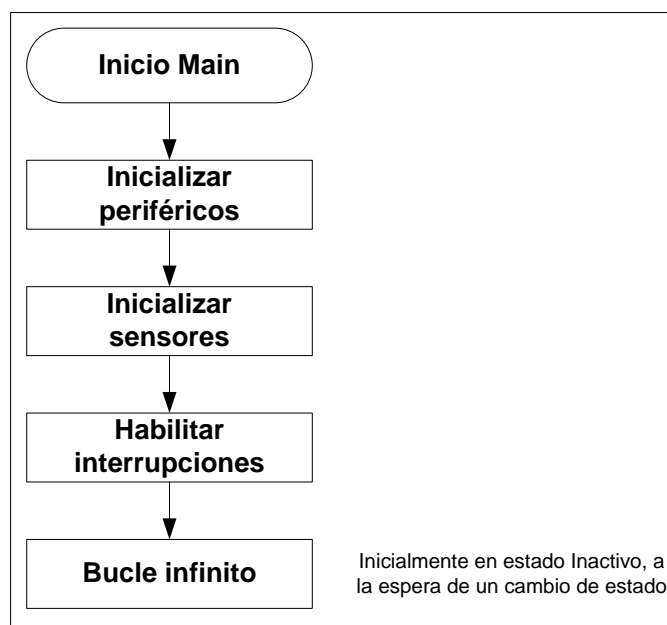


Figura 8.3. Diagrama general

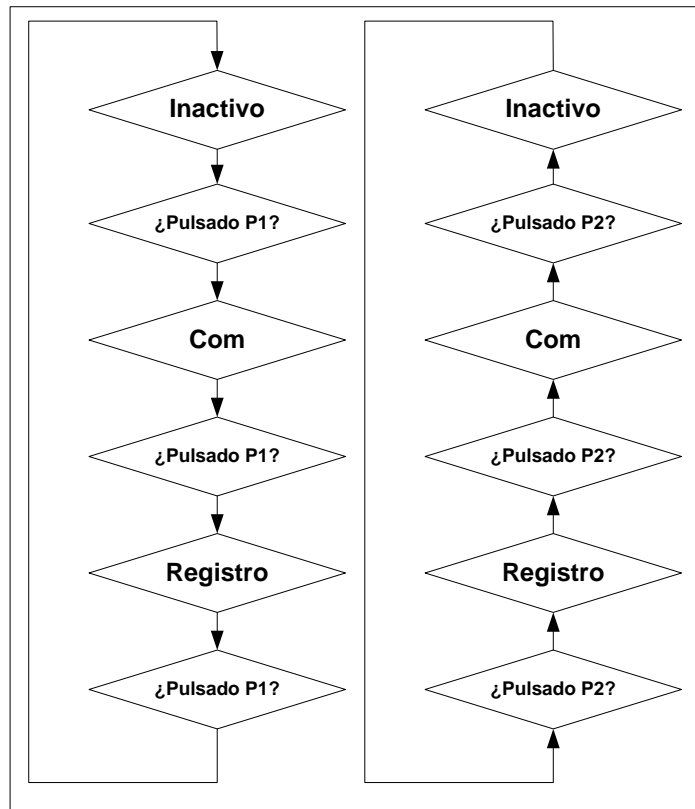


Figura 8.4. Diagrama de cambio de estado

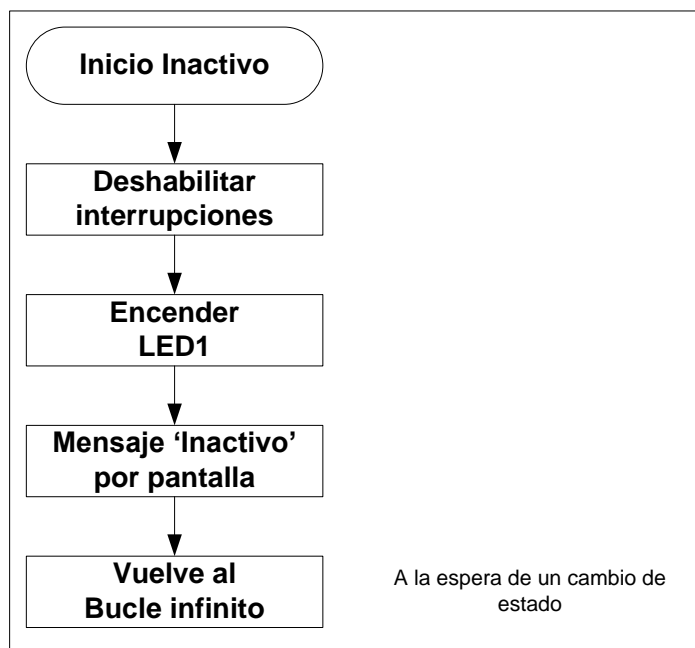


Figura 8.5. Diagrama de estado inactivo

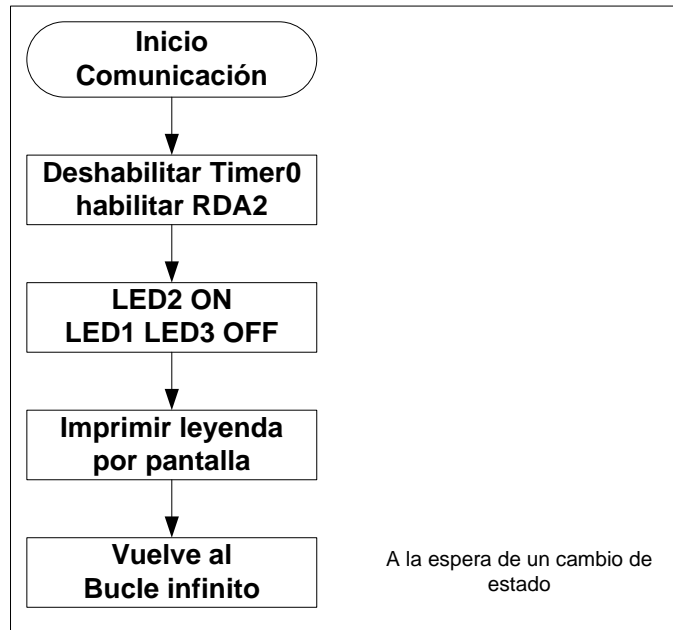


Figura 8.6. Diagrama estado comunicación

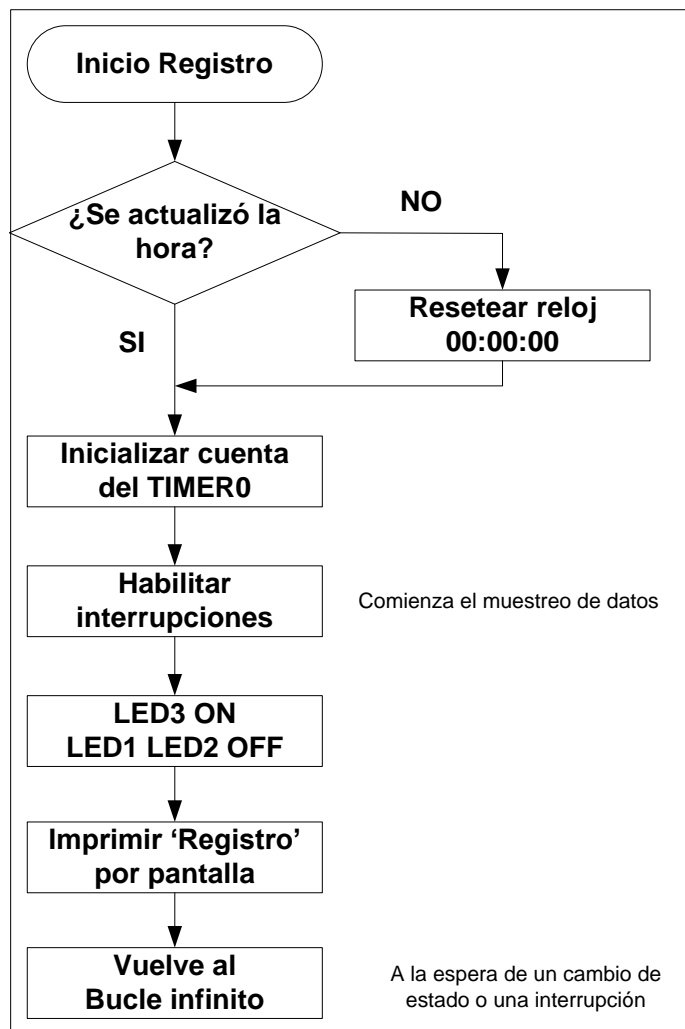


Figura 8.7. Diagrama estado registro

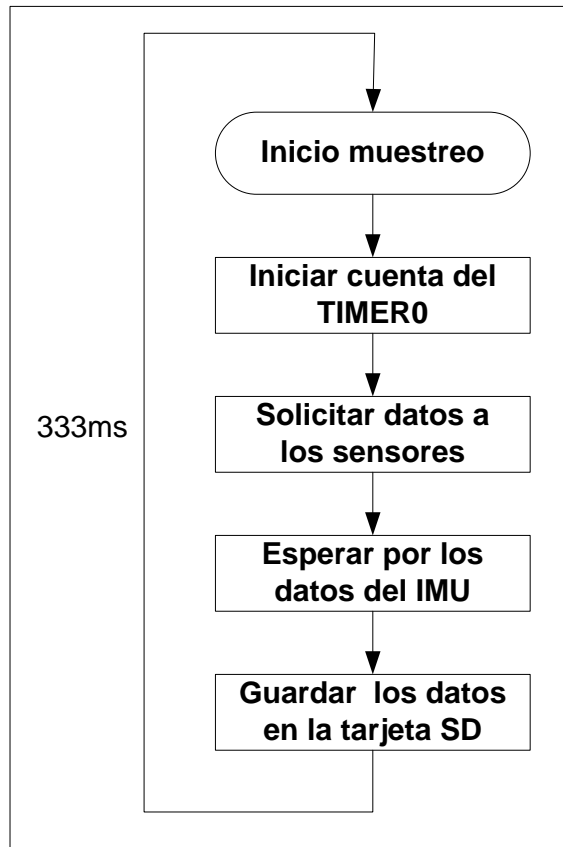


Figura 8.8. Diagrama muestreo de datos

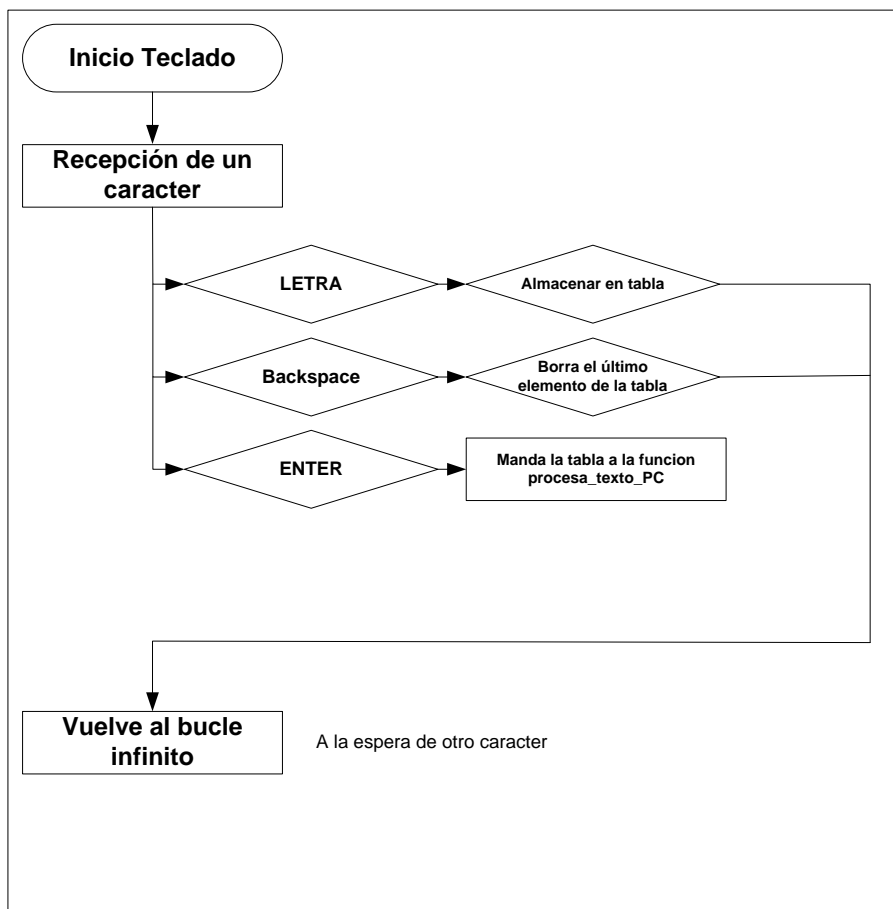


Figura 8.9. Diagrama pulsar tecla

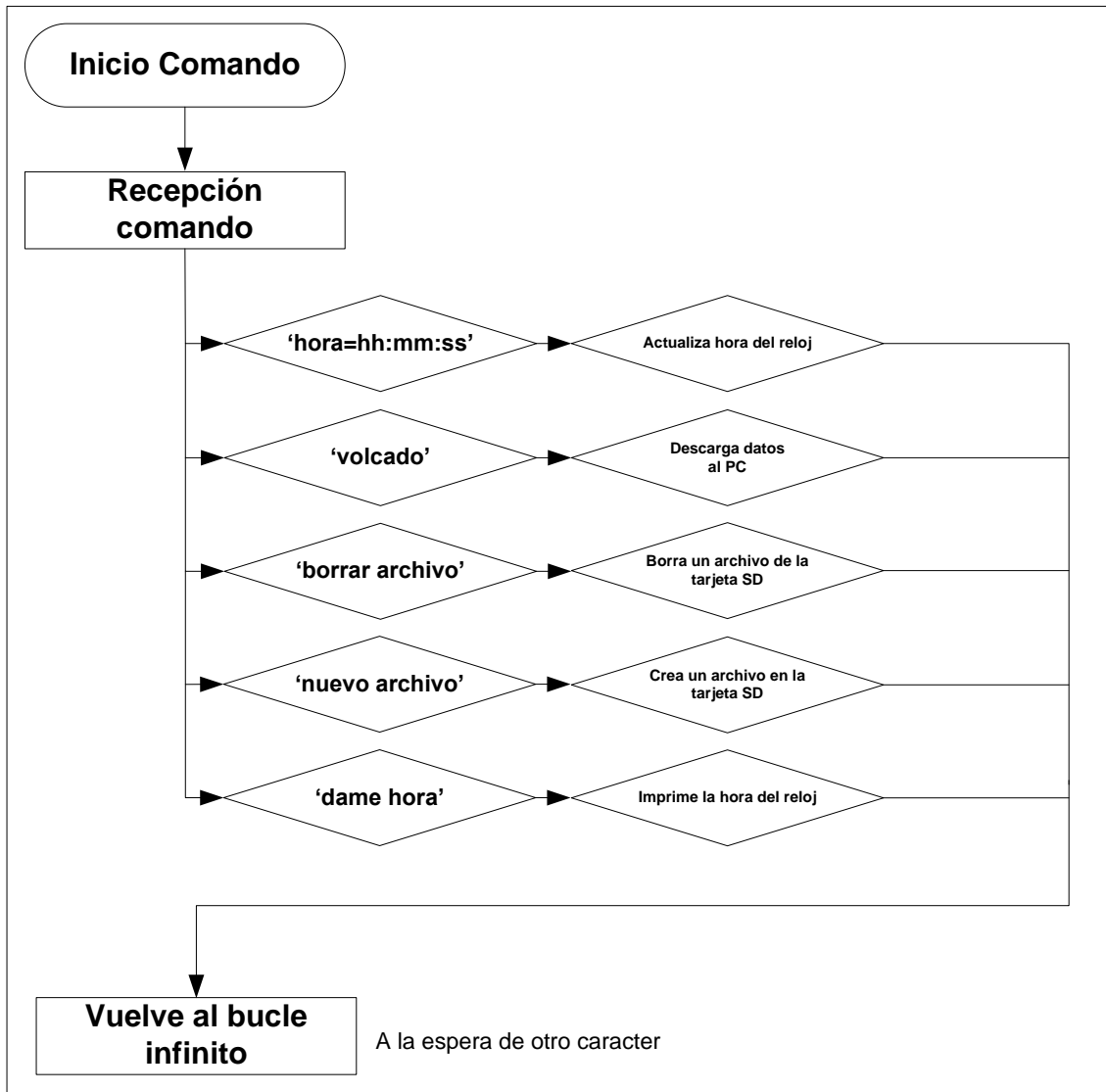


Figura 8.10. Diagrama comandos

9. ANEXO III: MANUAL DEL USUARIO

En este capítulo se detallan los pasos que debe seguir el usuario para poner en marcha el prototipo.

9.1. Ubicación de los componentes

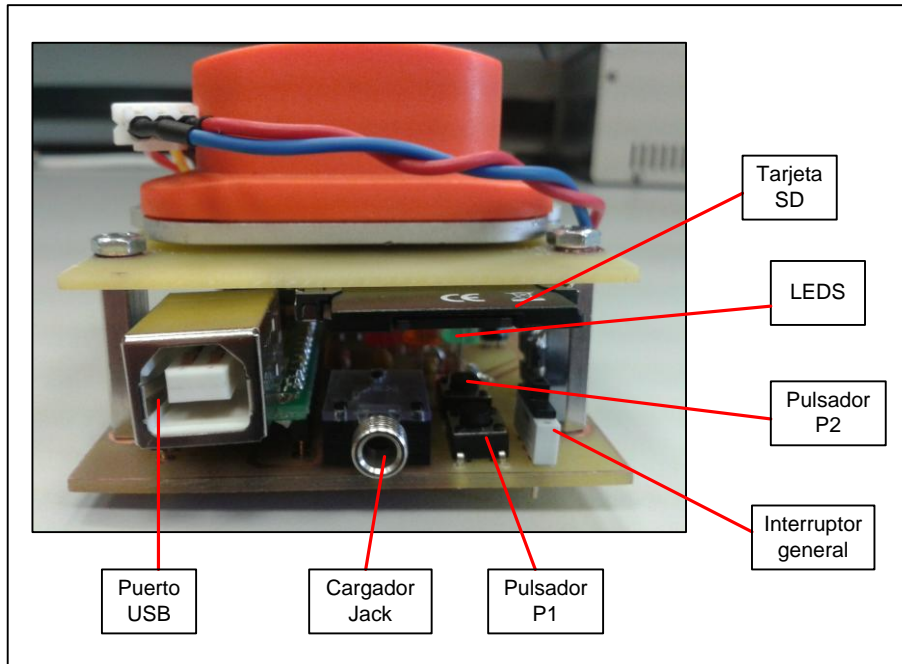


Figura 9.1. Distribución de componentes

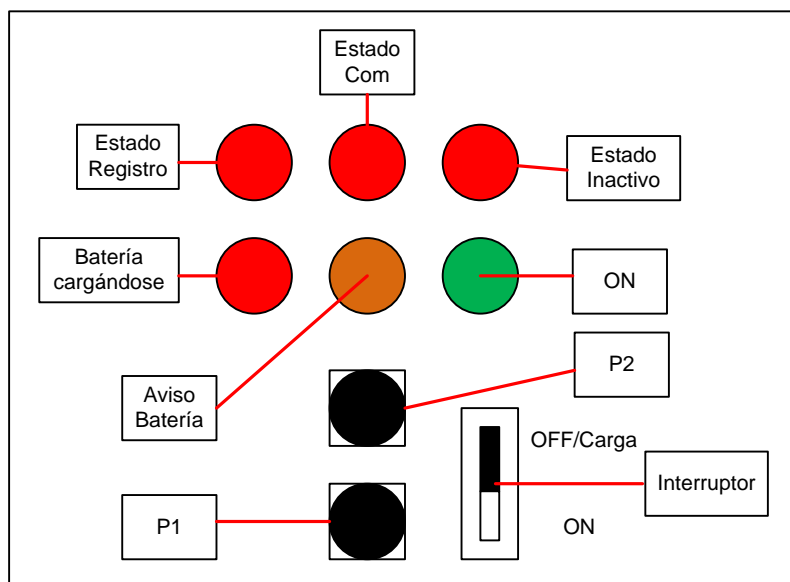


Figura 9.2. Esquema de componentes

9.2. Puesta en marcha

1. Retirar la tapa del soporte abriendo los ganchos laterales.

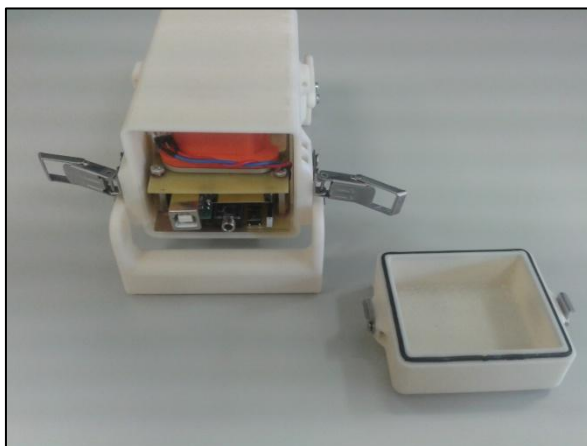


Figura 9.3. Paso 1 puesta en marcha

2. Introducir la tarjeta SD.

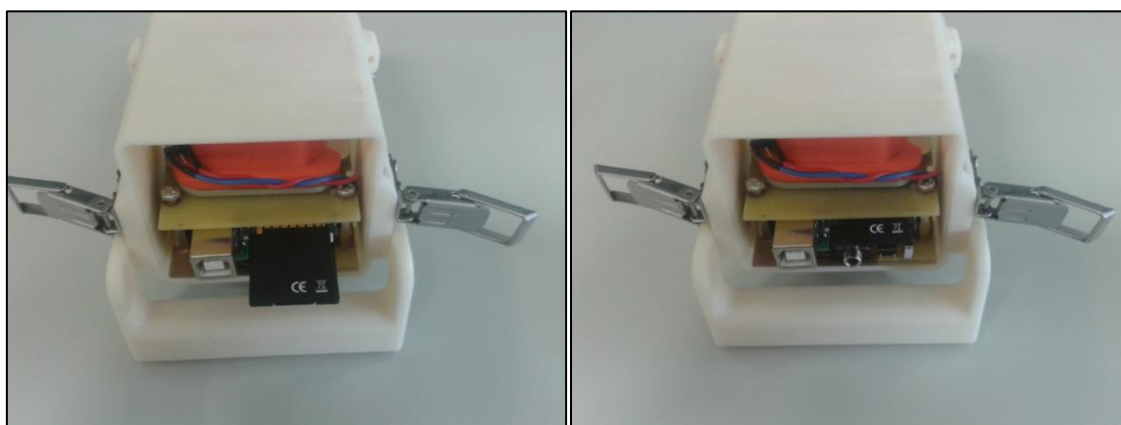


Figura 9.4. Paso 2 puesta en marcha

3. [Opcional] Si se desea configurar el sistema a través de un PC conectar el cable USB al puerto serie.

4. Mover el interruptor a la posición ON, tras unos segundos se encenderán los LEDS 'ON' e 'Inactivo'.

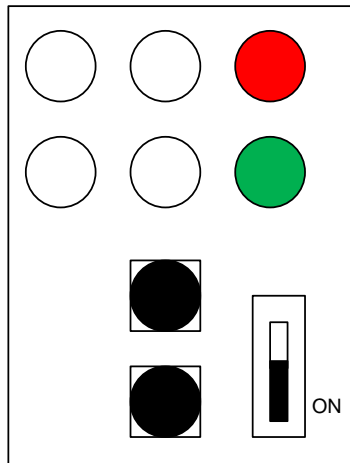


Figura 9.5. Paso 4 puesta en marcha

5. Manipular el sistema con los pulsadores P1 y P2 para cambiar al modo comunicación y registro.

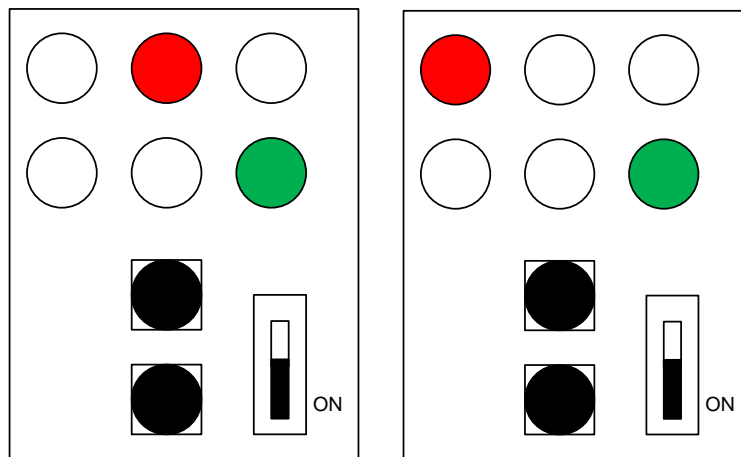


Figura 9.6. Paso 5 puesta en marcha

6. El modo comunicación solo es necesario si se llevó a cabo el paso 3.
7. Una vez en el modo registro cerrar el soporte con la tapa, esperar 60s y comenzar el ciclo de trabajo.

9.3. Cargar batería

1. El LED de aviso de batería se enciende cuando ésta se encuentra al 22,25% de su capacidad máxima.

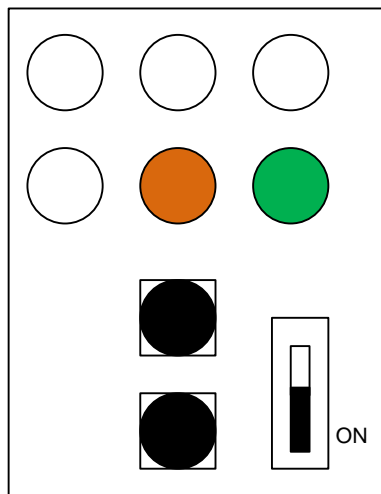


Figura 9.7. Paso 1 carga batería

2. Mover el interruptor a la posición OFF/Carga.
3. Conectar el adaptador de tensión configurado a 6v.



Figura 9.8. Paso 3 carga de batería

4. El LED de carga de batería se encenderá hasta que la batería se haya cargado (12 horas aprox).

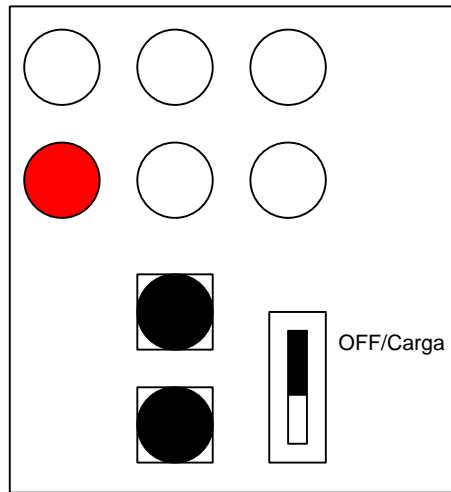


Figura 9.9. Paso 4 carga de batería

9.4. Comunicación con PC

1. [Opcional] Instalar los drivers del puerto UM232R USB. Descargar aquí <http://www.ftdichip.com/FTDrivers.htm>
2. Conectar el cable USB al PC.
3. Ejecutar el Hyperterminal de WINDOWS en modo 'administrador'.
4. Dar nombre a la sesión y aceptar.

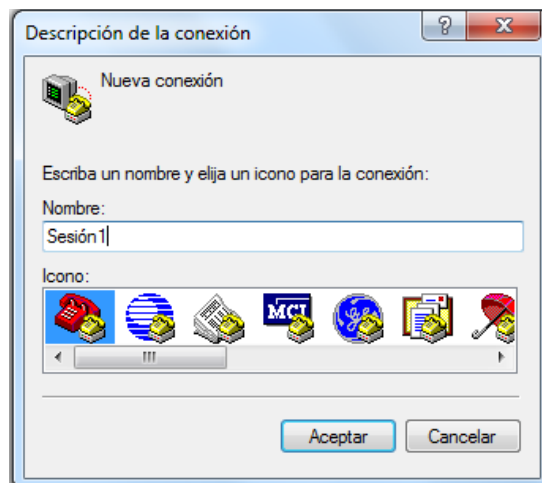


Figura 9.10. Paso 4 comunicación PC

5. Elegir el puerto donde conectamos el cable USB.

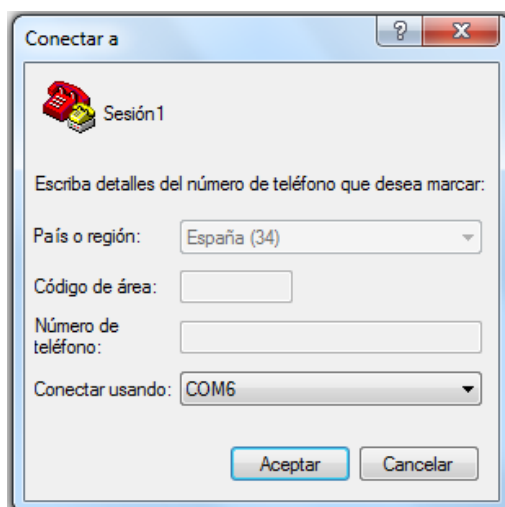


Figura 9.11. Paso 5 comunicación PC

6. Configurar el tipo de comunicación

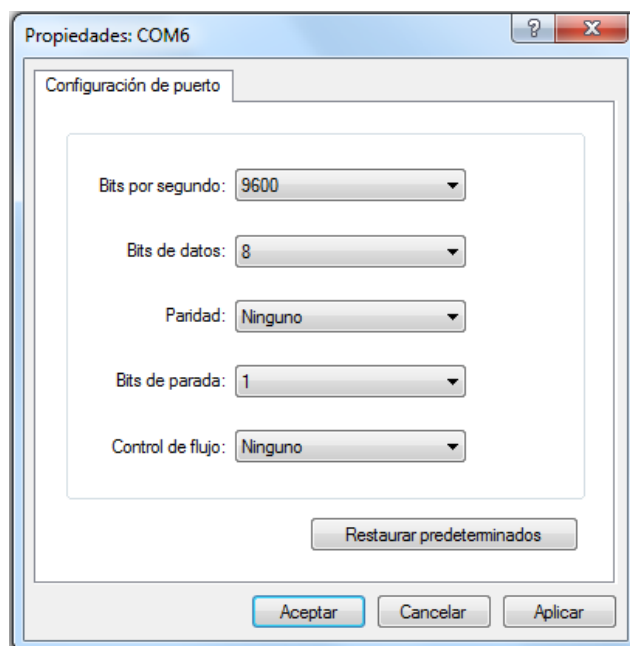


Figura 9.12. Paso 6 comunicación PC

7. Encender el prototipo y pasar al modo comunicación para comenzar la configuración desde el PC.

9.5. Pantalla de explotación ISA

1. Una vez concluido el ciclo de trabajo extraer la tarjeta SD y almacenar el archivo log.txt en la carpeta tubecheck/res.

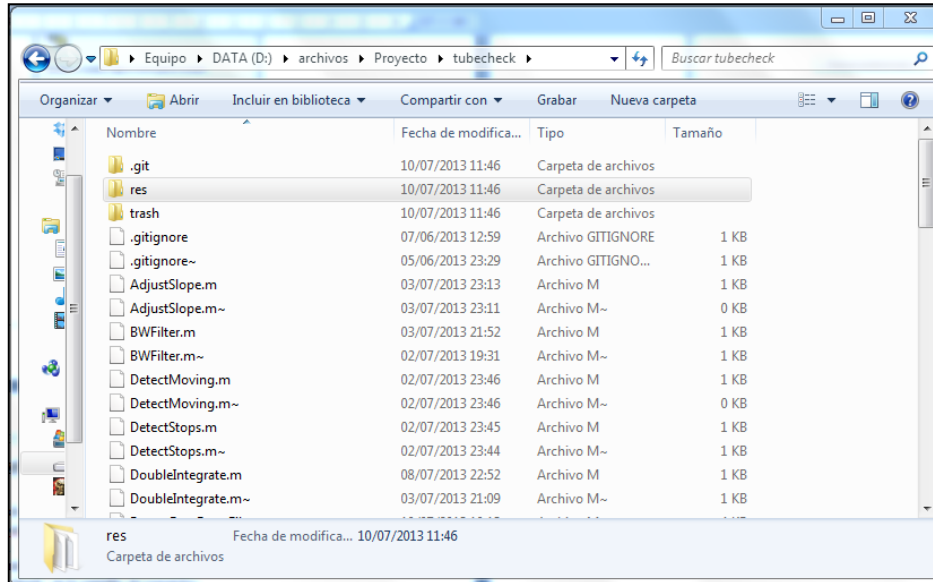


Figura 9.13. Paso 1 pantalla de explotación ISA

2. Ejecutar Matlab.
3. En el buscador introducir la dirección de la carpeta tubecheck.

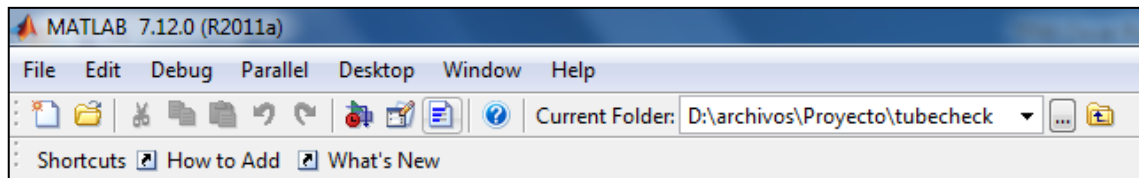


Figura 9.14. Paso 2 pantalla de explotación ISA

4. Introducir 'Tubecheck' en la pantalla de comandos.
5. Pulsar el botón 'Leer datos de archivo', la opción 'Configurar equipo no está habilitada'.



Figura 9.15. Paso 5 pantalla de explotación ISA

6. Buscar y seleccionar el archivo log.txt.
7. Seleccionar el tipo de sensor (tarjeta o IMU) y el sensor deseado (acelerómetro, giróscopo o magnetómetro).

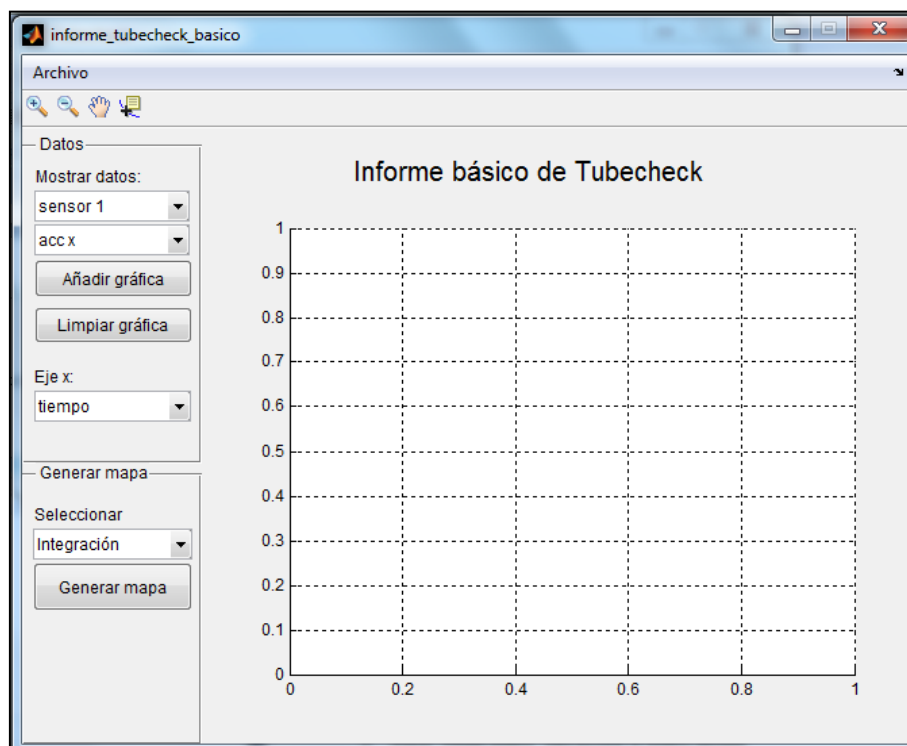


Figura 9.16. Paso 7 pantalla de explotación ISA

8. Seleccionar en 'EjeX' si se desea representar tiempo o número de muestras.
9. Pulsar 'Añadir gráfica' para mostrar la gráfica deseada

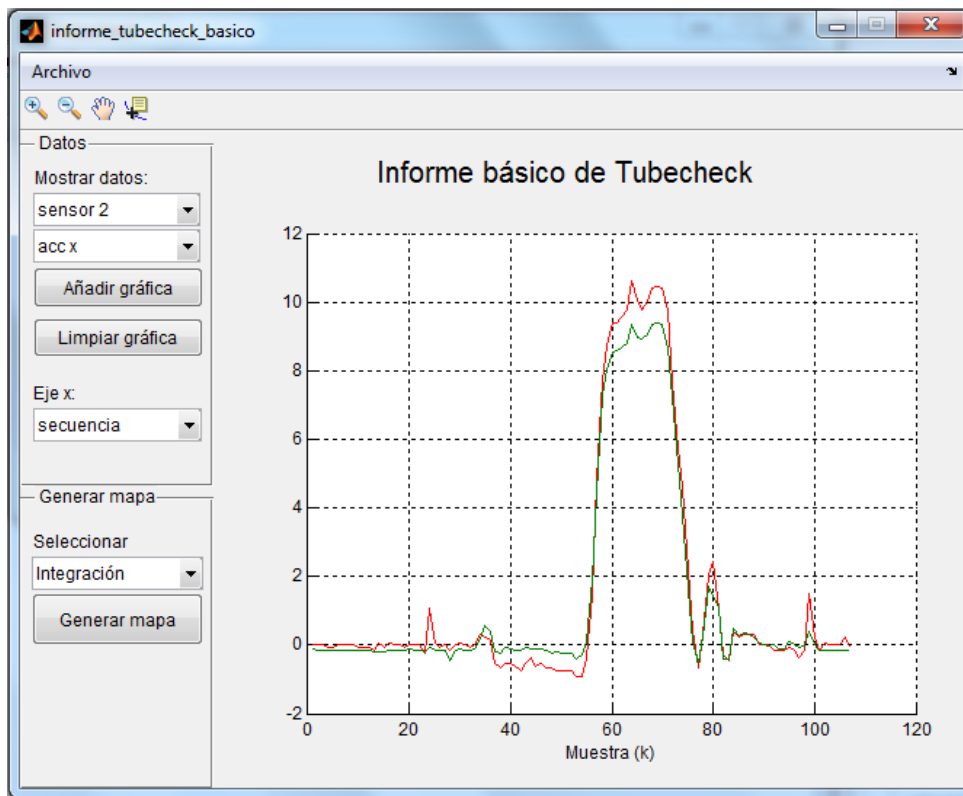


Figura 9.17. Paso 9 pantalla de explotación ISA

10. Pulsar 'Limpiar gráfica' para borrar todas las gráficas visibles.
11. La opción 'generar' mapa no está habilitada aún.

10. ANEXO IV: ENSAYOS Y PRUEBAS

En este capítulo se detallan las pruebas a las que ha sido sometido el prototipo para conocer el grado de satisfacción con el que cumple los requisitos.

10.1. Prueba de estanqueidad

Se pretende comprobar la estanqueidad del prototipo. Para ello se sumergirá totalmente en agua durante 60s para comprobar su efectividad.

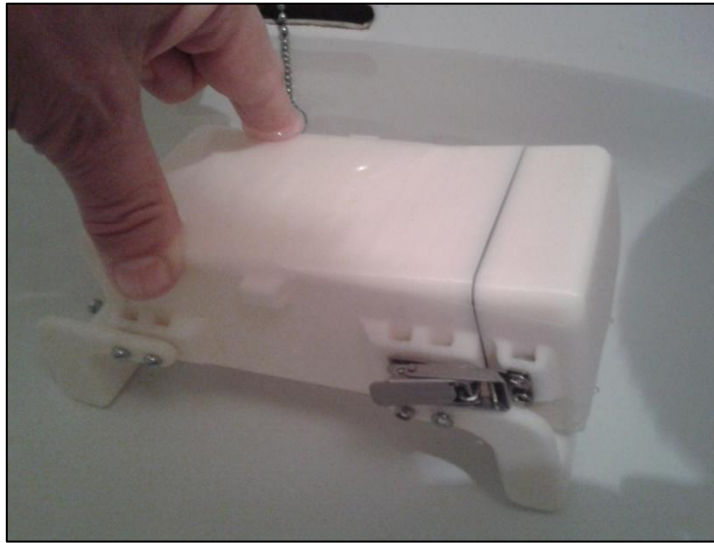


Figura 10.1. Prueba de estanqueidad

Tras finalizar la prueba se observa que el interior de la caja permanece seco por lo que se deduce que el sistema es estanco.

10.2. Error entre los sensores inerciales y el IMU

Se realiza un estudio para comprobar la similitud entre los datos de las dos fuentes de información; los sensores inerciales y el IMU.

Para ello se realiza un muestreo de datos de 3 horas de duración de donde se seleccionan 100 muestras seguidas para realizar el estudio.

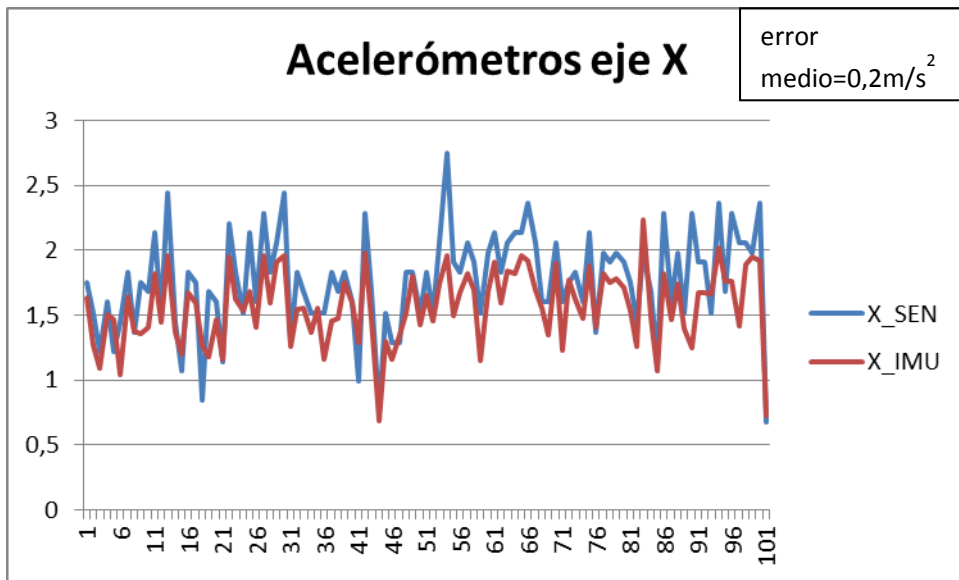


Figura 10.2. Diagrama de error entre acelerómetros eje X

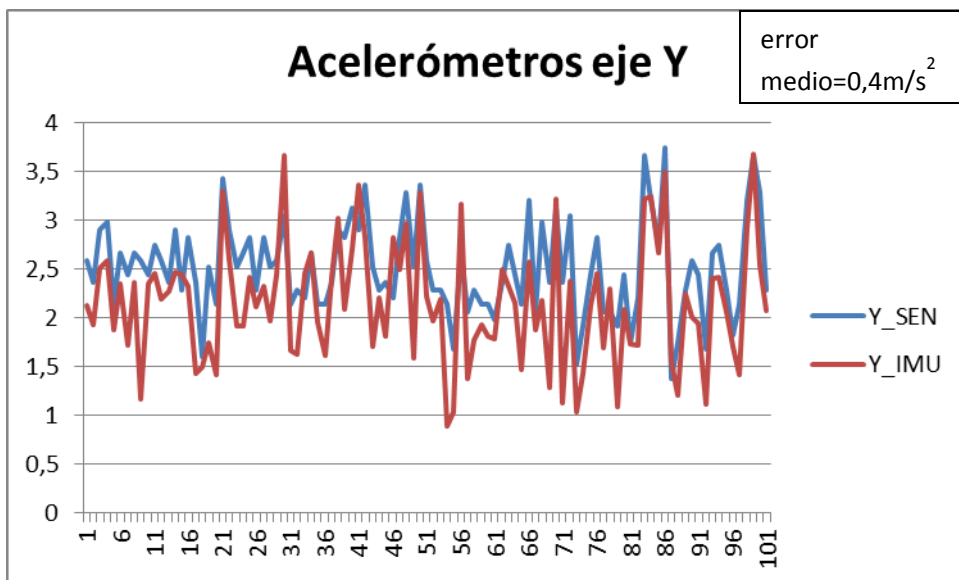


Figura 10.3. Diagrama de error entre acelerómetros eje Y

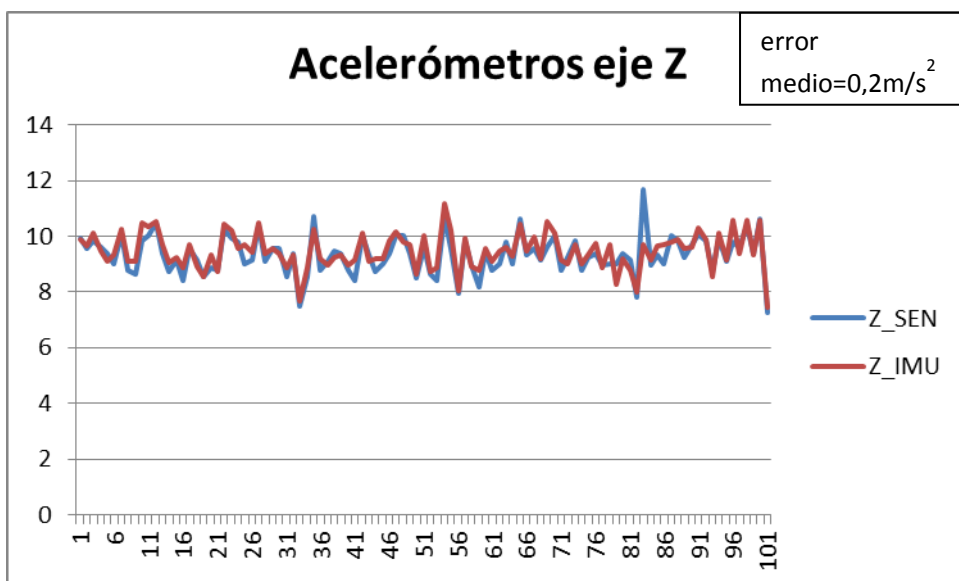


Figura 10.3. Diagrama de error entre acelerómetros eje Z

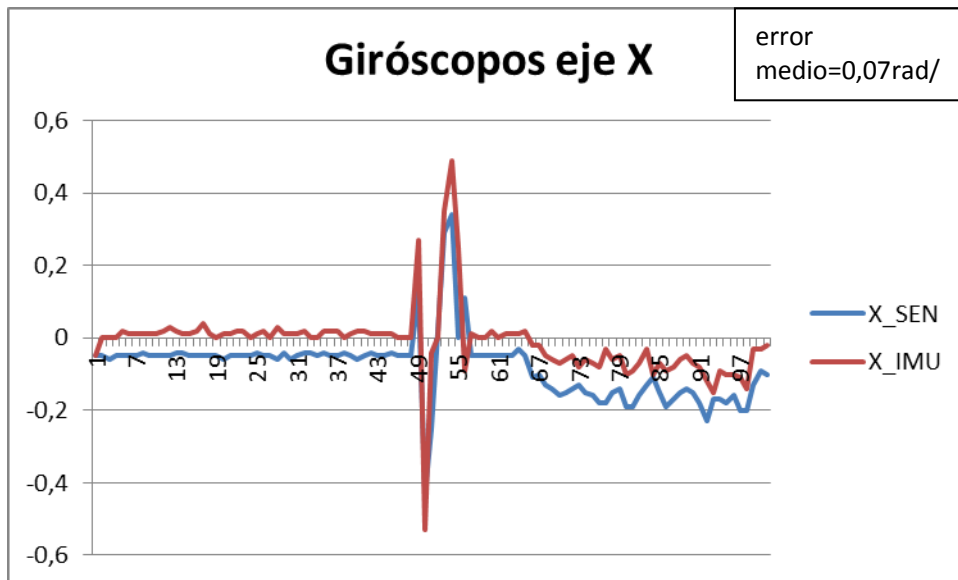


Figura 10.4. Diagrama de error entre giróscopos eje X

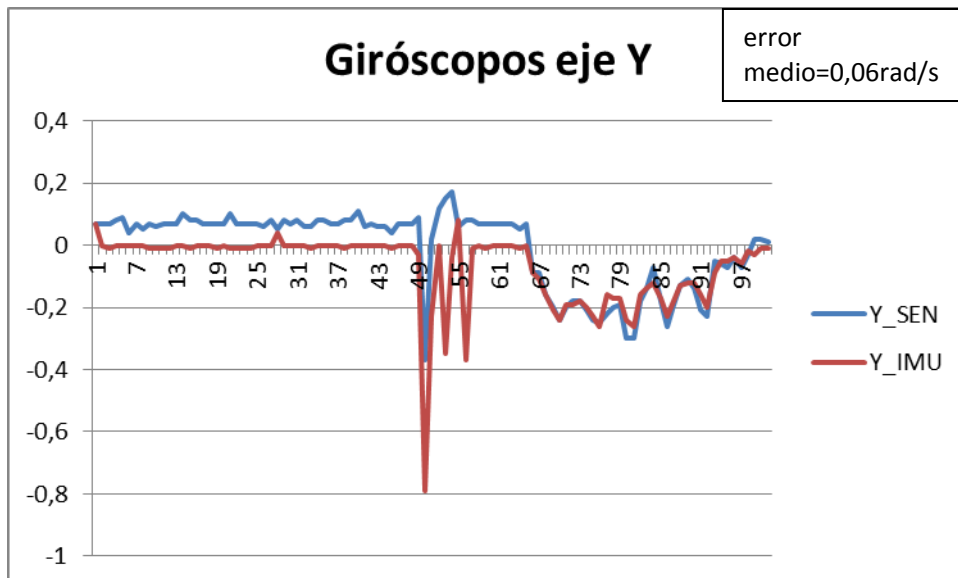


Figura 10.5. Diagrama de error entre giróscopos eje Y

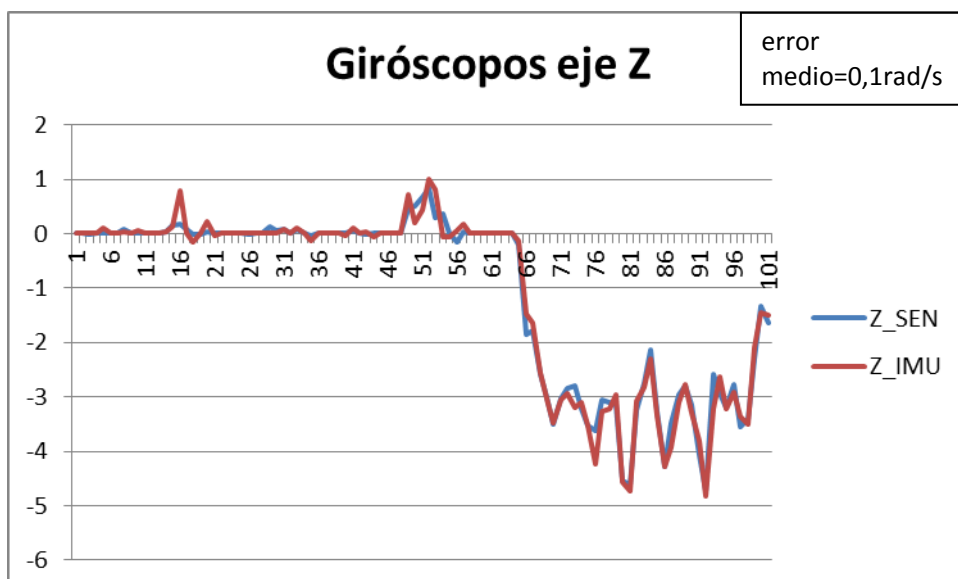


Figura 10.6. Diagrama de error entre giróscopos eje Z

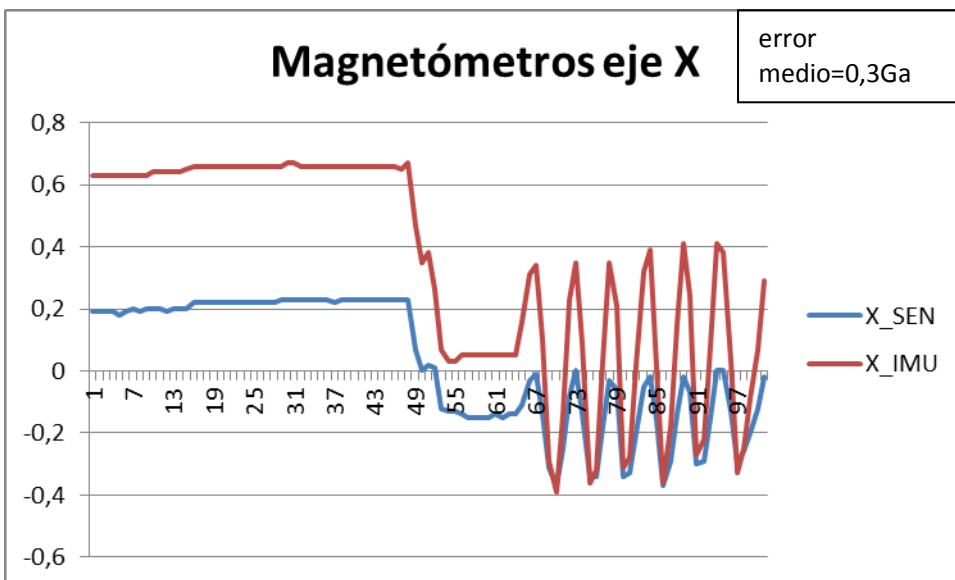


Figura 10.7. Diagrama de error entre magnetómetros eje X

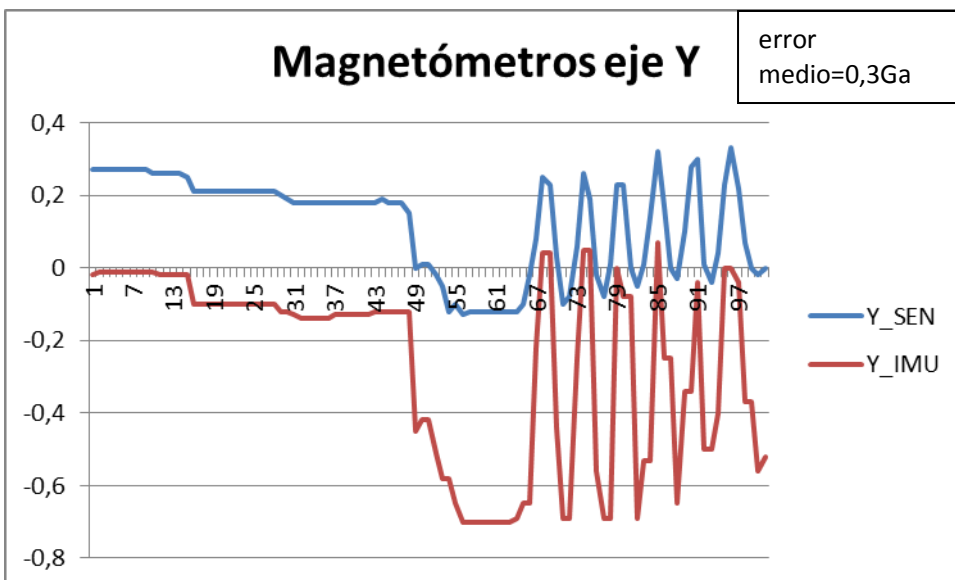


Figura 10.8. Diagrama de error entre magnetómetros eje Y

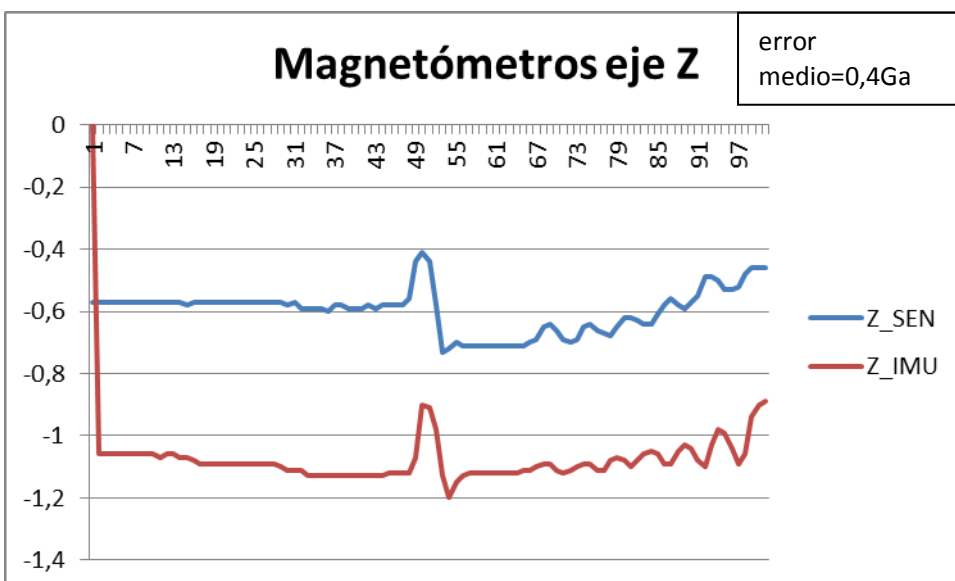


Figura 10.9. Diagrama de error entre magnetómetros eje Z

Observando las gráficas podemos apreciar un error entre ellas que en ocasiones parece bastante notable.

En el caso de los acelerómetros y giróscopos este error puede ser debido entre otras cosas a la distancia física que existe entre ambos componentes y a que los datos del IMU están calibrados, dando lugar a unos datos más fiable.

El caso de los magnetómetros es diferente, comparar ambos es difícil puesto que estos dispositivos se ven afectados en gran medida por su entorno, hasta el punto de que dos magnetómetros contiguos pueden dar lecturas diferentes.

10.3. Trazado de una trayectoria a partir de datos muestreados

Aunque la interpretación de los datos muestreados no es del alcance del presente proyecto, es necesario comprobar que es posible trazar una trayectoria a partir de los datos obtenidos por el prototipo.

Para ello se realiza una prueba en colaboración con el proyecto que se encargará de interpretar los datos obtenidos (Área de Ingeniería de Sistemas y Automática). Se nos facilita un prototipo de pantalla de explotación con el que se puede observar las gráficas de los datos y realizar un mapeo de la trayectoria.

Se debe tener en cuenta que el mapeo se realiza en base a un algoritmo diseñado específicamente para la trayectoria de la prueba, siendo imposible la reconstrucción de la trayectoria de otra prueba diferente. Además para simplificar el algoritmo se realiza una prueba que solo requiere de los acelerómetros para reconstruir la trayectoria.

Las limitaciones en la prueba se deben a que la reconstrucción de trayectorias a partir de datos inerciales supone un proyecto en sí mismo que todavía está en desarrollo, por lo que no es posible alcanzar todas sus funcionalidades aún.

La prueba consistirá en dos movimientos en línea recta, uno en X y otro en Y como se puede ver en la imagen. El movimiento en Y se realiza en el sentido positivo de eje de valor 16cm mientras que el movimiento en X se realiza en el sentido negativo de eje de valor 14cm.

En la gráfica resultante se puede observar como se ha podido reproducir la forma de la trayectoria; sin embargo las cotas, aunque del mismo orden, no coinciden con la reales. Esta deriva era previsible debido a la acumulación error al integrar los datos.

Un algoritmo más preciso y un aumento de la frecuencia de muestreo podrían minimizar este error haciendo que la trayectoria estimada se asemeje más a la trayectoria real.

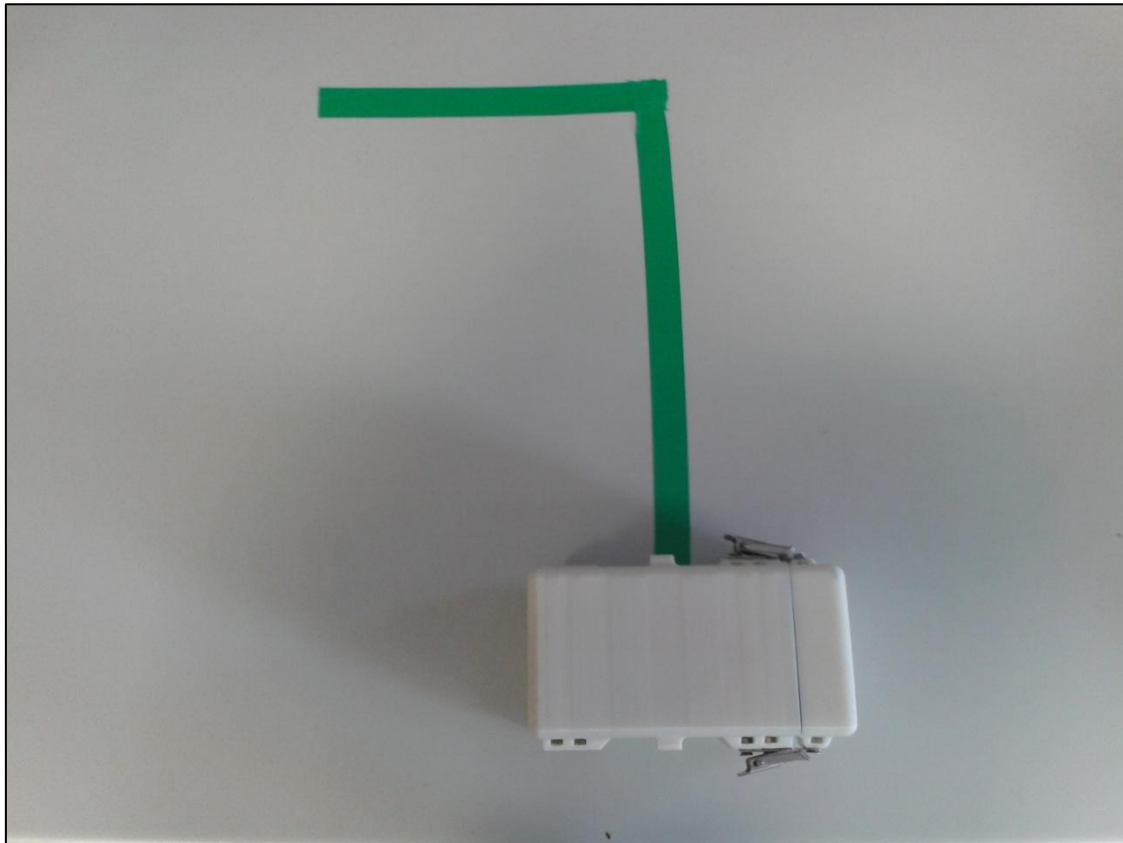


Figura 10.10. Trayectoria real de la prueba

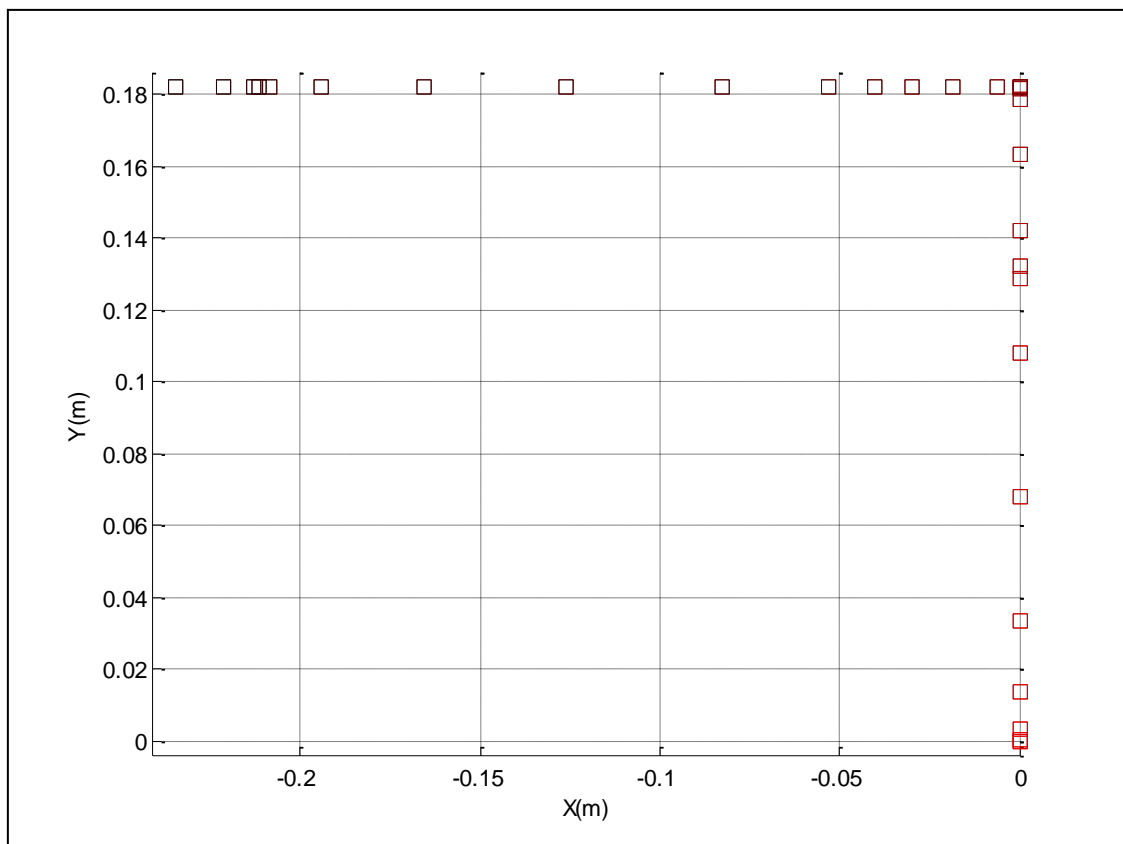


Figura 10.11. Trayectoria estimada de la prueba

11. CONCLUSIONES

- Se ha diseñado y fabricado un prototipo que cumple los requisitos mínimos detallados al inicio del documento.
- El prototipo es capaz de almacenar 5 horas de datos partiendo de una situación en la que la batería está totalmente cargada (4,2V).
- El diseño de la caja y el sistema de sujeción permite una rápida puesta en marcha del sistema.
- El sistema de sujeción al robot puede sustituirse para adaptar la caja a otra estructura.
- El prototipo ha sido puesto a prueba con una serie de ensayos con resultado satisfactorio.
- No ha sido posible probar el sistema de sujeción de la carcasa al robot explorador por falta de disponibilidad del mismo.

A continuación se presentan los puntos donde es posible mejorar el prototipo, dejándolo como tareas pendientes en futuras ampliaciones del proyecto.

- Conseguir un aumento de la frecuencia de muestreo para poder usar el prototipo en otras aplicaciones y realizar mapeados más precisos.
- Rediseñar el sistema de carcas de protección para su fabricación en taller.
- Minimizar el prototipo.
- Mejorar la accesibilidad de los botones e interruptores del sistema electrónico.
- Aumentar el número de funcionalidades en el interfaz de usuario.

12. BIBLIOGRAFÍA

- **Artículos consultados**

[1] *SD card simplified physical layer spec. Panasonic. 2006.*

[2] *SanDisk Secure Digital Card Manual. Sandisk. 2003.*

[3] *Carlos Alberto Heano. How to use a SD/MMC memory card with PIC. Universidad Tecnológica de Pereira. 2010.*

- **Páginas WEB y catálogos ON-LINE consultados**

[1] *RS ESPAÑA. Suministros industriales. Sitio Web, <http://es.rs-online.com>*

[2] *Farnell ESPAÑA. Suministros industriales. Sitio Web, <http://es.farnell.com>*

[3] *Sparkfun. Suministros electrónicos. Sitio Web, <https://www.sparkfun.com>*

[4] *Ucontrol. Foro de electrónica. Sitio Web, <http://www.ucontrol.com>*

[5] *CCS foro. Foro del compilador CCS. Sitio Web, <http://www.ccsinfo.com>*

[6] *Micros-designs. Proyectos y tutoriales electrónica/programación. Sitio Web, <http://www.micros-designs.com>*

[7] *Blog de inventos. Proyectos y tutoriales electrónica/programación. Sitio Web, <http://www.blog.deinventos.com>*

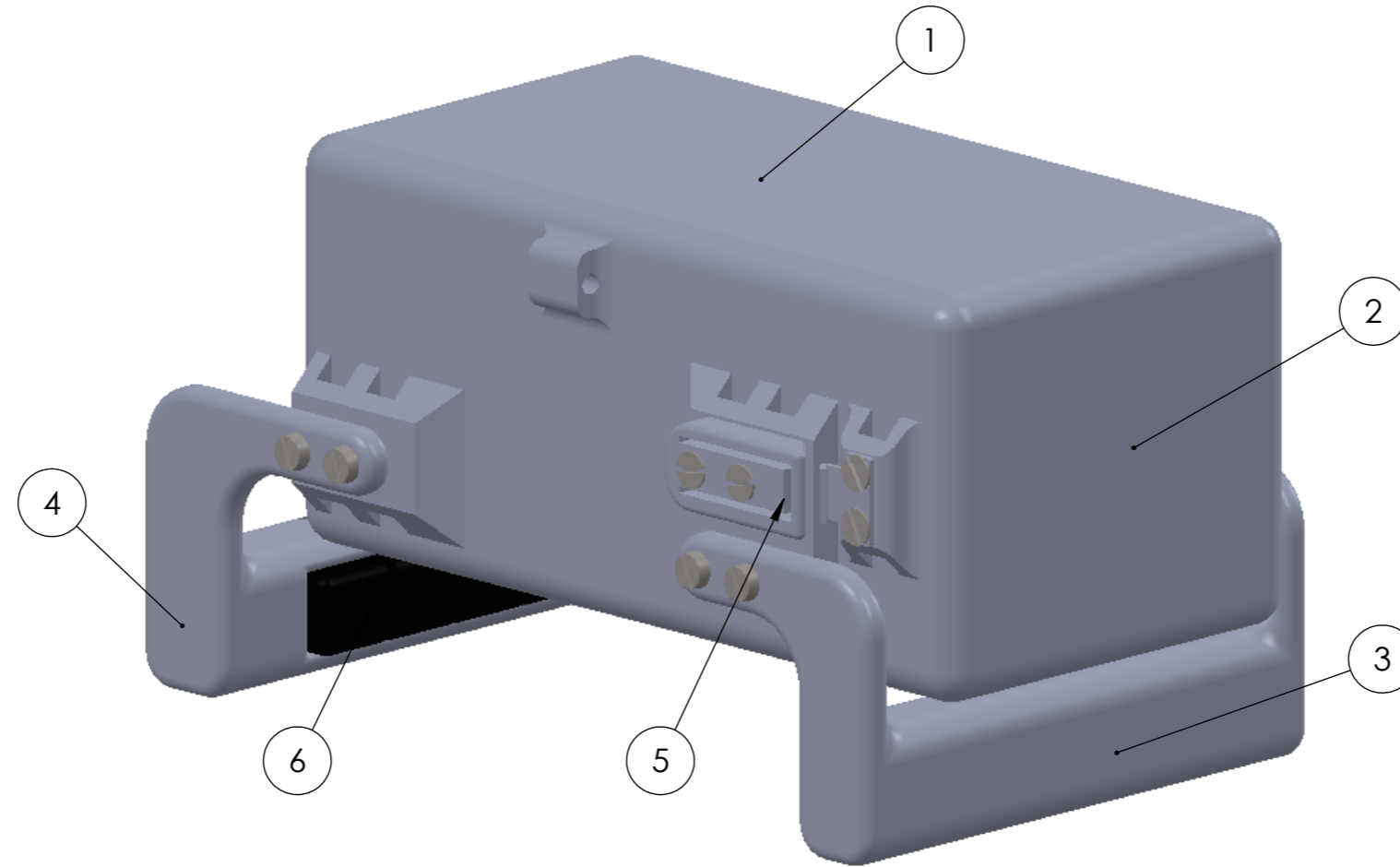
[8] *Ladelec. Proyectos y tutoriales electrónica/programación. Sitio Web, <http://www.ladelec.com>*

[9] *Jordi Bartolomé. Las tarjetas MultiMedia Card y SD.2004. Sitio Web <http://www.tolaemon.com>*

13. PLANOS

13.1. Índice de planos

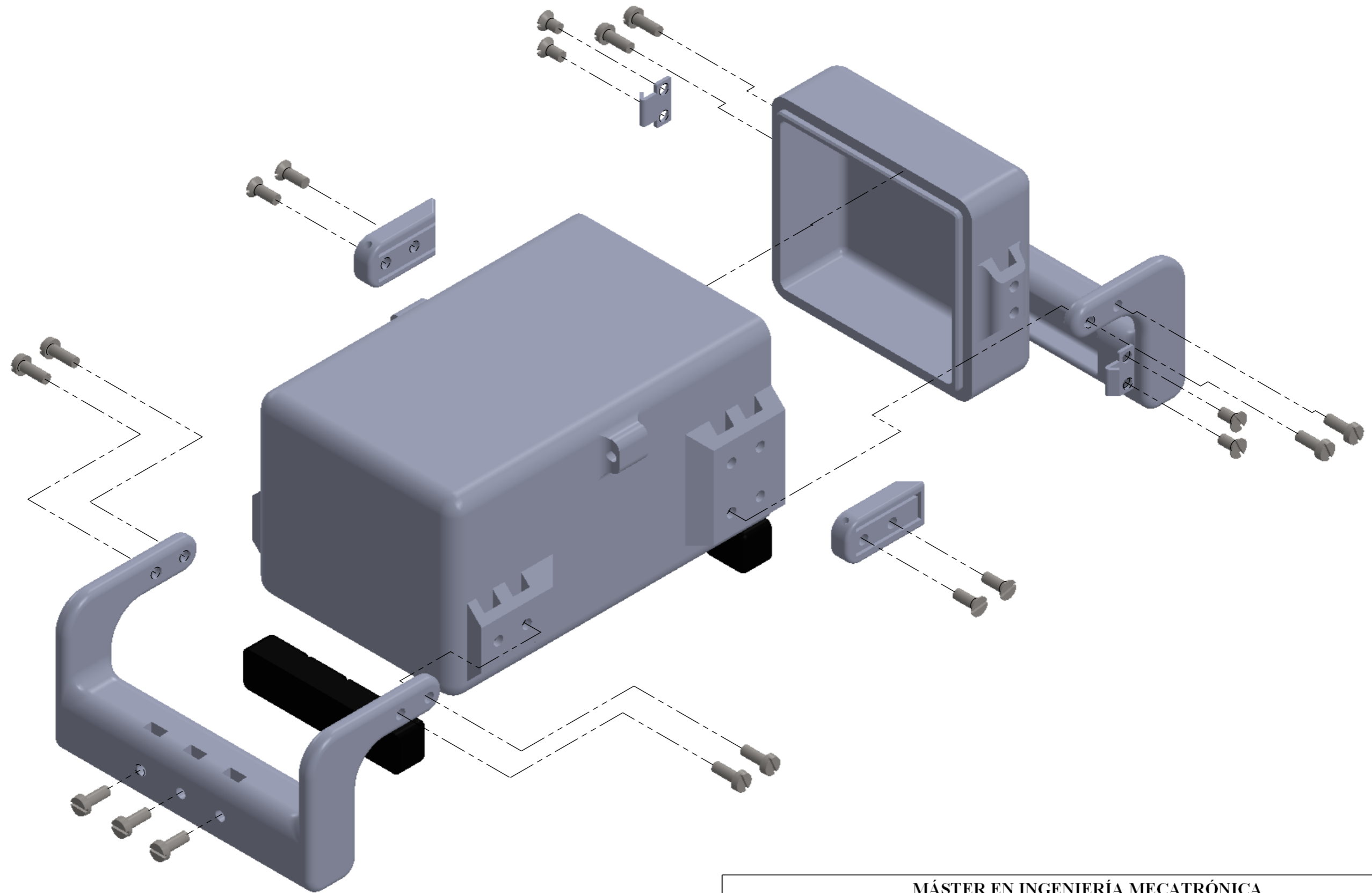
1. Plano general
2. Explosión de ensamble
3. Caja
4. Tapa
5. Pieza de sujeción
6. Pieza de sujeción con tornillos
7. Fotelito cara TOP de la tarjeta superior
8. Fotelito cara BOTTOM de la tarjeta superior
9. Fotelito cara TOP de la tarjeta inferior
10. Fotelito cara BOTTOM de la tarjeta inferior
11. Planos de componentes cara TOP tarjeta superior
12. Planos de componentes cara BOTTOM tarjeta superior
13. Planos de componentes cara TOP tarjeta inferior
14. Planos de componentes cara BOTTOM tarjeta inferior
15. Plano esquemático
16. Lista de materiales



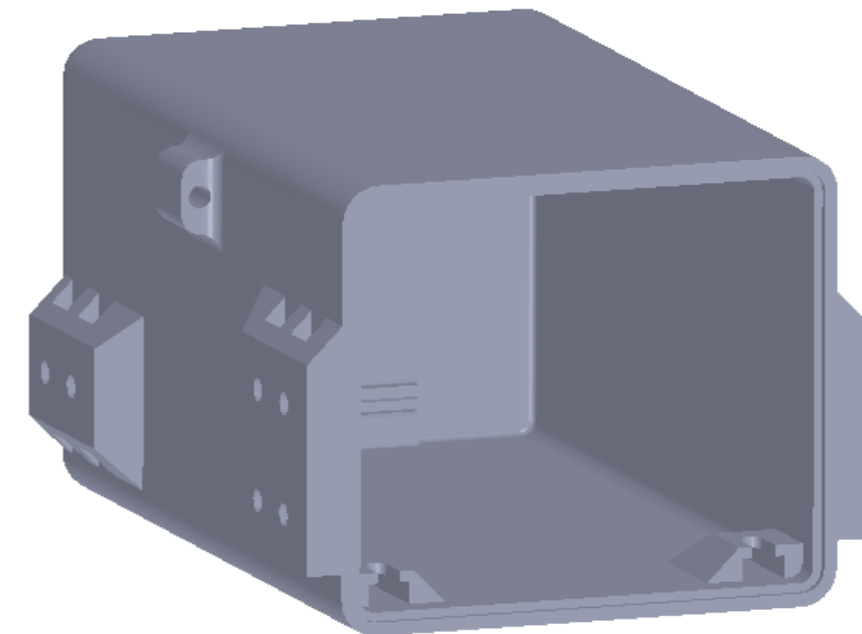
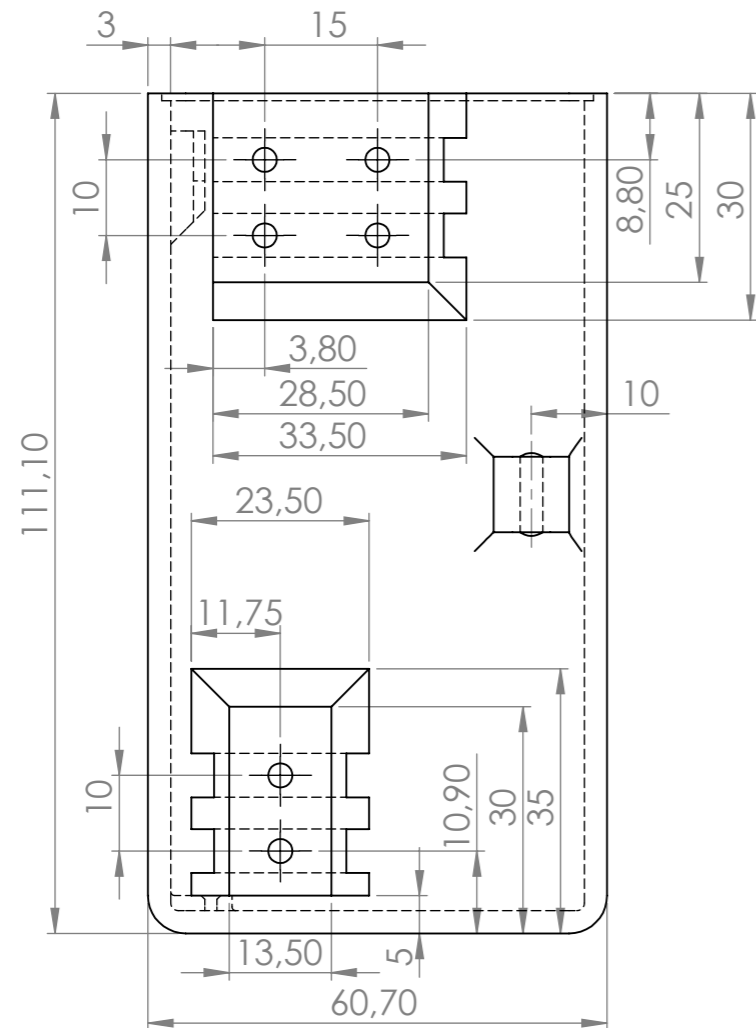
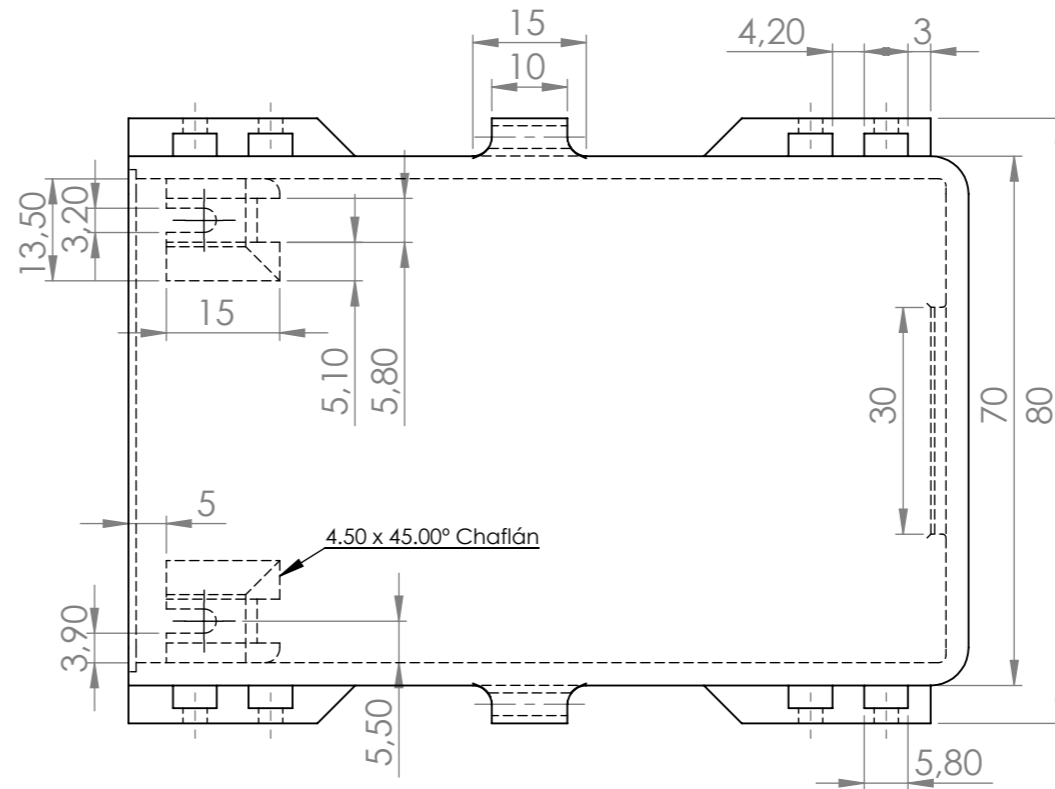
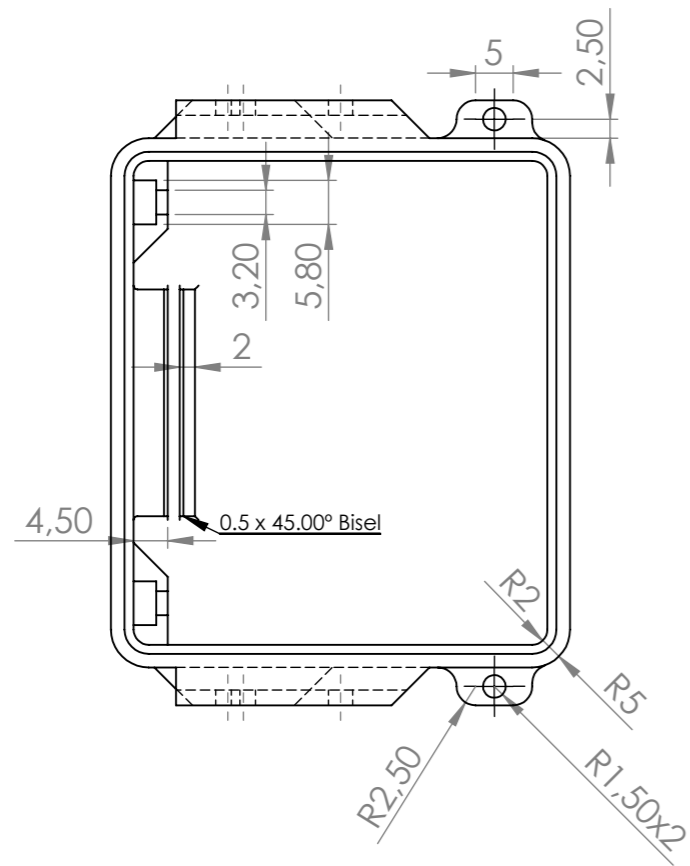
N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Caja	1
2	Tapa	1
3	Soporte	1
4	Soporte con tornillos	1
5	Cierre de palanca	2
6	Zapatatas	2

**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

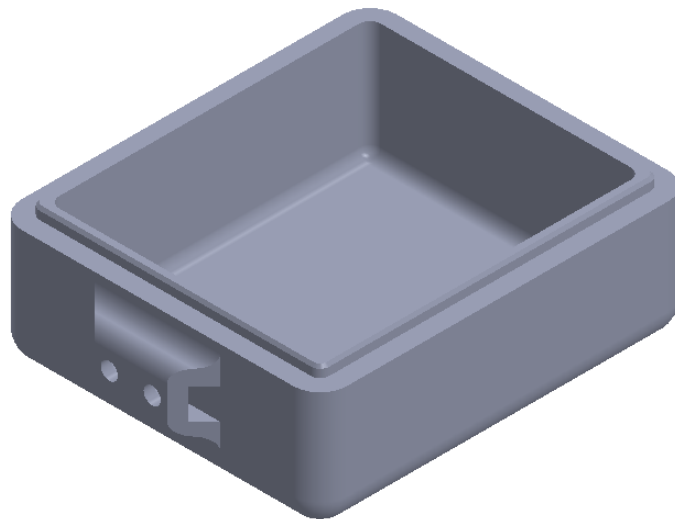
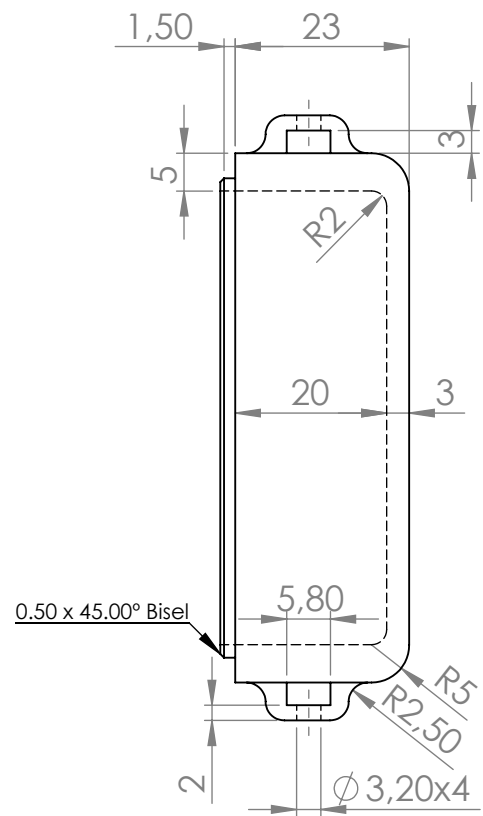
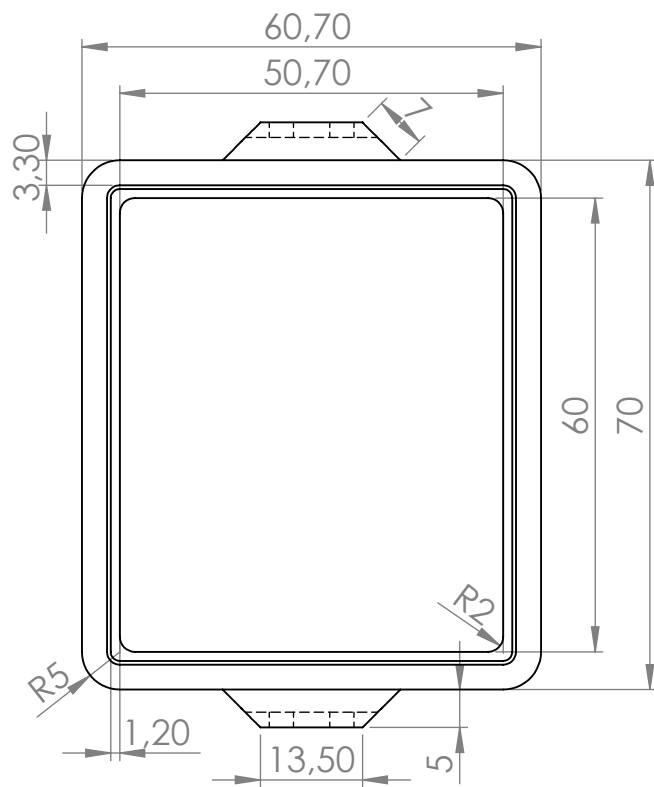
	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA				PLANO N° 1
1/1	Plano general			Sustituye a
				Sustituido por



MÁSTER EN INGENIERÍA MECATRÓNICA - TRABAJO FIN DE MÁSTER -			
	FECHA	NOMBRE	FIRMA
Proyectado	20/06/13	OSCAR RIVERA	
Dibujado	20/06/13	OSCAR RIVERA	
Comprobado	20/06/13	FERNANDO NUÑO	
ESCALA			PLANO N° 2
1/1	Explosión de ensamble		Sustituye a
			Sustituido por



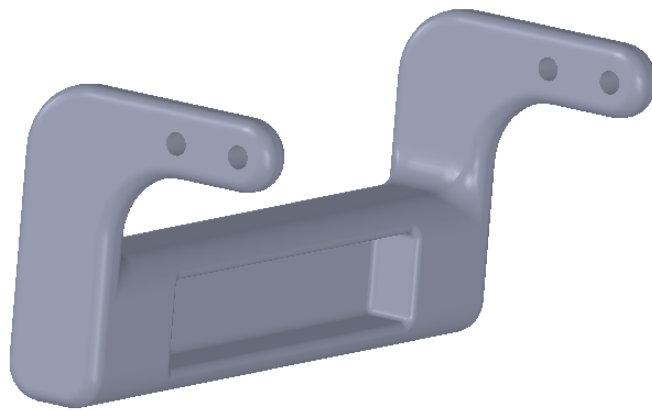
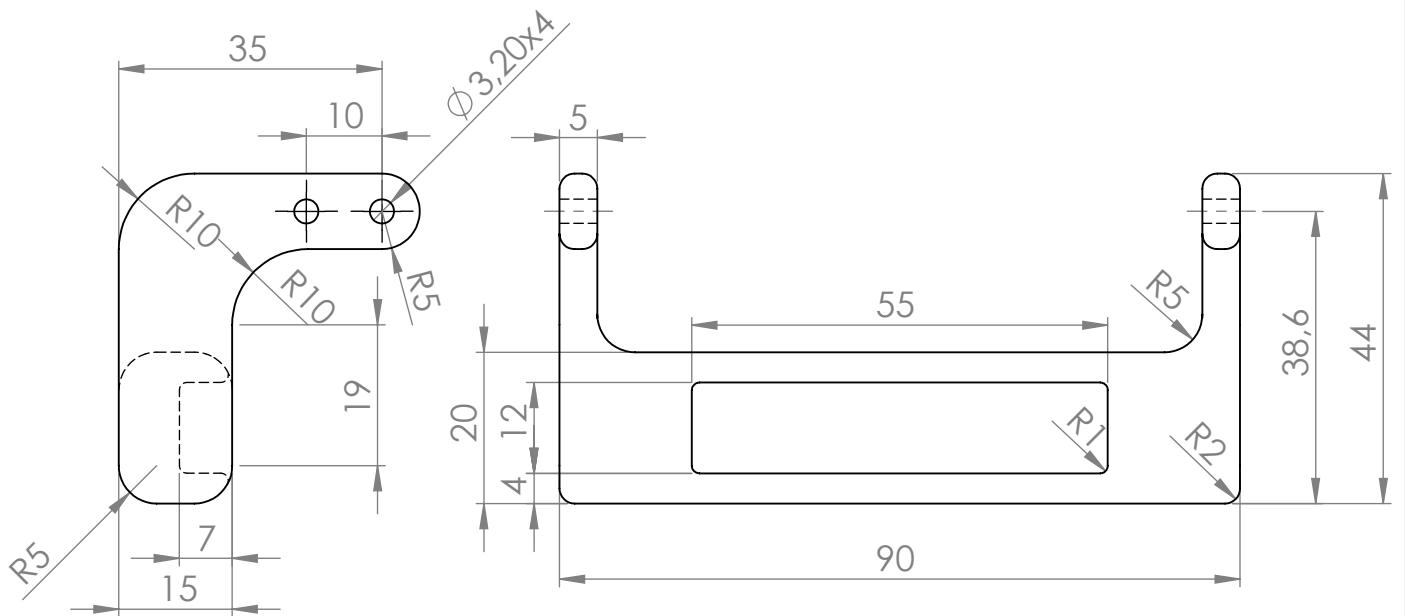
MÁSTER EN INGENIERÍA MECATRÓNICA - TRABAJO FIN DE MÁSTER -			
	FECHA	NOMBRE	FIRMA
Proyectado	20/06/13	OSCAR RIVERA	
Dibujado	20/06/13	OSCAR RIVERA	
Comprobado	20/06/13	FERNANDO NUÑO	
ESCALA	Caja		PLANO N° 3
1/1			Sustituye a
			Sustituido por



**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

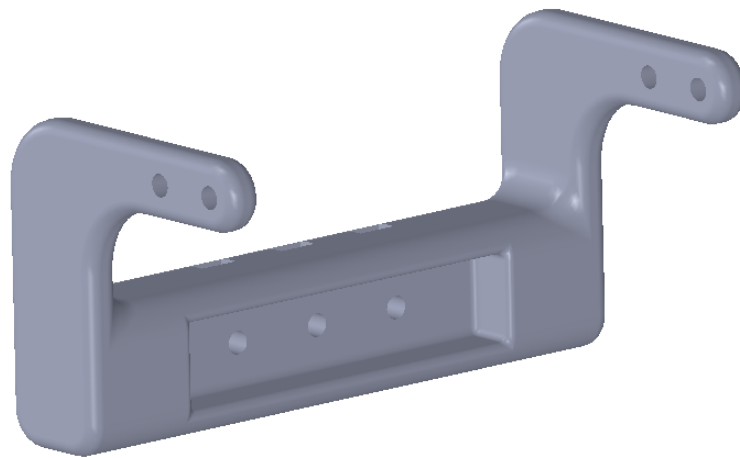
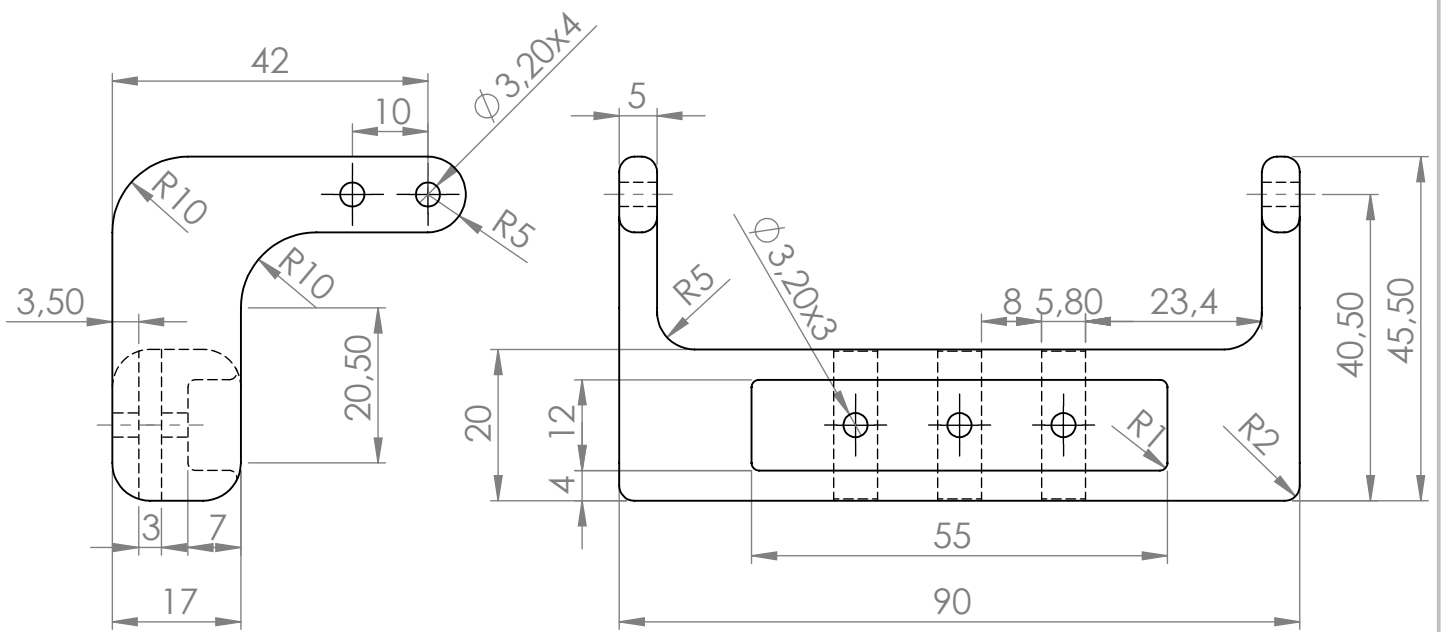
	FECHA	NOMBRE	FIRMA
Proyectado	20/06/13	OSCAR RIVERA	
Dibujado	20/06/13	OSCAR RIVERA	
Comprobado	20/06/13	FERNANDO NUÑO	

ESCALA	Tapa	PLANO N°	4
1/1		Sustituye a	
		Sustituido por	



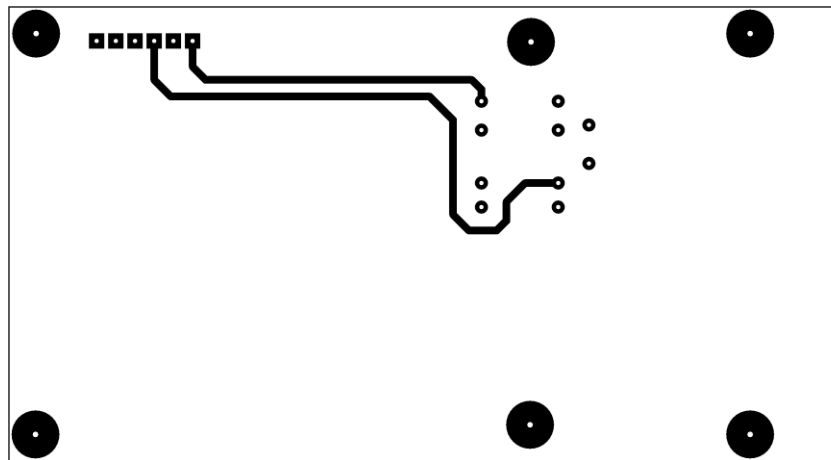
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA				PLANO N°
1/1	Pieza de sujeción			5
				Sustituye a
				Sustituido por



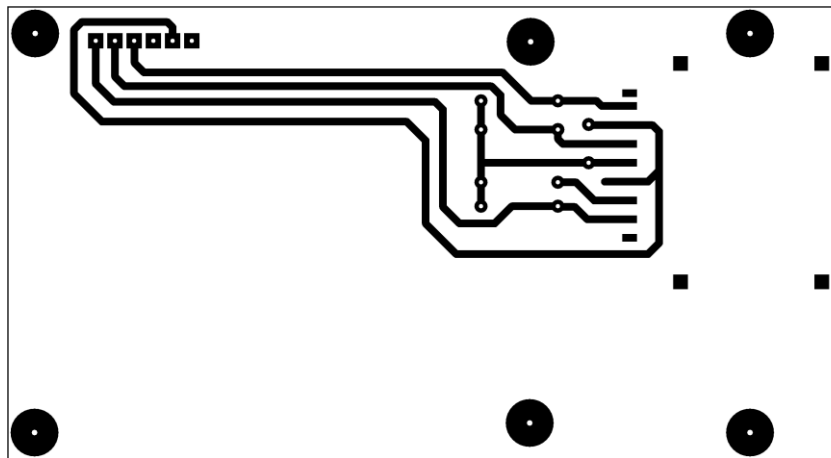
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA	Pieza de sujeción con tornillos			PLANO N° 6
1/1				Sustituye a
				Sustituido por



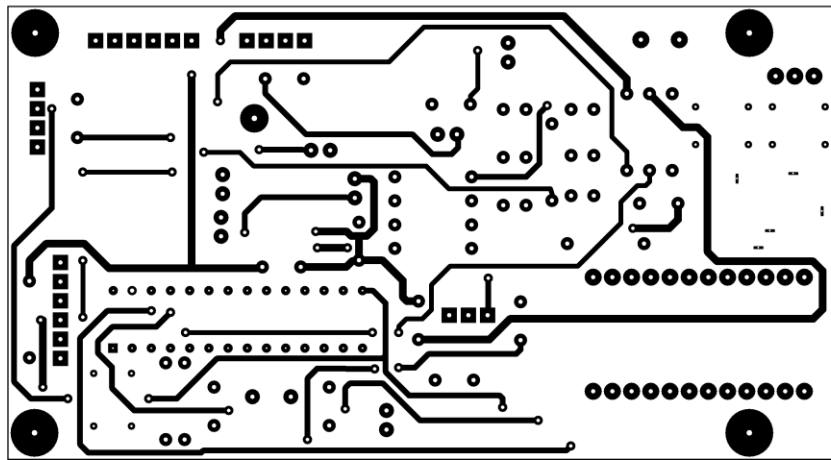
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA				PLANO N°
1/1	Fotolito cara TOP de la tarjeta superior			7
				Sustituye a
				Sustituido por



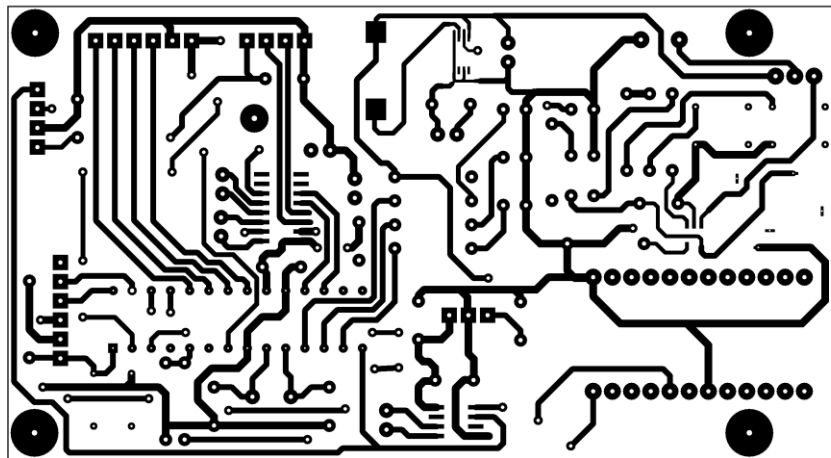
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA				PLANO N°
1/1	Fotolito cara BOTTOM de la tarjeta superior			8
				Sustituye a
				Sustituido por



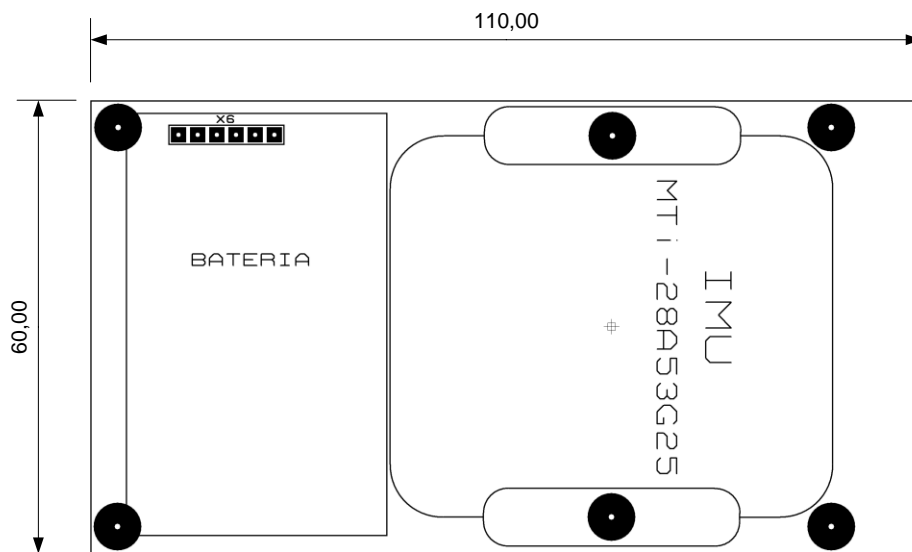
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA				PLANO N° 9
1/1	Fotolito cara TOP de la tarjeta inferior			Sustituye a
				Sustituido por



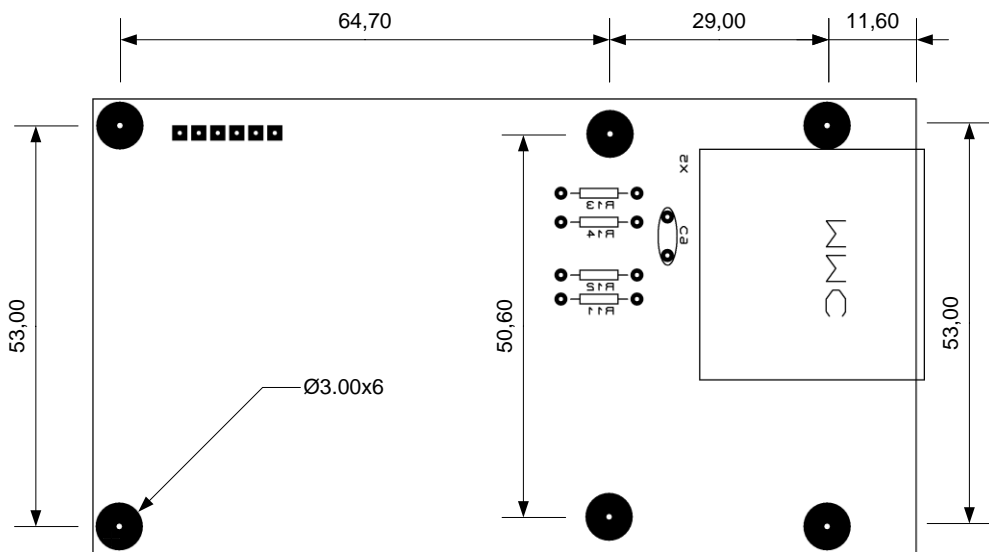
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA				PLANO N° 10
1/1	Fotolito cara BOTTOM de la tarjeta inferior			Sustituye a
				Sustituido por



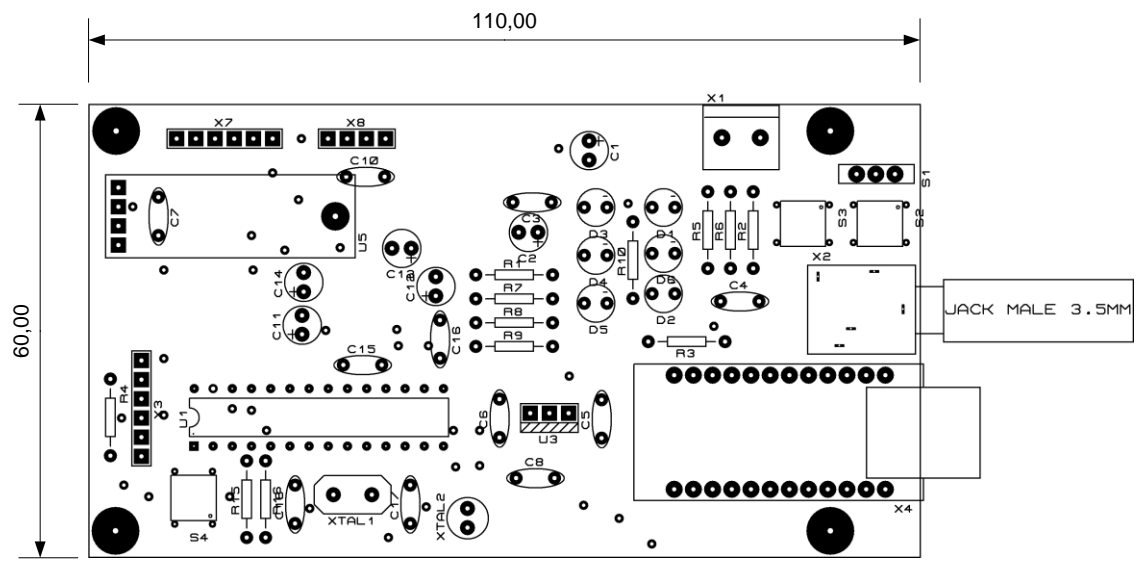
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA	Plano de componentes cara TOP placa superior			PLANO N° 11
1/1				Sustituye a
				Sustituido por



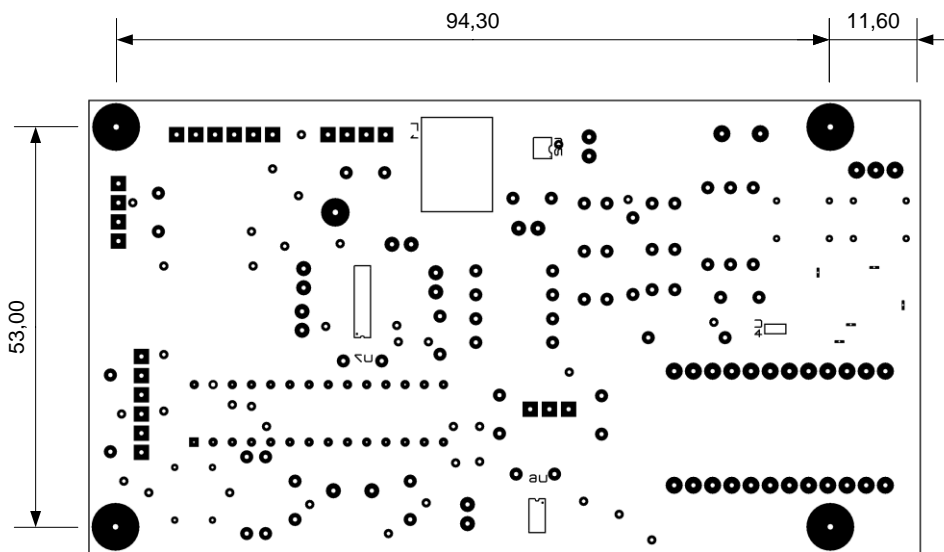
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA 1/1	Plano de componentes cara BOTTOM placa superior			PLANO N° 12
				Sustituye a
				Sustituido por



**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

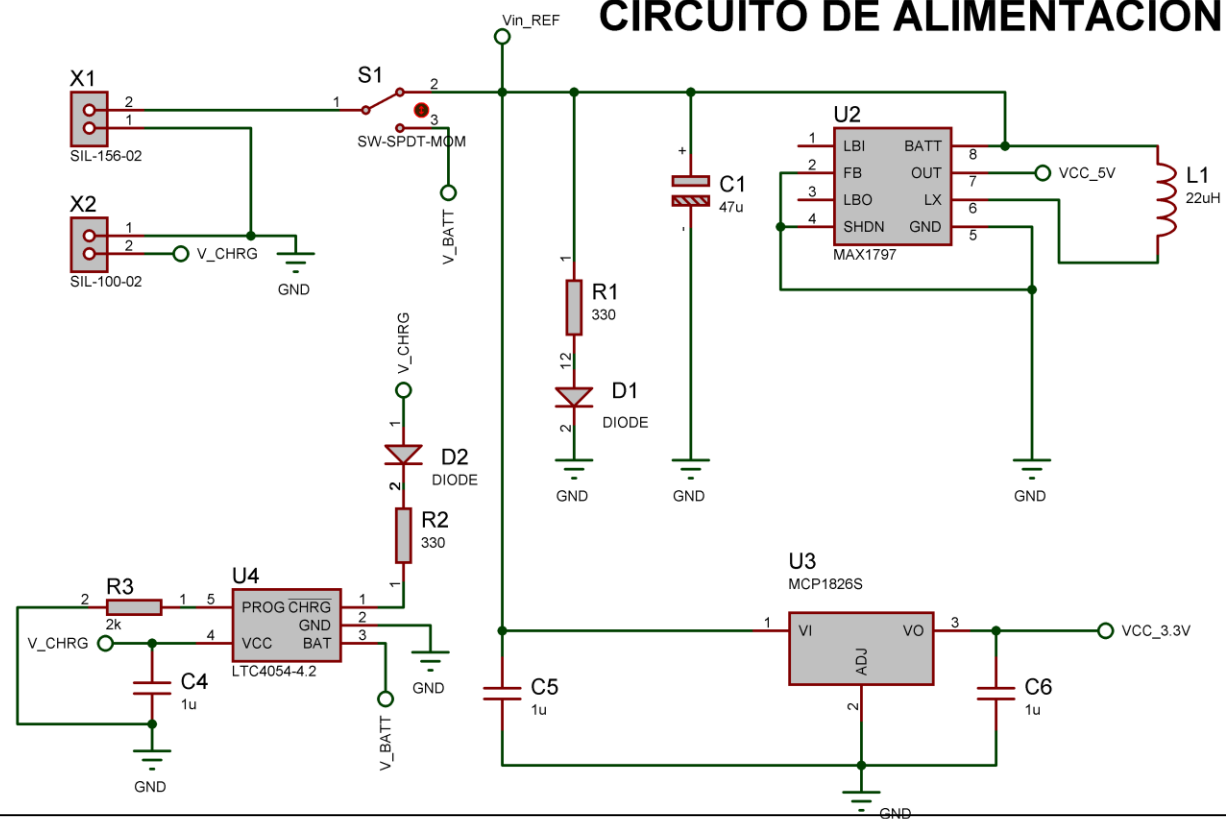
	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA 1/1	Plano de componentes cara TOP placa inferior			PLANO N° 13
				Sustituye a
				Sustituido por



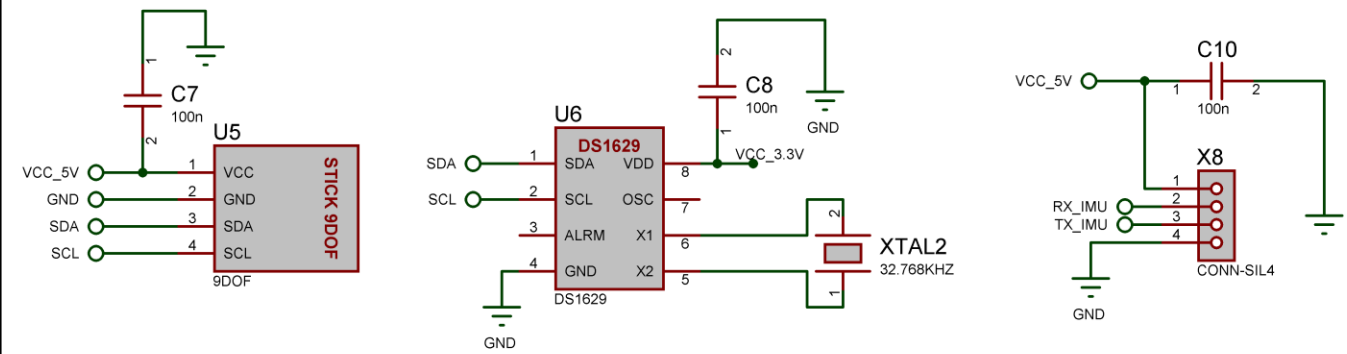
**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA 1/1	Plano de componentes cara BOTTOM placa inferior			PLANO N° 14
				Sustituye a
				Sustituido por

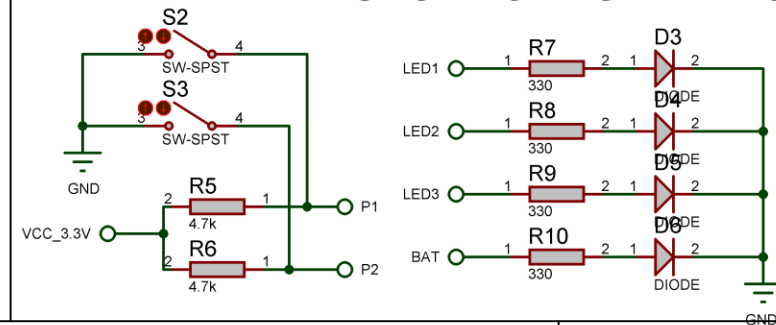
CIRCUITO DE ALIMENTACIÓN



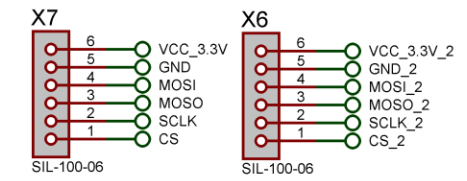
SENSORES



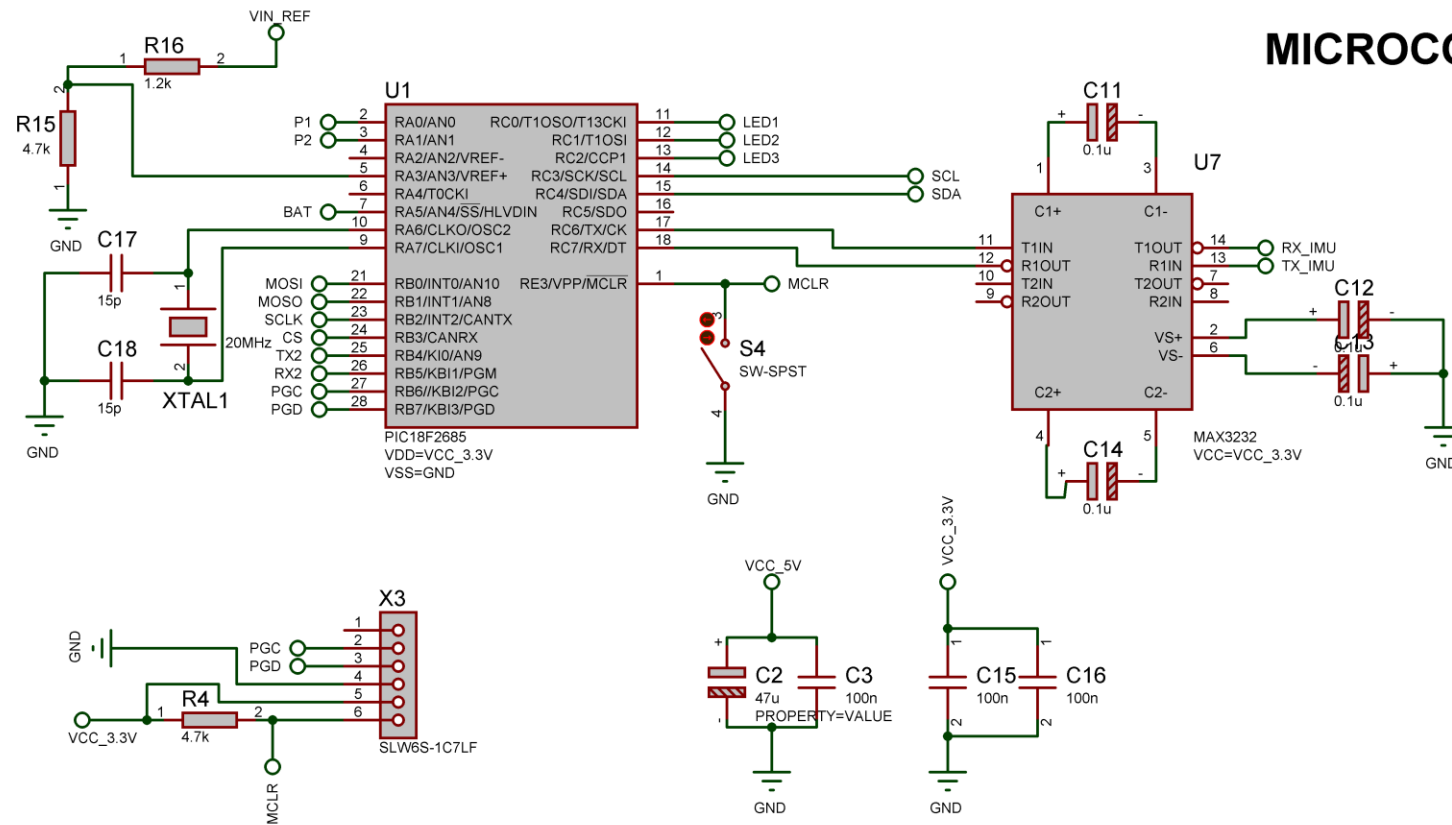
PULSADORES Y LEDS



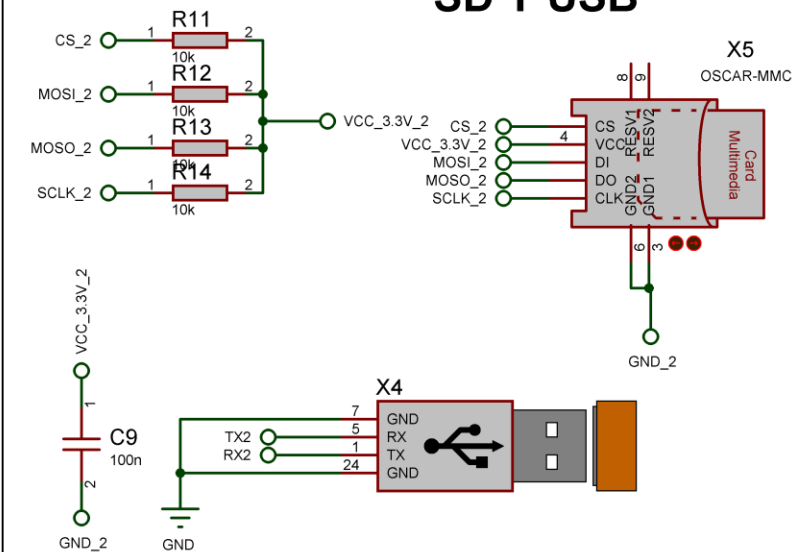
CONECTORES ENTRE PLACAS



MICROCONTROLADOR



SD Y USB



MÁSTER EN INGENIERÍA MECATRÓNICA - TRABAJO FIN DE MÁSTER -

	FECHA	NOMBRE	FIRMA
Proyectado	20/06/13	OSCAR RIVERA	
Dibujado	20/06/13	OSCAR RIVERA	
Comprobado	20/06/13	FERNANDO NUÑO	
ESCALA			
N/A	Plano esquemático		
			PLANO N° 15
			Sustituye a
			Sustituido por

ID.	NOMBRE	FABRICANTE	COD. FABRICANTE	SUMINISTRADOR	COD. SUMINISTRADOR
U1	Microcontrolador	MICROCHIP	PIC18F27J13-I/SP	RS	703-7753
U2	Convertidor elevador 5v 350mA	MAXIM	MAX1797EUA+	MAXIM	-
U3	Regulador de tensión 3,3v 1A LDO	MICROCHIP	MCP1826S-3302E/AB	RS	669-5174
U4	Cargador ion-litio 4,2v	LINEAR	LTC4054LES5-4.2	FARNELL	210-2610
U5	Tarjeta sensores inerciales	SPARKFUN	SEN-10724	SPARKFUN	-
U6	Reloj/Termómetro	MAXIM	DS1629	FARNELL	972-4745
U7	Chip adaptador de señal TTL	MAXIM	MAX3232CSE+	RS	761-3763
X1	Conector batería	PRECI-DIP	811-S1-010-10-014101	RS	701-9991
X2	Conector cargador hembra jack 3,5mm	Decelect Forgos	IES101-4	RS	513-0421
X3	Conector PICKIT 3	PRECI-DIP	811-S1-010-10-014101	RS	701-9991
X4	Conector puerto serie	FTDI chip	UM232R	FARNELL	114-6036
X5	Conector para tarjeta SD	HRS	DM1AA-SF-PEJ(72)	RS	685-0779
X6	Conector entre placas (Placa superior)	Winslow	W34425G	RS	401-699
X7	Conector entre placas (Placa inferior)	Winslow	W34425G	RS	401-699
S1	Interruptor general	EOZ	09.03290.01	RS	204-7865
S2	Pulsador cambio de estado P1	TE Connectivity	FSM8JH	RS	479-1435
S3	Pulsador cambio de estado P1	TE Connectivity	FSM8JH	RS	479-1435
S4	Pulsador de RESET del micro	TE Connectivity	FSM8JH	RS	479-1435

ID.	NOMBRE	FABRICANTE	CANTIDAD	SUMINISTRADOR	COD. SUMINISTRADOR
R1,R2,R7, R8,R9,R10	Resistencia 330Ω 1/2W 5%	RS	6	RS	739-8714
R3	Resistencia 2KΩ 1/2W 5%	RS	1	RS	739-8773
R4,R5,R6,R15	Resistencia 4K7Ω 1/2W 5%	RS	4	RS	739-8802
R11,R12,R13,R14	Resistencia 10KΩ 1/2W 5%	RS	4	RS	739-8837
R16	Resistencia 1K2Ω 1/2W 5%	RS	1	RS	739-8755
C1,C2	Condensador electrolítico 47uF 25v	Panasonic	2	RS	727-1091
C4,C5,C6	Condensador cerámico 1uF/50v	Kemet	4	RS	538-1578
C3,C7,C8,C9, C10,C15,C16	Condensador cerámico 100nF/50v	AVX	7	RS	537-3707
C11,C12,C13,C14	Condensador electrolítico 1uF/50v	Panasonic	4	RS	414-9143
C17,C18	Condensador cerámico 15pF/50v	Vishay	2	RS	736-8833
D1,D2,D3,D4,D5,D6	Diodo LED	Kingbright	6	RS	247-1814
L1	Bobina 22uH	Colicraft	1	FARNELL	228-9312
XTAL1	Cristal oscilador 40MHz	Euroquartz	1	RS	693-8759
XTAL2	Cristal oscilador 32,768Hz	IQD	1	RS	226-1443

**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	
Proyectado	20/06/13	OSCAR RIVERA		
Dibujado	20/06/13	OSCAR RIVERA		
Comprobado	20/06/13	FERNANDO NUÑO		
ESCALA	Lista de materiales			PLANO N° 16
N/A				Sustituye a
				Sustituido por