

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

**ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA**



PROYECTO FIN DE CARRERA

**Diseño e Implementación de una Unidad de
Telemetría de bajo costo para Sistemas de
Navegación Autónomos.**

Presidente:

Secretario:

Vocal:

Tutores:

Autor:

NOTA:

**TITULACIÓN: Sistemas Electrónicos
TUTOR: Roberto Esper-Chaín Falcón
AUTOR: Frank Pérez Paz
FECHA: Junio 2012**

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

**ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA**



PROYECTO FIN DE CARRERA

**Diseño e Implementación de una Unidad de
Telemetría de bajo costo para Sistemas de
Navegación Autónomos.**

**TITULACIÓN: Sistemas Electrónicos
TUTOR: Roberto Esper-Chaín Falcón
AUTOR: Frank Pérez Paz
FECHA: Junio 2012**

*A mis abuelos.
Por todo el amor que ellos me han dado.*

Agradecimientos

Los agradecimientos es la parte que más me ha costado escribir de la memoria y los he dejado para el final, ya que a mí personalmente me cuesta muchísimo expresar mis sentimientos en palabras. Quisiera agradecer a tanta gente que me ha arropado a lo largo de mi vida que me es imposible nombrarlos a todos y cada uno de ellos, así que nombraré a las personas que me han ayudado en este proyecto y a los que me han hecho crecer como persona sin olvidarme de todos los demás.

En primer lugar agradecer a todos los profesores que me han impartido clases y me han ayudado a entender este mundo tan complejo de los dispositivos electrónicos. Algunos de ellos aparte de ser simples profesores, han ido más allá de su labor y me han tratado como un amigo más, que siempre es de agradecer y que contribuyeron a que estos años de universidad no se hicieran tan duros.

También quiero agradecer especialmente a mi tutor Roberto, el cual de manera desinteresada y teniendo muchísimo trabajo, no tuvo ningún problema en aceptar guiarme en este proyecto y en hacer huecos en su apretada agenda para atenderme y responder a mis preguntas.

De mis años de universidad tengo que agradecer a esos amigos con los que he pasado muchas horas en el laboratorio y también en asaderos. A Fayna con la cual he compartido muchas mañanas y tardes en la universidad y que ha sido mi apoyo en muchos otros temas que no tienen que ver con los estudios. A Ray que aunque no me haya mencionado en su memoria, sé que es un amigo con el cual puedo contar para todo lo que necesite. Y a Vitín que está un poco loco de la cabeza dejándose esa barba, lo aprecio mucho.

Agradecer a todos mis tíos, por quererme y cuidarme no sólo como un sobrino, sino como si fuera su hijo. Todos son muy importantes para mi y aunque están todos muy lejos, yo siempre los tengo presentes.

Agradecer a mis abuelos, los cuales han sido mi referencia para ser la persona que soy hoy en día. Mi abuelo Julio que fue el mejor hombre que he conocido y en el que me he querido reflejar para llegar a ser como él. A mi abuelo Firin, que se ha dejado la piel para sacar adelante a toda la familia y es otro de mis referentes como persona. A mi abuela Gladis, la persona más bondadosa que existe, repartiendo hasta lo que ella no tiene, y a mi abuela Nelva que me ha ayudado espiritualmente, debido a la distancia, a conseguir mis objetivos y a consolarme en los momentos malos. Los quiero mucho.

No puede faltar mi padre, que aunque por circunstancias de la vida, nos vimos obligados a separarnos, yo sé que él nunca ha dejado de quererme como su hijo “el campeón”, y que yo después de lo que hemos pasado, vuelvo a verlo como la persona que era “mi padre”.

Como no, también tengo que incluir a la persona más importante para mí, me refiero a esa mujer que no sólo me ha dado la vida, sino que me ha dado todo lo que tengo y me ha educado lo mejor que ha podido pese a las momentos difíciles. Nunca le agradezco lo suficiente la forma en la que me ha criado y esa alegría que trasmite a todos. Gracias mi madre querida, te quiero muchísimo, eres la mejor madre del mundo.

Por último pero no menos importante, agradecer a Alba mi novia y a la cual le debo más cosas que la que ella se imaginará nunca. Te encontré en un momento delicado para mí y tú has sabido guiarme y hacerme ver muchas cosas de las cuales no me deba cuenta. Eres la que me da fuerzas todos los días y la que me hace vencer todos los obstáculos. Gracias conseguir aguantarme y a la vez quererme cada día. Te quiero mucho.

ÍNDICE

1.0 INTRODUCCION	1
1.1 Antecedentes	1
1.2 Objetivos del Proyecto.....	1
1.3 Estructura de la Memoria	2
1.4 Peticionario	3
2.0 SENSORES.....	5
2.1 ¿Que Son los Sensores?.....	5
2.2 Características Principales	5
2.3 Redes de Sensores	7
2.4 Características de las Redes	8
2.5 Aplicaciones	10
3.0 EI ESTÁNDAR IEEE 802.15.4	13
3.1 Introducción.....	13
3.2 Topología de la Red.....	15
3.3 Arquitectura.....	16
3.3.1 Subcapa MAC.....	17
3.3.2 Capa Física (PHY)	22
3.4 Modelo de transferencia de datos.....	25
3.4.1 Transferencia de datos de un dispositivo a un coordinador.....	26
3.4.2 Transferencia de datos desde un coordinador a un dispositivo.....	27
3.4.3 Transferencia de igual a igual (peer to peer).....	28
3.5 Seguridad.....	28
4.0 SISTEMAS MICROELECTROMECÁNICOS (MEMS)	31
4.1 ¿Qué son los MEMS?	31
4.2 Aplicaciones	33
4.3 Fabricación.....	34
4.4 Tipos de MEMS utilizados	35
4.4.1 Acelerómetros.....	35
4.4.2 Giroscopios.....	37
4.4.3 Barométricos.....	39
4.4.4 Magnéticos	40
5.0 COMPONENTES USADOS EN EL DISEÑO	43
5.1 Requisitos del proyecto	43

5.1.1 Alimentación	43
5.1.2 Sensores inerciales (acelerómetros y giroscopio)	44
5.1.3 Sensores Barométricos.....	44
5.1.4 Sensor Magnético	45
5.1.5 Comunicación inalámbrica	45
5.1.6 Conectividad USB.....	46
5.2 Alimentación (TPS60500).....	46
5.3 Microcontrolador (ATXMEGA128A3).....	49
5.3.1 Diagrama de Bloques	51
5.3.2 Características Principales.....	52
5.3.3 Interrupciones	53
5.3.4 Puertos	54
5.3.5 Bus SPI.....	57
5.3.6 Bus USART	59
5.3.7 Bus I2C.....	65
5.4 Radio (AT86RF212)	66
5.4.1 Características	66
5.4.2 Diagrama de bloque.....	68
5.4.3 Diagrama de conexión	69
5.4.4 Conexiones al microcontrolador.....	69
5.4.5 Protocolo SPI del AT86Rf212	73
5.4.6 Estados del Radio Transmisor.	76
5.4.7 Descripción funcional.....	78
5.5 Acelerómetro (ADXL345)	81
5.5.1 Descripción	81
5.5.2 Diagrama Funcional.....	82
5.5.3 Salida de datos	83
5.6 Giroscopio (L3G4200D).....	84
5.6.1 Descripción	84
5.6.2 Diagrama de bloques.....	85
5.6.3 Diagrama funcional	86
5.7 Barómetro (MPL115A1).....	87
5.7.1 Características del MPL115A1	87
5.7.2 Diagrama de bloques y conexiones	88
5.7.3 Comunicación y coeficientes.....	89
5.8 Magnético (HMC5883L)	90

5.8.1 Características Principales del HMC5883	90
5.8.2 Diagrama de Aplicación del IC antiguo (HMC5883)	91
5.8.3 Diagrama de Aplicación del IC nuevo (HMC5843)	91
5.9 Ultrasonido	92
5.9.1 Características	93
5.9.2 Aplicaciones.....	93
5.9.3 Diagrama de aplicación.....	95
5.9.4 Especificaciones	97
5.10 Dispositivo USB (CP2102).....	98
5.10.1 Características	98
5.10.2 Diagrama de bloques.....	99
5.10.3 Diagrama de conexiones	100
6.0 DISEÑO DE LA PCB	101
6.1 Diseño de las bibliotecas.....	101
6.1.1 Bibliotecas graficas	102
6.1.2 Biblioteca de FootPrint.....	103
6.2 Esquemáticos.....	105
6.3 Paso de los esquemáticos a la PCB.....	107
6.3.1 Reglas de diseño	108
6.3.2 Posicionamiento y Enrutamiento.....	109
7.0 FABRICACIÓN DE LA PCB.....	113
7.1 Soldadura.....	113
7.2 Tipos de encapsulados.....	114
7.3 Proceso de Soldadura.....	116
8.0 PROGRAMACIÓN	121
8.1 Programa Utilizado.....	121
8.2 Dispositivo JTAG.....	124
8.3 Pruebas Pre-programación	126
8.4 Descripción del Software	127
9.0 Pruebas de funcionamiento	129
10.0 Conclusiones y Líneas Futuras	137
11.0 Bibliografía.....	139
12.0 PRESUPUESTO.....	143
12.1 Coste de los Recursos Humanos	144
12.2 Coste de los Recursos Hardware	145
12.3 Costes de los Recursos Software.....	146

12.4 Costes de Fabricación de Prototipos	146
12.5 Costes de la Redacción del Proyecto	147
12.6 Gastos Generales y Beneficio Industrial	147
12.7 Coste Total	148
Anexos	149
Lista de Materiales	149
Esquemáticos.....	155
Máscaras PCB	160

ÍNDICE DE FIGURAS

Figura 1 Estructura de una red de sensores	8
Figura 2 Baud Rate de los diferentes estándar.	13
Figura 3 Tipos de Redes.....	16
Figura 4 Arquitectura del estándar 802.15.4.	16
Figura 5 Formato General de una Trama MAC.	18
Figura 6 Estructura de trama de datos.	19
Figura 7 Estructura de la trama ACK.	19
Figura 8 Estructura de la Trama de Comandos.....	20
Figura 9 Estructura de la Trama Beacon.....	20
Figura 10 Estructura de una Supertrama.	21
Figura 11 Estructura de una supertrama con GTS.	22
Figura 12 PPDU.....	23
Figura 13 Estructura de Canales.....	24
Figura 14 Canalización de la banda 2.4 GHz.	24
Figura 15 Dispositivos al coordinador con beacons.	26
Figura 16 Dispositivos al coordinador sin Beacons.	26
Figura 18 Coordinador al dispositivo sin beacons habilitados.	27
Figura 17 Coordinador al dispositivo con beacons habilitados.	27
Figura 19 Ejemplos de MEMS basados en microactuadores.	32
Figura 20 Proceso de grabado.....	34
Figura 21 Proceso de fabricación de un engranaje.	35
Figura 22 Acelerómetro de 3 ejes	36
Figura 23 Placas de un sensor capacitivo.....	36
Figura 24 Acelerómetro Piezoeléctrico.....	37
Figura 25 Giroscopio MEMS.	38
Figura 26 Sistema de Referencia Inercial.	38
Figura 27 Avión Despegando.....	39
Figura 28 Sensor Hall.	41
Figura 29 Transformador Lineal Variable.....	41
Figura 30 Regulador de tensión.....	47
Figura 31 Diagrama de Aplicación del TPS60500.....	48
Figura 32 Esquema Xmega.	50
Figura 33 Registro de Estado (SREG)	50
Figura 34 Arquitectura del Xmega.....	51

Figura 35 Diagrama de Bloques del microcontrolador.....	52
Figura 36 Funcionalidad general de los pines.....	55
Figura 37 Push-Pull.....	55
Figura 38 Pull-Down.....	56
Figura 39 5 Pull-Up.....	56
Figura 40 Bus-Keeper.....	56
Figura 41 Interconexión Maestro-Esclavo del bus SPI.....	57
Figura 42 Modos de transferencia del SPI.....	58
Figura 43 Diagrama de Bloques de la USART.....	59
Figura 44 UCPHA y INVEN.....	60
Figura 45 Formato de la trama USART.....	61
Figura 46 Ejemplo de la trama de comienzo en modo asíncrono.....	63
Figura 47 Proceso de recuperación de datos.....	63
Figura 48 Ejemplo del bit de Stop y del siguiente bit de Start.....	64
Figura 49 Topología del Bus I2C.....	65
Figura 50 Modo de Transacción del bus I2C.....	66
Figura 51 Diagrama de bloques del AT86Rf212.....	68
Figura 52 Circuito de aplicación del AT86Rf212.....	69
Figura 53 Conexión microcontrolador con el At86Rf212.....	70
Figura 54 Transmisión del SPI.....	71
Figura 55 Acceso de Lectura.....	73
Figura 56 Acceso de escritura.....	74
Figura 57 Ejemplo de la secuencia de escritura y de lectura.....	74
Figura 58 Ejemplo de lectura de una trama del buffer.....	74
Figura 59 Ejemplo de escritura de una trama del buffer.....	74
Figura 60 Ejemplo de lectura de la SRAM.....	75
Figura 61 Ejemplo de escritura de la SRAM.....	75
Figura 62 Estados del AT86Rf212.....	76
Figura 63 Trama de la estructura de la capa PHY.....	79
Figura 64 Estructura de la MPDU.....	80
Figura 65 Diagrama de bloques ADXL345.....	82
Figura 66 Diagrama del acelerómetro (ADXL345).....	82
Figura 67 Salida de datos utilizando justificación a la derecha.....	83
Figura 68 Salida de datos utilizando justificación a la izquierda.....	83
Figura 69 Ejes de medida.....	84
Figura 72 Stream Mode.....	85
Figura 73 Diagrama de bloques del giroscopio L3G4200D.....	85

Figura 70 Bypass Mode.....	85
Figura 71 FIFO Mode.....	85
Figura 74 Diagrama Funcional del Giroscopio.....	86
Figura 75 Diagrama de Bloques MS5540C.....	87
Figura 76 Dimensiones del Barómetro.....	88
Figura 77 Diagrama de bloques y conexiones del MPL115A1.....	88
Figura 78 Secuencia de lectura del Barómetro.....	89
Figura 79 Diagrama de conexiones del HMC5843.....	91
Figura 80 Diagrama de bloques y conexión del HMC5883.....	91
Figura 81 Rango de frecuencias de las ondas sonoras.....	92
Figura 82 Ultrasonido en delfines.....	92
Figura 83 Dispositivos de sonar.....	94
Figura 84 Ecografía.....	95
Figura 85 Conexión del Ultrasonido.....	96
Figura 86 Análisis Transient del Simulador MicroSim.....	96
Figura 87 Integrado CP2102.....	98
Figura 88 Bloques del CP2102.....	99
Figura 89 Diagrama de conexión del CP2102.....	100
Figura 90 Programa Altium Designer Release 10.....	101
Figura 91 Símbolo de un condensador de Tántalo.....	102
Figura 92 Footprint de un conector mini USB.....	103
Figura 93 Medidas del conector mini USB.....	104
Figura 94 Biblioteca grafica incluyendo el footprint.....	105
Figura 95 Página en Blanco de esquemáticos.....	106
Figura 96 Introducción del componente.....	106
Figura 97 Iconos.....	106
Figura 98 Esquemático General.....	107
Figura 99 Footprints importados a la PCB.....	108
Figura 100 Ejemplo de conexión Bus USB.....	109
Figura 101 Capas del diseño.....	110
Figura 102 Diseño de nuestra PCB.....	111
Figura 103 Mascara de Kapton y placa PCB.....	117
Figura 104 Maquina de succión.....	117
Figura 105 Maquina LPKF ZelPlace BGA.....	118
Figura 106 LPKF ProtoFlow.....	119
Figura 107 Vistas del microscopio.....	120
Figura 108 Entorno programa AVR-STUDIO.....	122

Figura 109 Opciones generales y librerías.....	123
Figura 110 Programador JTAG.....	124
Figura 111 Conector de JTAG de 10 Pins.....	125
Figura 112 Flujograma del Código C.	127
Figura 113 Lectura del estado del ATXMEGA128A3.	130
Figura 114 Configuración HyperTerminal.....	131
Figura 115 Test del puerto USART.....	132
Figura 116 Datos del Acelerómetro.....	133
Figura 117 Datos del Giroscopio.....	134
Figura 118 Datos Acelerómetro y Giroscopio.....	134
Figura 119 Lectura del Barómetro 1.....	135
Figura 120 Lectura de los barómetros.	136

ÍNDICE DE TABLAS

Tabla 1 Características IEEE 802.15.4.	14
Tabla 2 Parámetros de Modulación.	25
Tabla 3 Señales del Bus SPI.	57
Tabla 4 Ecuaciones para el cálculo del Baud Rate.	60
Tabla 5 Tipos de interrupciones.	71
Tabla 6 Funciones del Pin SLP_TR.	72
Tabla 7 Protocolo SPI.	73
Tabla 8 Tiempos de la PPDU.	79
Tabla 9 Reglas de Diseño.	109
Tabla 10 Tipos de encapsulados.	114
Tabla 11 Pines del puerto JTAG.	125
Tabla 12 Coste Total.	143

1.0 INTRODUCCIÓN

1.1 Antecedentes

Tradicionalmente los sistemas de navegación han sido sistemas muy voluminosos y muy caros, puesto que los sensores inerciales, giróscopos y brújulas giroscópicas eran elementos extremadamente caros, y pesados. Con el avance de las tecnologías de sensores integrados en dispositivos monolíticos, estos dispositivos han reducido costos, volumen y consumo de potencia. Gracias a esto, están apareciendo una nueva serie de dispositivos de navegación autónoma de bajo coste.

1.2 Objetivos del Proyecto

El objetivo del proyecto es hacer una unidad sensora que incluya:

- 1) Unidad inercial de 6 ejes (3 aceleraciones lineales y 3 aceleraciones angulares).
- 2) Brújula triaxial, para determinación de rumbo.
- 3) Sensores de presión atmosférica, para estimación de altitud y velocidad del aire indicada (AIS).
- 4) Sensores ultrasónicos para detección de altura en proximidad/obstáculos.
- 5) Capacidad de cómputo para procesar la información de los sensores.
- 6) Capacidad inalámbrica para transmitir los parámetros y recibir órdenes.
- 7) Capacidad para expandir los sensores.

Todo mediante una red inalámbrica de sensores que es un conjunto de pequeños chips integrados en una placa que se encargan de medir las distintas magnitudes físicas, y de enviar la información a un microcontrolador que se va a encargar de procesarlas y enviarlas por radio mediante el estándar IEEE 802.15.4 a un nodo central que se dedicará a procesar la información obtenida.

1.3 Estructura de la Memoria

Esta memoria estará dividida en varias partes, empezando por las explicaciones de los diferentes dispositivos que utilizamos, hasta cómo se ha realizado el diseño de la PCB y cómo se ha programado los diferentes circuitos integrados para ver y comprobar su funcionamiento.

En el segundo capítulo hablaremos de los sensores, ¿qué son?, sus principales características y para qué se utilizan.

En el tercer capítulo nos adentraremos en el estándar de comunicación IEEE 802.15.4. Aquí hablaremos de las características de este protocolo: cómo es su arquitectura, los diferentes tipos de comunicación que existen y del tema de seguridad, que es muy importante para las redes inalámbricas.

En el cuarto capítulo describiremos los dispositivos conocidos como MEMS; nos adentramos en el mundo de la nanotecnología y detallaremos a estos minúsculos sistemas electro-mecánicos.

En el quinto capítulo mencionaremos los componentes que utilizamos para realizar nuestro proyecto, y desglosaremos cada circuito integrado por separado para ir explicando su funcionamiento detalladamente.

En el sexto capítulo se hará referencia al diseño de nuestra placa PCB. En el mismo, se describirá el proceso de diseño para conseguir nuestros esquemáticos.

El séptimo capítulo se dedicará a describir el proceso con el que se fabricó la PCB.

El octavo capítulo tiene como objetivo explicar la parte en la que programamos nuestro microcontrolador, así como el programa que utilizamos para desarrollar nuestro software y el programador.

En el noveno capítulo se expondrán las razones por las que los dispositivos funcionaron correctamente, dándose a conocer las pruebas que se realizaron a cada integrado.

En los últimos capítulos se detallarán las conclusiones, el presupuesto, la bibliografía y los anexos.

1.4 Peticionario

Actúa como petionario del presente Proyecto Fin de Carrera la Escuela de Ingeniería de Telecomunicación y Electrónica de la ULPGC, como requisito para la obtención del título de Ingeniero Técnico de Telecomunicación, especialidad en Sistemas Electrónicos.

2.0 SENSORES

2.1 ¿Que Son los Sensores?

Los sensores son dispositivos electrónicos con la capacidad de detectar la variación de una magnitud física tal como temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc.; y de convertir el valor de ésta, en una señal eléctrica, ya sea analógica o digital.

Un sensor es un elemento idóneo para tomar, percibir o censar una señal física proveniente del medio ambiente y convertirla en una señal de naturaleza transducible. Un sensor o captador convierte las variaciones de una magnitud física en variaciones de una magnitud eléctrica o magnética.

Cada dispositivo medirá una magnitud y nos entregará la medida mediante alguna codificación entendible. Si queremos saber más de una magnitud, lo normal es que utilicemos una red de sensores, con los dispositivos que midan las magnitudes que necesitamos.

Las redes de sensores serán explicadas en el apartado 2.3 “Redes de Sensores”. En dicho apartado, explicaremos las distintas redes que podemos tener y la forma de conexión de cada una de ellas.

2.2 Características Principales

Los sensores tendrán que informarnos de cómo realiza las medidas y cómo nos las da a conocer, bien sea mediante algún indicador, o bien sea a través de algún conversor analógico digital o display.

Las características de los transductores se pueden agrupar en dos grandes bloques:

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

- Características estáticas, que describen la actuación del sensor en régimen permanente o con cambios muy lentos de la variable a medir.
- Características dinámicas, que describen el comportamiento del sensor en régimen transitorio.

Características Estáticas:

- Rango de medida: el conjunto de valores que puede tomar la señal de entrada comprendidos entre el máximo y el mínimo detectados por el sensor con una tolerancia de error aceptable.
- Resolución: indica la capacidad del sensor para discernir entre valores muy próximos de la variable de entrada. Indica qué variación de la señal de entrada produce una variación detectable en la señal de salida.
- Precisión: define la variación máxima entre la salida real obtenida y la salida teórica dada como patrón para el sensor.
- Repetitividad: Indica la máxima variación entre valores de salida obtenidos al medir varias veces la misma entrada con el mismo sensor y en idénticas condiciones ambientales.
- Linealidad: un transductor es lineal si existe una constante de proporcionalidad única que relaciona los incrementos de la señal de salida con los respectivos incrementos de la señal de entrada en todo el rango de medida.
- Sensibilidad: indica la mayor o menor variación de la señal de salida por unidad de la magnitud de entrada. Cuanto mayor sea la variación de la señal de salida producida por una variación en la señal de entrada, el sensor es más sensible.
- Ruido: cualquier perturbación aleatoria del propio sistema de medida que afecta la señal que se quiere medir.

Características Dinámicas:

- **Velocidad de respuesta:** mide la capacidad del sensor para que la señal de salida siga sin retraso las variaciones de la señal de entrada.
- **Respuesta en frecuencia:** mide la capacidad del sensor para seguir las variaciones de la señal de entrada a medida que aumenta la frecuencia, generalmente los sensores convencionales presentan una respuesta del tipo pasabajos.
- **Estabilidad:** indica la desviación en la salida del sensor con respecto al valor teórico dado, al variar parámetros exteriores distintos al que se quiere medir (condiciones ambientales, alimentación, etc.).

2.3 Redes de Sensores

La mayor parte de estos sensores adolecen de la capacidad de procesar y analizar los datos que detectan, limitándose a funcionar como un transductor que realiza la medición de una o más variables del entorno y envía dicha información a un procesador central. Pero si hacemos que estos sensores se comuniquen con otros sensores semejantes podemos crear las llamadas redes de sensores inalámbricos (Wireless Sensor Networks, WSN), consistentes en macronets ad-hoc formadas por multitud de sensores individuales que intercambian información entre sí, sin necesidad de cables, y mediante un protocolo de comunicación preestablecido.

Cada nodo de la red consta de un dispositivo con microcontrolador, sensores y transmisor/receptor, y forma una red con muchos otros nodos. Y al coordinar la información entre un importante número de nodos, éstos tienen la habilidad de medir un medio físico dado con gran detalle. Los nodos distribuidos se comunican se manera inalámbrica a un gateway central, el cual proporciona una conexión al entorno cableado donde usted puede adquirir, procesar, analizar y presentar sus datos de medida.

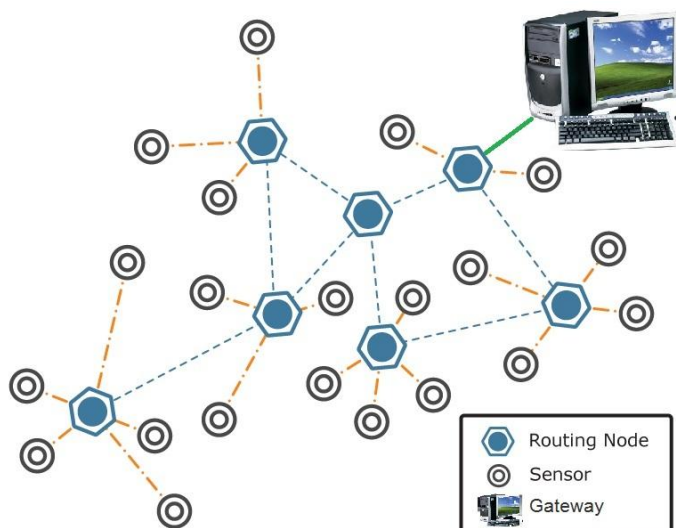


Figura 1 Estructura de una red de sensores

Los nodos no tienen un conocimiento de la topología de la red, deben descubrirla. La idea básica es que cuando un nuevo nodo, al aparecer en una red, anuncia su presencia y escucha los anuncios *broadcast* de sus vecinos. El nodo se informa acerca de los nuevos nodos a su alcance y de la manera de encaminarse a través de ellos, a su vez, puede anunciar al resto de nodos que pueden ser accedidos desde él. Transcurrido un tiempo, cada nodo sabrá que nodos tiene alrededor y una o más formas de alcanzarlos.

2.4 Características de las Redes

Las características principales que posee una red inalámbrica de sensores son las siguientes:

- **Tiempo de vida:** las redes de sensores están formadas por distintos chips que están fabricados para aguantar muchas condiciones de funcionamiento, por lo que son bastante duraderos y resistentes.
- **Cobertura de la red:** al ser aparatos de bajo consumo, las transmisiones no cubren largas distancias, pero al poder comunicarse entre los nodos, se amplía el campo de cobertura.

- Coste y facilidad de instalación: los costes son muy pequeños y la instalación es sencilla, puesto que los dispositivos estarán la mayoría de las veces alimentados por baterías.
- Precisión y frecuencia de las mediciones: los dispositivos que existen actualmente, son bastante precisos, y trabajan a una frecuencia suficiente como para que no se pierda ninguna información de los datos que estamos midiendo.

Los valores principales que caracterizan al nodo sensor son los siguientes:

- Flexibilidad: las WSN tienen capacidad de auto-restauración, es decir, si se avería un nodo, la red encontrará nuevas vías para encaminar los paquetes de datos.
- Robustez: partiendo de la flexibilidad, la red sobrevivirá en su conjunto, aunque haya nodos individuales que pierdan potencia o se rompan.
- Seguridad: un nodo sensor puede ser fácilmente accesible, ya que los datos son enviados a través de un medio inalámbrico, por ello se han creado Primitivas de seguridad e Infraestructuras de claves, Sistemas de autocontrol (p. ej. Sistemas de auditoría e IDS), Protocolos de funcionamiento seguro, etc.
- Capacidad de comunicación: los nodos actúan como elementos de transmisión-recepción de datos al poder reenviar los mensajes transmitidos por nodos más lejanos.
- Capacidad de computación: cada nodo consta de un microcontrolador por lo tanto es capaz de procesar los datos recibidos de los sensores.
- Facilidad de sincronización: habitualmente construidas *ad hoc*, por lo que se facilita la capacidad de comunicación entre los nodos.
- Tamaño y coste: los nodos son de tamaño reducido por lo que los costes de fabricación son reducidos, y si se fabrican a grandes escalas, los costes de fabricación serían considerablemente pequeños.

- Gasto de energía: Debido a las limitaciones de la vida de la batería, los nodos se construyen teniendo presente la conservación de la energía, y generalmente pasan mucho tiempo en modo 'durmiente' (*sleep*) de bajo consumo de potencia.

2.5 Aplicaciones

Existen infinidad de aplicaciones de las redes de sensores, gran parte de estas aplicaciones están en el campo de la agricultura, en la medicina, en seguridad, y en muchísimos lugares más.

La agricultura constituye uno de los campos en los que se prevé poder implantar rápidamente una red de sensores, para favorecer:

- Monitorización del medio ambiente: Monitorización del tiempo atmosférico, análisis de factores medioambientales en zonas de riesgo (p. ej. Cauces fluviales, cultivos...).
- Agricultura de precisión: control de condiciones climáticas, recolección de datos sobre el terreno, control de riegos, suministro de datos para los agricultores, calculo de insumos y de agua.
- Agricultura ecológica: medición de niveles de contaminantes, o de sustancias no autorizadas, determinación de ausencia de determinados productos químicos.
- Servicios de Automatización: control de invernaderos, alimentación e identificación de animales y controles sanitarios.

En la medicina se pueden utilizar para:

- Monitorización de los pacientes: ritmo cardíaco, temperatura, etc.
- Control de enfermos: registrando los datos de electrocardiogramas, tasa de oxígeno en sangre, etc. y mandándolo al PDA del médico para su control.

En los entornos de seguridad, también son muy útiles su utilización para:

- Control de la seguridad de un perímetro ante posibles intrusos.
- Identificación y seguimiento de tropas o vehículos militares.
- Detección de armas químicas y biológicas.

Otras aplicaciones:

- Monitorización continua o esporádica de entornos y situaciones críticas (centrales nucleares, bomberos, manejo de sustancias peligrosas...) o de entornos (oficinas, zonas residenciales...).
- Monitorización de redes de ordenadores.
- Avisador de riesgo de epidemias (gripe, fiebres, etc.).
- Observatorios terrestres y marítimos (costas).

3.0 EI ESTÁNDAR IEEE 802.15.4

3.1 Introducción

La clave de la motivación para el uso de tecnología inalámbrica es la reducción en los gastos de instalación, ya que nunca es necesario cambiar el cableado. Las redes inalámbricas implican un gran intercambio de información con un mínimo de esfuerzo de instalación. Esta tendencia es impulsada por la gran capacidad de integrar componentes inalámbricos de una forma más barata y el éxito que tienen otros sistemas de comunicación inalámbrica, como los teléfonos móviles.

Además se prefiere que los aparatos consuman muy poca energía ya que el cambio constante de las baterías se considera impráctico. La tecnología 802.11 (WLAN) resultaría sofocante ya que sólo satisface los requerimientos de conexión. Bluetooth se concibió originalmente como un sustituto del cable, pero se ha llevado sobre un camino más complejo, haciéndolo impráctico para aplicaciones de bajo consumo. La tendencia a la complejidad ha incrementado los costos previstos para esta tecnología. Ambos dispositivos, bluetooth y 802.11, requerirían un cambio de baterías algunas veces al año, lo que resulta impráctico.

Al ver estas necesidades se decide crear un nuevo estándar llamado comercialmente ZigBee que tenga una baja transferencia de datos y tenga un consumo considerablemente bajo, para no tener que cambiar la batería. En la siguiente figura se establece la velocidad de ZigBee, en comparación con los otros tipos de estándares:

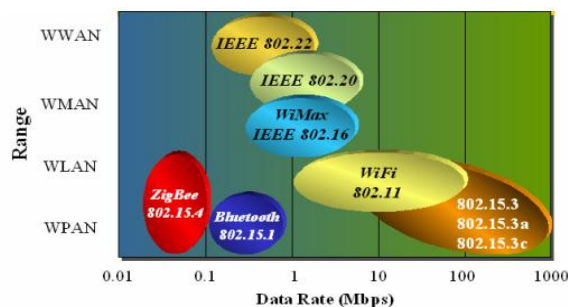


Figura 2 Baud Rate de los diferentes estándar.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

El estándar IEEE 802.15.4 nació gracias a la unión de dos grupos especialistas en estándares (ZigBee y el grupo 15 de trabajo IEEE 802), que se unieron para crear un nuevo estándar para redes inalámbricas de bajo consumo y reducidos costos para utilizarlo en ambientes industriales y caseros.

En el año 2003 se aprueba y se da a conocer oficialmente como un nuevo grupo de trabajo con el nuevo estándar de baja transmisión en redes inalámbricas para áreas personales (LR-WPAN), conocido como IEEE 802.15.4.

IEEE 802.15.4 se especificó como un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (low-rate wireless personal area network, LR-WPAN). La actual revisión del estándar se aprobó en 2006. También es la base sobre la que se define la especificación de ZigBee, cuyo propósito es ofrecer una solución completa para este tipo de redes construyendo los niveles superiores de la pila de protocolos que el estándar no cubre.

Este estándar se creó con las siguientes características básicas:

Bandas de Frecuencia	868 MHz(Europa) 915 MHz(América) 2.4 GHz(Internacional)
Alcance	10-20 Metros
Latencia	<15ms
Nº de Canales	868 MHz :1 915 MHz: 10 2.4 GHz: 16
Rango de Transmisión de Datos	868 MHz: 20 Kb/s 915 MHz: 40 Kb/s 2.4 GHz: 250 Kb/s

Tabla 1 Características IEEE 802.15.4.

3.2 Topología de la Red

Las redes inalámbricas son las encargadas de realizar la topología de la red, el mantenimiento de la misma y de mantener los servicios de enlaces necesarios para el direccionamiento y la seguridad de los datos. Esta red también tiene que cumplir con lo mismo, sólo que tiene el reto de que tiene que hacerlo bajo los requerimientos de bajo consumo y ahorro de energía.

El estándar IEEE 802.15.4 dispone de dos tipos de dispositivos:

- FFD (Full Function Device): tienen funcionalidad completa, pueden operar tanto como coordinador (PAN) o como dispositivo de red (end point).
- RFD (Reduced Function Device): sólo puede recibir órdenes o informar a un coordinador cercano. Es decir, son los sensores y actuadores que se utilizan para las aplicaciones más simples.

Este estándar soporta 2 tipos de topologías para la conexión de red, la topología en estrella y la topología de igual a igual:

- Topología en estrella (star): sólo posee un dispositivo que hace la función de nodo principal y que tiene las características FFD, los demás dispositivos son RFD y van conectados a este coordinador de red.
Como el nodo principal tiene la capacidad de organización, necesita un consumo de energía mayor, por lo que suele tener una fuente de alimentación conectada a la red eléctrica.
- Topología de igual a igual (peer to peer): en esta topología, todos los nodos suelen ser FFD y todos se pueden comunicar con todos, sin que el coordinador (PAN) les tenga que dar permiso. Esta topología es muy similar a las topologías en mallas o ad-hoc.

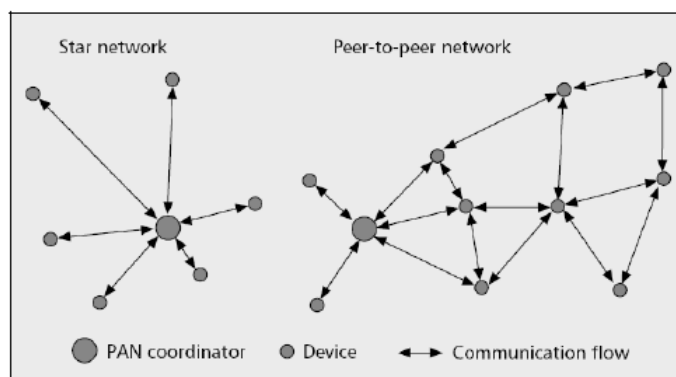


Figura 3 Tipos de Redes.

La topología que se escogerá, dependerá en gran parte de la aplicación que utilizaremos, y es que en periféricos podríamos usar la topología en estrella con conexiones de baja potencia, mientras que en entornos de seguridad, necesitaríamos cubrir mayor área de cobertura, por lo que implementaríamos la topología de igual a igual.

Cada PAN puede seleccionar un identificador único. Este identificador permite la comunicación entre dispositivos dentro de una red usando direcciones cortas y permitiendo transmisiones entre dispositivos sobre redes independientes.

3.3 Arquitectura

La arquitectura del estándar 802.15.4 incluye 2 capas en diseño. Por un lado está la capa física (PHY) que define las Frecuencias de Radio (RF) que se utilizan para los enlaces; y por otro lado incluye la subcapa MAC que proporciona acceso al canal físico, a través de los diferentes tipos de transferencias.

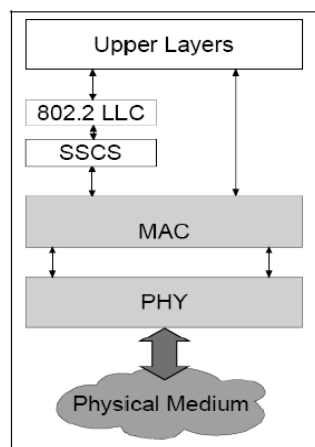


Figura 4 Arquitectura del estándar 802.15.4.

Además incluye dos capas superiores que se pueden ver en la figura 4, que son la capa de red y la capa de aplicación. La capa de red provee la configuración de red, manipulación y encaminamiento de mensajes. Por otra parte, la capa de aplicación se encarga de realizar las tareas que se asignan al dispositivo.

3.3.1 Subcapa MAC

La subcapa Mac nos brinda dos servicios, el Servicio de Datos y el Administrador de Servicios:

- Servicio de Datos: habilita la transmisión y recepción de datos del protocolo MAC a través de la capa física.
- Administrador de Servicios: es el que realiza la interface entre la subcapa MAC y el dispositivo. Este administrador de servicios tiene muy pocas primitivas y comandos (26) comparado con 802.15.1 (bluetooth) (163), por lo que lo hace más simple, versátil, barato de fabricar y con un hardware sencillo.

Formato general de las Tramas MAC

Tiene cuatro tipos de tramas:

- Tramas de datos (Data Frame)
- Tramas de confirmación (Acknowledgment Frame)
- Tramas de comando MAC (MAC Command Frame)
- Tramas beacon "guía"(Beacon Frame)

Sólo las tramas de datos y beacon contienen información proveniente de capas superiores, mientras que las tramas de confirmación y comandos son utilizadas por la MAC para comunicarse entre dispositivos de igual a igual (peer to peer).

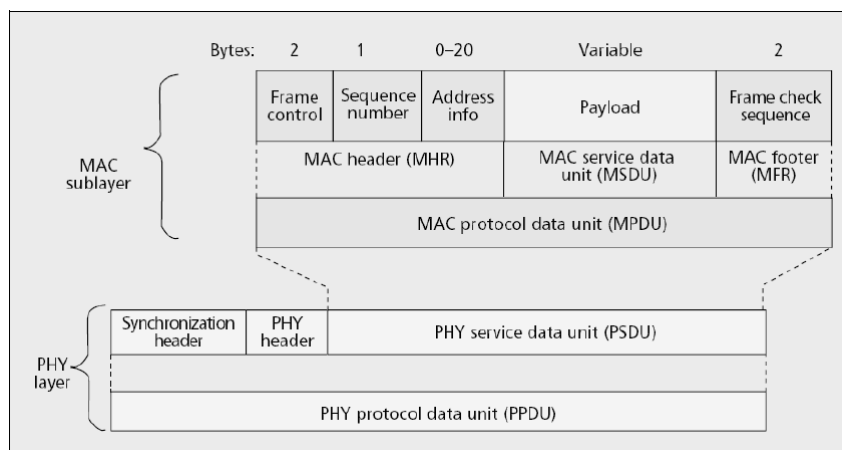


Figura 5 Formato General de una Trama MAC.

En la figura 5 se puede observar la estructura de una trama MAC. A continuación se detallarán de forma breve las distintas unidades que la forman:

La trama de Datos MAC: está compuesta por el encabezado (MAC Header), unidad de servicio de datos (MAC Service Data) y termina con el chequeo de la trama (MAC footer).

- Frame Control: se encuentra dentro del encabezado (MAC Header) y es la que nos dirá el contenido del resto de la trama de datos, es decir, nos informará del tipo de trama que se pretende transmitir, su formato, el campo de dirección e incluso también controla los mensajes ACK.
- Data Sequence Number: es una secuencia de números que se envían para que luego de la transmisión poder comprobar si ha sido satisfactoria o no, comparando esta secuencia con la secuencia ACK, que debe de tener la misma secuencia de números que la anterior transmitida.
- Address information: contiene la información de la dirección de la fuente, y puede variar desde 0 a 20 bytes ajustándose y logrando mayor eficacia del protocolo al mantener los paquetes lo más reducidos posible.
- Payload: es de longitud variable; sin embargo, la trama completa no debe exceder de 127 bytes de información. Estos datos dependerán del tipo de trama.

- Frame Check Sequence: es la trama de chequeo de 16 bits CRC (Cyclic Redundancy Check).

Para que la capa MAC de Datos pueda ser enviada tiene que añadirse la capa PHY que es la capa física. Esta capa añade SHR y PHR y formarán parte del paquete de datos de la capa física PPDU como se observa en la siguiente figura:

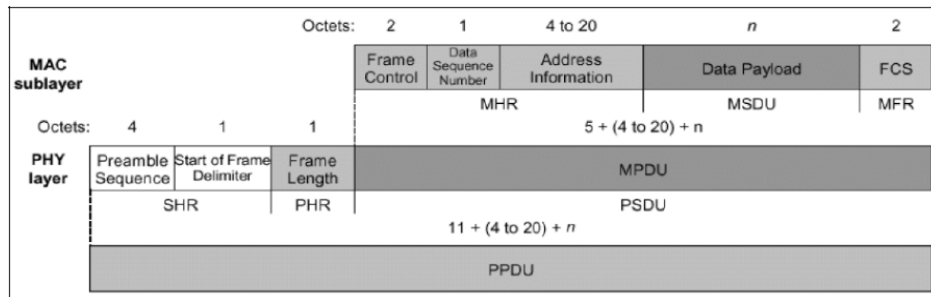


Figura 6 Estructura de trama de datos.

La trama ACK (Acknowledgment Frame) se produce para hacer saber que la trama de datos ha sido recibida sin dar error. Esta trama es producida por el receptor que aprovecha el tiempo de silencio que se origina después de la transmisión para enviarla a la subcapa MAC.

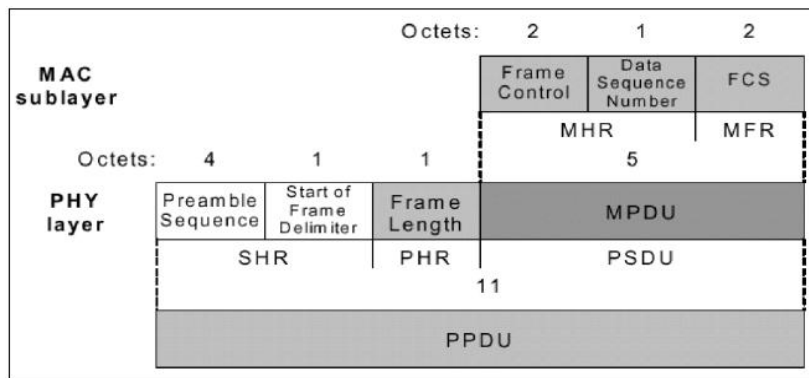


Figura 7 Estructura de la trama ACK.

La trama de Comandos Mac se origina de igual forma que la trama de Datos, lo que se agregan las tramas específicas de comandos. Estos comandos permiten la configuración de los nodos a distancia, permitiendo que un director de red pueda configurar los dispositivos de la red sin importar el tamaño de esta.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

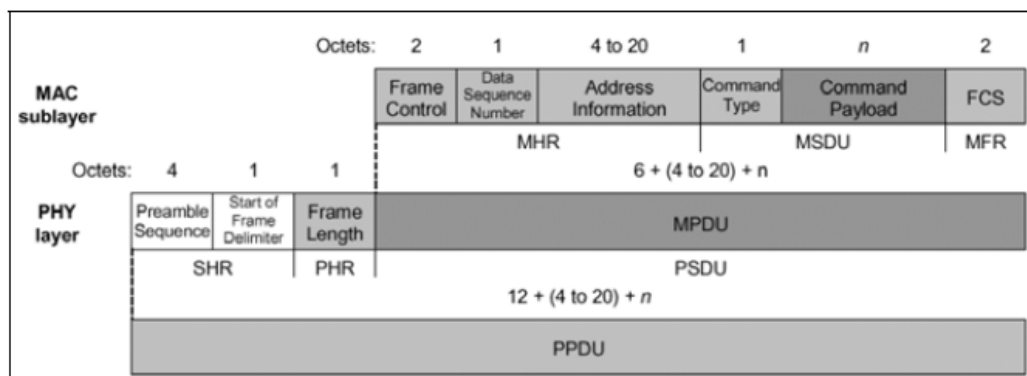


Figura 8 Estructura de la Trama de Comandos

Como puede verse en la figura 8, sólo añade la trama Command Type a la estructura de Datos MAC.

La Trama Beacon es la encargada de la sincronización de los dispositivos; mediante esta trama, los dispositivos despiertan de su estado de letargo al escuchar su dirección y regresan al estado sleep una vez concluida la transmisión. Con esto conseguimos un ahorro de energía.

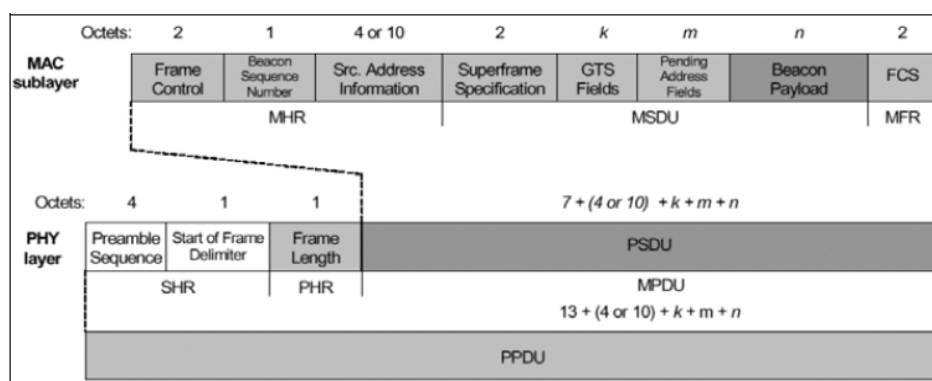


Figura 9 Estructura de la Trama Beacon.

Esta estructura de beacon contiene las denominadas estructuras supertramas.

Estructuras Supertramas (SuperFrames)

Algunas aplicaciones requieren un ancho de banda dedicado para lograr estados de espera y conseguir un consumo inferior, pasando de modo activo a entrar a un modo inactivo. Para esto, el estándar IEEE 802.15.4 creó la estructura llamada supertramas.

Su formato viene definido por el coordinador de red y se limita mediante los beacon. La trama beacon se divide en 16 intervalos de tiempo de igual duración.

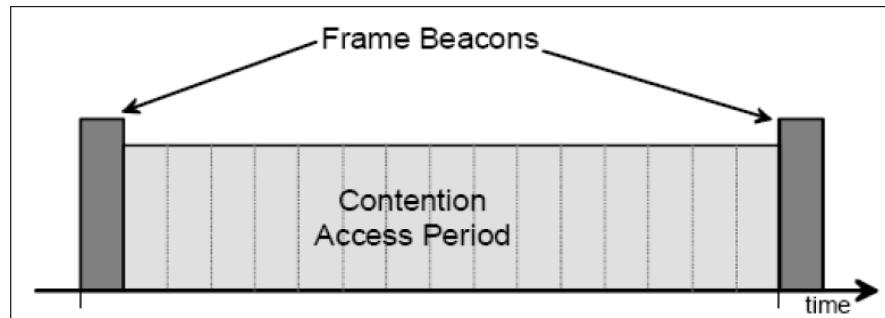


Figura 10 Estructura de una Supertrama.

El beacon es transmitido en el primer intervalo de tiempo de cada supertrama. Si algún dispositivo no requiere utilizar la estructura de las supertramas, habrá que deshabilitar la transmisión de beacons.

Los beacons son utilizados por el coordinador de red para sincronizar los dispositivos vinculados, identificar el PAN y para describir la estructura de la supertrama.

Para las aplicaciones que requieran un ancho de banda específico o que por haber muchos dispositivos y mucho tráfico en la red se necesite tener prioridad para obtener una mínima latencia, el coordinador PAN puede hacer que partes de una trama activa sean dedicadas para estas aplicaciones. Estas partes son llamadas Intervalos de Tiempo Garantizados (GTS).

El GTS forma el período libre de contención (CFP), el cual siempre aparece al final de una supertrama activa, inmediatamente después del Periodo de Contención de Acceso (CAP), que esta después de un beacon. El coordinador PAN puede asignar hasta siete GTS y un GTS puede ocupar más de un intervalo de tiempo.

Todas las transacciones basadas en contención de acceso deben de terminarse antes de que el CFP comience.

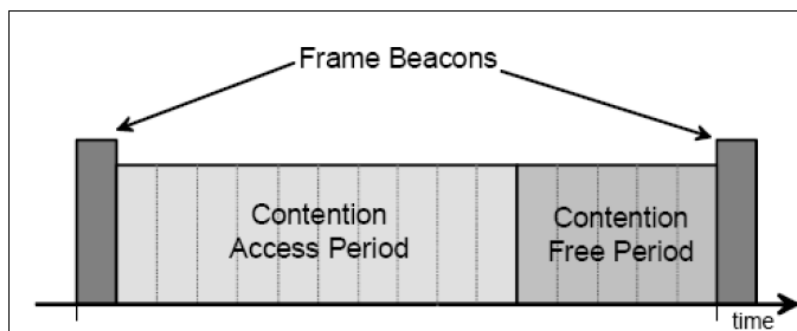


Figura 11 Estructura de una supertrama con GTS.

También cada dispositivo transmitiendo un GTS debe asegurarse de que su transacción se ha completado antes de que ocurra el próximo GTS o el fin de la CFP.

3.3.2 Capa Física (PHY)

La 802.15.4 ofrece dos opciones de capa PHY. Ambas se basan en métodos de secuencia directa de espectro extendido (DSSS), que son de bajo costo a lo hora de implementar el circuito integrado y comparten la misma estructura de paquetes low-duty-cycle (Bajo Ciclo de Ocupación) con operaciones de bajo consumo de energía. La principal diferencia entre las capas está en las frecuencias que utilizan.

La PHY 2.4 GHz está disponible a nivel mundial, mientras que la capa PHY 868/915 MHz está únicamente para Europa y Estados Unidos. La banda de 2.4 GHz ofrece menores costes y un mercado más amplio, mientras que la 868/915 MHz ofrece mayor rango de alcance y menos interferencias al estar en una frecuencia alejada de las bandas WLAN, microondas, teléfonos móviles, etc.

Otra característica que distingue la diferencia entre las dos opciones de transmisión es que la PHY de 2.4 GHz ofrece un rango de transmisión de 250 kb/s, mientras que la PHY de 868/915 MHz tiene un rango de transmisión de 20 kb/s (868 MHz) y 40 kb/s (915 MHz).

Cada aplicación podrá elegir la PHY que más le convenga, pues habrá aplicaciones que no necesiten mucha densidad de datos y que podrán operar con

la banda 868/915 MHz consiguiendo más rango, cubriendo mayor área geográfica, y logrando así tener que poner menos nodos en la red. Y aplicaciones para las que se necesite mayor rango de transmisión con poca latencia.

Estructura de la capa física (PHY)

Para mantener la interface con la capa MAC, la capa PHY tiene la misma estructura que la MAC y comparten una sola estructura de paquete:

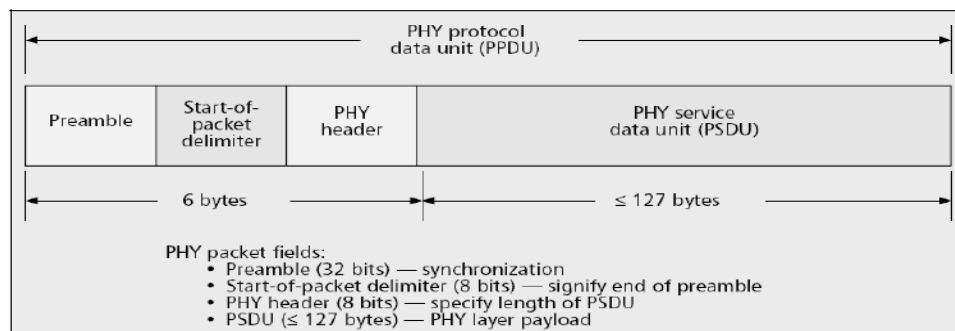


Figura 12 PPDU.

Cada paquete o Unidad de Datos del Protocolo de la PHY (PPDU) empieza con un encabezado de sincronización de 32 bits (Preamble), seguido de un encabezado de la capa física que indica la longitud del paquete (PHY Header) y a continuación, la capa física de la unidad de servicio de datos que contiene una longitud ≤ 127 bytes (PSDU).

El preámbulo (Preamble) es utilizado para administración de símbolos y tiempos de chips, aunque también puede utilizarse para ajustar variaciones de frecuencia.

La PHY header se utiliza para indicar el tamaño que tendrá la PSDU, y para ello se utilizan 8bits.

La PSDU es donde se encuentra los datos y puede tener una longitud entre 0 y 127 bytes. Dependiendo de la aplicación se puede utilizar una longitud mayor o menor de datos.

Canalización de la PHY

La PHY de 868 MHz soporta un sólo canal que va desde los 868.0 – 868.6 MHz. La PHY de 915 MHz soporta diez canales que van desde los 902.0 – 928.0 MHz. Para reducir costes de manufactura, se considera que las dos bandas están lo suficientemente cercanas para compartir el mismo hardware.

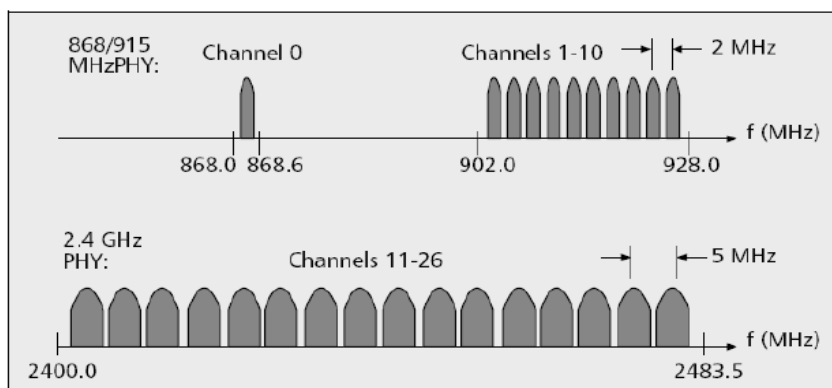


Figura 13 Estructura de Canales.

La banda de los 2.4 GHz, contiene 16 canales de 2 MHz que van desde los 2.4 – 2.4835 GHz. Con un amplio espectro (5 MHz) entre canales, para facilitar la filtración en la transmisión y en la recepción.

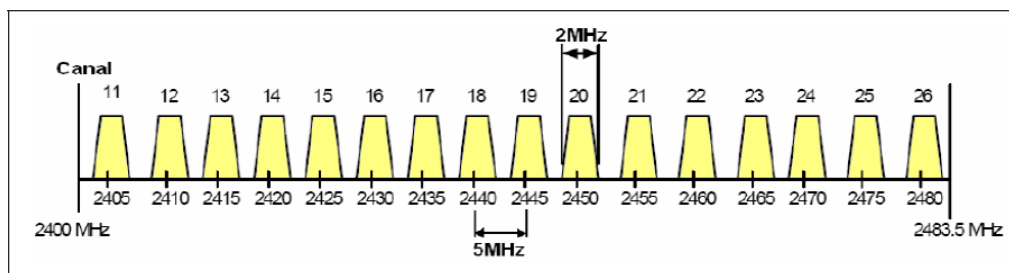


Figura 14 Canalización de la banda 2.4 GHz.

Como la banda de los 2.4 GHz se utiliza para múltiples aparatos electrónicos dentro del hogar, la red tendrá que tener gran capacidad de relocalización dentro del espectro.

Para ello la PHY tiene algunas funciones como la detección de niveles de energía, indicadores de calidad en el enlace así como la conmutación de canales, lo que permite una mejor asignación de canales y una mayor agilidad en la selección de frecuencias.

Modulación

Para la PHY 868/915 MHz se utiliza la modulación de DSSS (Espectro Expandido de Secuencia Directa) y se modula utilizando una portadora en BPSK (Transmisión por Desplazamiento Binario de Fase). Antes de la modulación se utiliza una codificación de datos diferencial para permitir una recepción diferencial coherente de baja complejidad.

En la siguiente tabla se resumen los parámetros de modulación para ambas bandas PHY:

PHY.	Banda.	Parámetros de los Datos			Parámetros del chip	
		Velocidad de bits (kbps)	Velocidad de símbolos (kbaud)	Modulación	Velocidad de chip (Mchips/s)	Modulación
868/915 MHz	868.0-868.6 MHz	20	20	BPSK	0,3	BPSK
	902.0-928 MHz	40	40	BPSK	0,6	BPSK
2.4 GHz PHY	2.4-4.4835 GHz.	250.	62.5	16-ary ortogonal.	2	O-QPSK

Tabla 2 Parámetros de Modulación.

La PHY de 2.4 GHz emplea otra técnica en la cual se emplea modulación semi-ortogonal basada en métodos DSSS. Los datos binarios son agrupados en símbolos de 4 bits y cada símbolo especifica una de las 16 secuencias de transmisión semi-ortogonales de código de pseudo-ruido (PN).

3.4 Modelo de transferencia de datos

Hay tres modelos de transferencia de datos:

1. Desde un dispositivo a un coordinador.
2. Desde un coordinador a un dispositivo de red que reciba los datos.
3. Transferencia de datos entre dos dispositivos iguales (peer to peer).

Dependiendo de la topología que utilizamos, se pueden usar los tres modos o sólo utilizar dos, puesto que la transferencia de igual a igual en la topología en estrella no es factible, ya que no existe comunicación entre dispositivos y coordinador, sino sólo entre los dispositivos.

Dependiendo si la red soporta los beacons, se puede habilitar o no, utilizándose la transmisión normal, sin embargo el beacon es requerido para la asociación de la red.

3.4.1 Transferencia de datos de un dispositivo a un coordinador.

Si la red está utilizando los beacons, el dispositivo que desea transmitir esperara el beacon y cuando es encontrado, el dispositivo se deberá sincronizar con la estructura de la supertrama. Cuando el dispositivo encuentra el momento adecuado transmite la trama de datos utilizando CSMA-CA ranurado al coordinador. El coordinador enviará una trama de confirmación (ACK) si ha recibido correctamente los datos; de esta manera se completa la transacción.

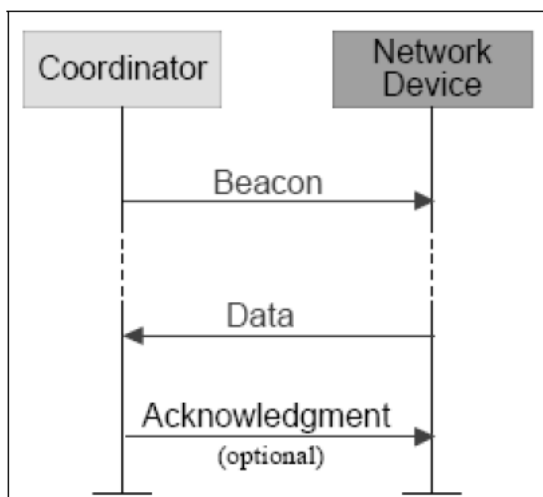


Figura 15 Dispositivos al coordinador con beacons.

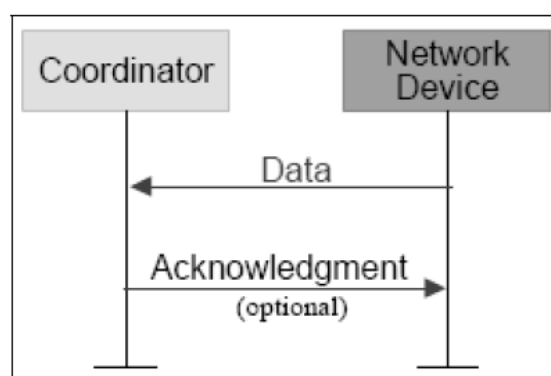


Figura 16 Dispositivos al coordinador sin Beacons.

Cuando no se utilizan los beacons, simplemente el dispositivo envía los datos al coordinador utilizando CSMA-CA no ranurado y espera la trama de reconocimiento, completándose así la transacción.

3.4.2 Transferencia de datos desde un coordinador a un dispositivo.

Si se utilizan beacons, el coordinador enviará un beacon indicando en él que hay un mensaje de datos pendiente. El dispositivo comprobará los beacons y si ve que hay un mensaje pendiente, transferirá un comando MAC con CSMA-CA ranurado, pidiendo los datos al coordinador. El coordinador expedirá una trama de confirmación exitosa de datos y luego enviará el mensaje de datos que tenía pendiente. El dispositivo lo recibirá y transmitirá otra trama de confirmación. Una vez finalizada la transmisión el coordinador borrará el mensaje de la lista de mensajes pendientes de la beacon.

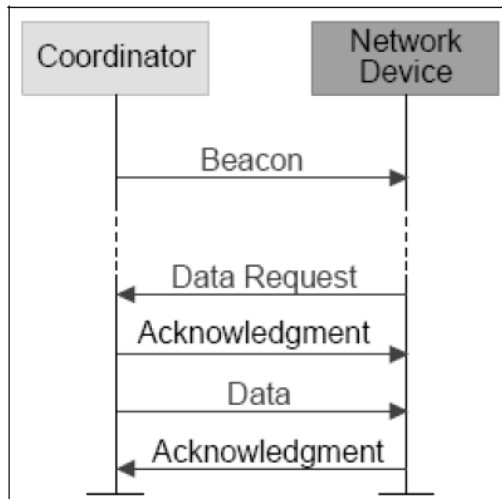


Figura 17 Coordinador al dispositivo con beacons habilitados.

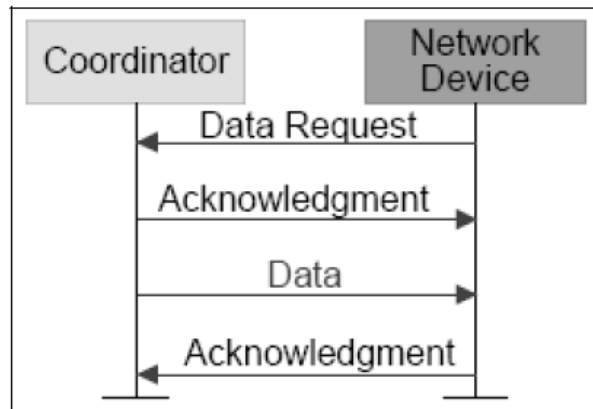


Figura 18 Coordinador al dispositivo sin beacons

habilitados.

Cuando no se utilizan beacons el coordinador guardará el mensaje hasta que el dispositivo envíe una petición de información. Cuando el dispositivo haga contacto con el coordinador mediante CSMA-CA no ranurado transmitiendo un comando MAC, el coordinador enviará una trama de confirmación y posteriormente si hay datos pendientes, el coordinador transferirá estos datos al dispositivo mediante CSMA-CA no ranurado. Si no hay datos pendientes, el coordinador enviará una trama con longitud cero, para indicar que no hay datos pendientes al dispositivo. El dispositivo contestará con otra trama de confirmación de datos.

3.4.3 Transferencia de igual a igual (peer to peer)

Como todos los dispositivos pueden comunicarse entre ellos, dentro del campo de influencia deben estar sincronizados en todo momento. Cuando algún dispositivo quiera comunicarse, simplemente enviará los datos utilizando CSMA-CA no ranurado.

3.5 Seguridad

Para garantizar la interoperabilidad y el buen funcionamiento entre los dispositivos, se implementan en la subcapa MAC algunos parámetros de seguridad. Se incluye una lista de control de acceso (ACL) y se usa criptografía simétrica para proteger las tramas transmitidas. Estos parámetros no serán implementados en todo momento y en todos los dispositivos. Las capas superiores son las encargadas de ver cuando se utilizan o no la seguridad en la capa MAC:

- **Servicios de Seguridad:** utiliza un mecanismo de seguridad de clave simétrica que generaran claves controladas por capas superiores. Estas claves serán generadas, transmitidas y almacenadas en modo seguro. La administración y establecimiento de estas claves serán responsabilidad del desarrollador.
- **Control de Acceso:** es un servicio que brinda la posibilidad de comunicarse con ciertos dispositivos y bloquear a los que no estén en una lista de dispositivos con los que se pueden comunicar o no. Los dispositivos que no estén en la lista, no pueden recibir tramas de datos.
- **Encriptación de Datos:** se utiliza un cifrado para proteger los datos de ser leídos por otros dispositivos. Esta clave de cifrado es conocida por los dispositivos que forman el grupo. Se puede encriptar tanto la trama de datos, de beacons así como de comandos.

- **Integridad de tramas:** utiliza el Código de Integridad de Mensajes (MIC), para proteger los datos que no tienen clave criptográfica. La clave usada para dar integridad a los mensajes puede ser compartida por todo el grupo o entre dos dispositivos.
- **Refresco Secuencial:** usa una secuencia ordenada de entrada para rechazar tramas que han sido reemplazadas por otras. Cuando una trama es recibida, el valor de refrescamiento es comprobado para ver si es la esperada.

Dependiendo del modo en el que el dispositivo trabaje y el nivel de seguridad deseado, se podrán ofrecer diferentes servicios de seguridad:

Modo inseguro: al no usarse seguridad, no existe ningún servicio de seguridad operando sobre el dispositivo.

Modo ACL: tienen servicios de seguridad limitada para la comunicación con otros dispositivos. En este modo, las capas superiores de la MAC se encargaran de rechazar las tramas de los dispositivos que no estén en la lista de los dispositivos aceptados.

Modo seguro: pueden disponer de cualquiera de los servicios de seguridad descritos anteriormente.

4.0 SISTEMAS MICROELECTROMECAÑICOS (MEMS)

4.1 ¿Qué son los MEMS?

Los MEMS son considerados miniaturas mecánicas y se consiguen mediante técnicas de micro-fabricación. Son sistemas electro-mecánicos de un tamaño considerablemente pequeño (del orden de nanómetros), pueden medir desde una micra hasta algunos milímetros. Existen dispositivos simples sin estructuras móviles, mientras que otros son extremadamente complejos con múltiples elementos móviles bajo el control de la microelectrónica de integrados.

Para que un dispositivo se considere un MEMS tiene que contener alguna funcionalidad mecánica, aunque las piezas se puedan o no mover. Existen muchos tipos de MEMS, algunos son del tipo microsensores, microactuadores, microeléctricos y microestructuras. Los microsensores y microactuadores son considerados como transductores (dispositivos que convierten la energía de una forma a otra). En concreto, los microsensores convierten una señal mecánica en otra eléctrica.

El desarrollo y las técnicas han hecho que se produzcan microsensores para realizar toda clase de mediciones como son la temperatura, presión, fuerzas inerciales, especies químicas, campos magnéticos, radiación ...

Sorprendentemente muchos de estos microsensores son más exactos en sus medidas que los homólogos de tamaño normal, teniendo un rendimiento mayor y dando mejores prestaciones.

Los MEMS utilizan la misma tecnología de fabricación que los circuitos impresos, produciéndose normalmente a partir de materiales como el silicio. Su presencia se ha ido incrementando en el mercado y su comercialización cada vez es mayor.

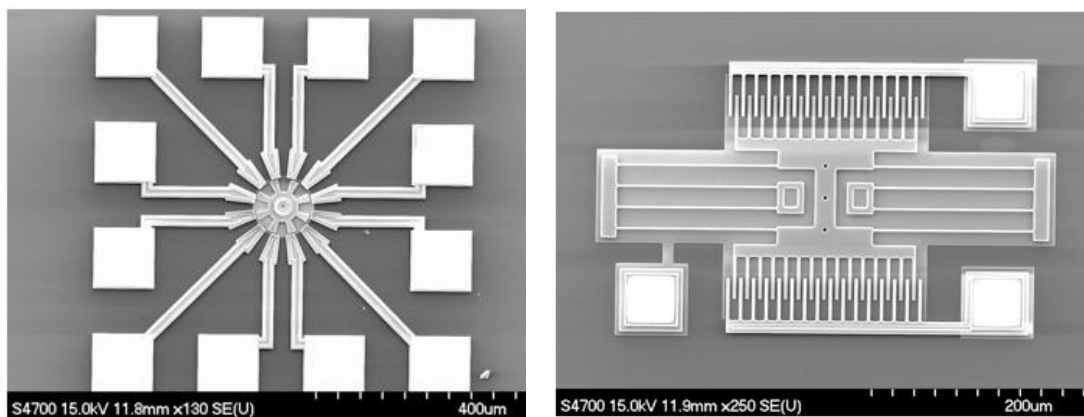


Figura 19 Ejemplos de MEMS basados en microactuadores.

Recientemente, un grupo de investigación sobre MEMS ha desarrollado una serie de microactuadores incluyendo: microválvulas para el control de los flujos de gas y líquidos, interruptores ópticos y espejos para redirigir o modular los haces de luz, control independiente para las pantallas de las matrices microespejos, microbombas para desarrollar presiones positivas de fluidos, microflaps para modular corrientes de aire sobre superficies de sustentación y muchos otros.

El verdadero potencial de MEMS comienza a verse cuando estos sensores miniaturizados, actuadores, y todas las estructuras se pueden combinar en un sustrato de silicio común, junto con los circuitos integrados. Mientras que los componentes electrónicos se fabrican con circuito integrado (IC), los componentes micromecánicos se fabrican utilizando sistemas compatibles y las piezas se graban en la oblea de silicio de manera selectiva o agregando nuevas capas estructurales para formar los dispositivos mecánicos y electromecánicos.

Se pueden combinar no sólo con la microelectrónica, también con otras tecnologías como la fotónica, la nanotecnología, etc., siendo evidente que estas tecnologías están llenas de numerosas oportunidades en el mercado comercial.

4.2 Aplicaciones

Son innumerables las aplicaciones en las que se pueden utilizar los MEMS y la nanotecnología.

A continuación pondremos algunas aplicaciones en las que ya se están utilizando los MEMS, incluso en alguna de ellas, los sensores son imprescindibles para que se pueda realizar tal aplicación:

- ***Biología***

En este campo está permitiendo nuevos descubrimientos en la ciencia y la ingeniería, tales como la Reacción en Cadena de la Polimerasa (PCR), microsistemas para la amplificación de ADN y su identificación, la electroforesis capilar, microscopios micromecanizados de efecto túnel (STM), los biochips de detección de sustancias químicas peligrosas y agentes biológicos y microsistemas para la detección de drogas de alto rendimiento, etc.

- ***Medicina***

Hay una amplia variedad de aplicaciones para MEMS en medicina. Los sensores de presión son los más utilizados a lo largo de los últimos años para medir la presión arterial, presión intrauterina durante el parto, presión sanguínea, etc.

- ***Comunicaciones***

Los componentes eléctricos, tales como inductores y condensadores sintonizables se pueden mejorar de manera significativa en comparación con sus homólogos integrados, si se realizan con MEMS y nanotecnología. Con la integración de tales componentes, el rendimiento de circuitos de comunicación mejorará, mientras que el área total del circuito, el consumo de energía y el coste se reducirían.

- ***Sensores inerciales***

Sensores MEMS inerciales, específicamente acelerómetros y giroscopios, están ganando rápidamente la aceptación del mercado. La tecnología MEMS ha

hecho posible integrar el acelerómetro y la electrónica en un solo chip de silicio con un costo de sólo unos pocos euros. Estos acelerómetros MEMS son mucho más pequeños, más funcionales, más ligeros, más fiables, y se producen por una fracción del costo de los elementos del acelerómetro macroescala convencional. Más recientemente, los giroscopios MEMS (es decir, los sensores de velocidad) se han desarrollado tanto para automóviles y aplicaciones de electrónica de consumo. Sensores inerciales MEMS hoy en día se están utilizando en los nuevos modelos de coches vendidos, así como en iPhone, Ipad, Wii, teléfonos móviles de última generación, etc.

Como estamos hablando de una tecnología punta, se podrá seguir creando cada vez más nuevas aplicaciones.

4.3 Fabricación

El proceso de fabricación de un MEMS es muy similar al de un circuito integrado, añadiéndose y eliminándose capas para lograr la estructura deseada.

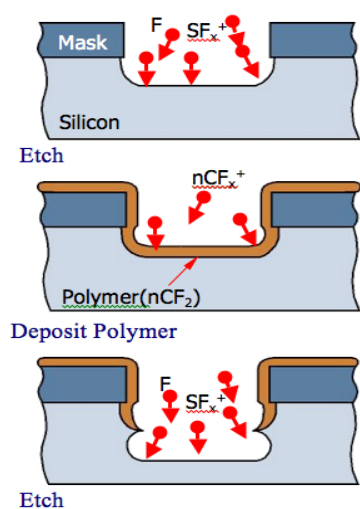


Figura 20 Proceso de grabado.

Podemos observar en la figura 20 cómo se realiza el proceso básico para fabricar un MEMS.

Primero se parte de la oblea de silicio, la cual es radiada con iones para conseguir darle la forma deseada.

Luego se coloca un polímero para cubrir las partes que no se quieren ionizar y se sigue ionizando las demás partes.

Siguiendo el proceso casi idéntico al de un circuito integrado.

En la figura 21 podemos observar otro proceso que se utiliza para la fabricación de un MEMS. En concreto, se pueden ver los pasos para la obtención de un engranaje a partir de una oblea. Se puede apreciar cómo primero se radian las zonas que no están protegidas por la máscara, para luego decapitar la zona

radiada y colocar el material del que sea el MEMS para que adquiera la forma que se quiere conseguir.

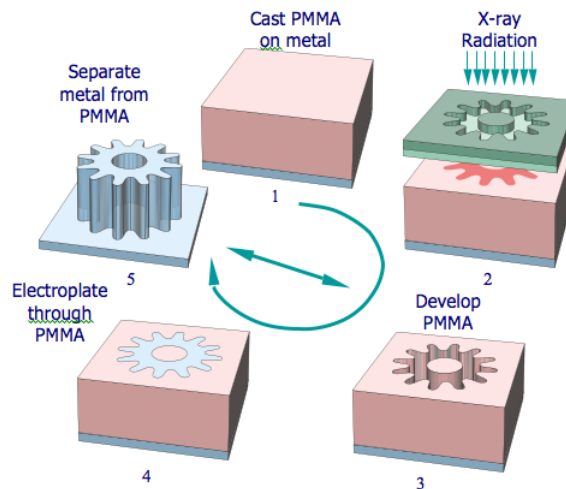


Figura 21 Proceso de fabricación de un engranaje.

Además existen otros muchos mecanismos para la construcción de un MEMS, como son por medio de Laser Micromecanizado, XeF₂ Grabado Fase Seca, Electro-descarga Micromecanizado y Haz de Iones Enfocado Micromecanizado.

4.4 Tipos de MEMS utilizados

4.4.1 Acelerómetros

Un acelerómetro es un instrumento capaz de medir la aceleración rectilínea que experimenta. Esto lo puede hacer midiendo dos parámetros: los cambios experimentados por la fuerza gravitatoria y la aceleración. Estos parámetros los consigue midiendo una masa inercial que tienen en su interior.

La aceleración es lo que experimenta un objeto al cambiar de velocidad y el intervalo de tiempo en que esto ocurre. La aceleración es vectorial, es decir, tiene una dirección y es capaz de indicarnos la dirección en la cual acelera el objeto. Para esto, los acelerómetros tienen 1,2 o 3 ejes.

Los acelerómetros que solo tienen 1 eje, pueden medir la aceleración en una dirección del espacio vectorial. Los que tienen 2 ejes, son capaces de medir la

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

aceleración en un plano. Y los que tienen 3 ejes (3-axis) pueden medir hacia todas las direcciones posibles.

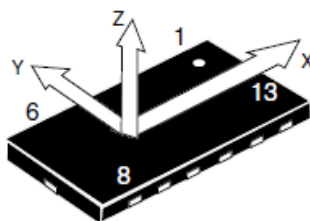


Figura 22 Acelerómetro de 3 ejes

Hay muchos tipos diferentes de acelerómetros, podemos tener acelerómetros basados en tecnologías capacitivas, piezoeléctricos y mecánicos.

Acelerómetros Basados en Tecnologías Capacitivas

Este tipo de acelerómetros se encargan de variar la posición relativa de las placas de un micro-condensador cuando éste se encuentra sometido a aceleraciones, es decir, se encargan de variar la capacidad entre dos o más conductores entre los cuales existe siempre un material dieléctrico.

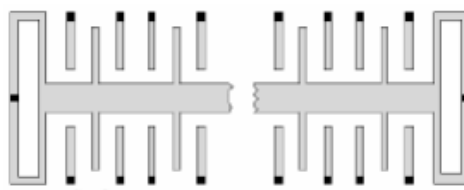


Figura 23 Placas de un sensor capacitivo.

En la figura 23 se observa cómo está formado internamente el acelerómetro capacitivo, el cual tiene un conjunto de capacidades fijas (ancladas al circuito) y por otro lado existe un conjunto de placas centrales que se encuentran unidas a unos anillos elásticos que permiten el movimiento de estas placas. Al moverse las placas, cambia la capacidad del condensador.

Acelerómetros Piezoeléctricos

El funcionamiento de estos acelerómetros viene regido por el efecto piezoeléctrico. Por tanto, al sufrir una deformación física del material se produce un cambio en la estructura cristalina y por consecuencia cambian sus características eléctricas.

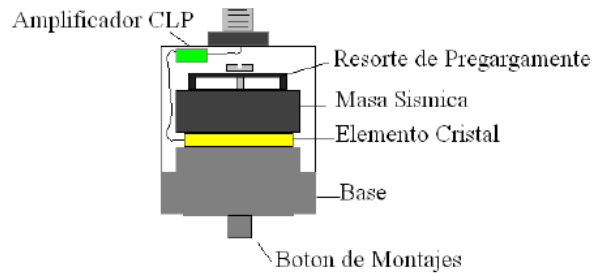


Figura 24 Acelerómetro Piezoeléctrico.

Estos tipos de acelerómetros tienen un problema y es que no permiten que para una entrada común se pueda mantener una señal de salida constante, y además hay que tener una frecuencia de trabajo baja.

Acelerómetros PiezoResistivos

Basan su funcionamiento en la propiedad que tienen las resistencias eléctricas de cambiar su valor cuando el material se deforma mecánicamente. Ese cambio depende del tipo de material y de cómo haya sido dopado. Los acelerómetros piezoresistivos usan una masa suspendida por cuatro semiconductores (también llamados *strain gages*). La masa es acelerada perpendicularmente a la superficie deformando de esta forma las galgas y cambiando así la resistencia en ellas de forma proporcional a la aceleración.

Acelerómetros Mecánicos

Estos acelerómetros utilizan bobinas, imanes para medir aceleración. Los componentes utilizados son una masa y resortes. Estos acelerómetros también son usados en sistemas rotativos desequilibrados.

4.4.2 Giroscopios

Los giroscopios son capaces de medir la inclinación angular de un movimiento, y el giro que origina éste.

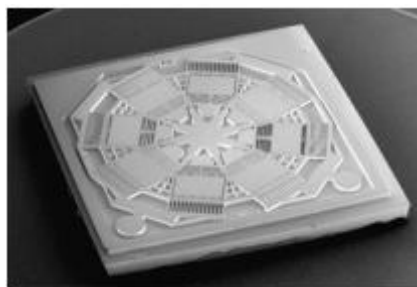


Figura 25 Giroscopio MEMS.

Los sistemas MEMS integran elementos mecánicos (ruedas, engranajes...), sensores, actuadores y elementos electrónicos en un mismo substrato de silicio. De este modo, los giroscopios basados en MEMS evitan los ruidos, fuerzas inerciales y fallos mecánicos típicos de los giroscopios clásicos basados en ruedas giratorias.

Los giroscopios de estructura vibrante se caracterizan por disponer de un elemento vibrante que, al ser rotado, es afectado con una fuerza de Coriolis que causa una segunda vibración perpendicular a la dirección de la vibración original. Los giroscopios de estructura vibrante se pueden implementar de diversas formas: giroscopios piezoeléctricos, giroscopios de tipo diapasón, giroscopios de rueda vibrante, giroscopios resonantes hemisféricos...

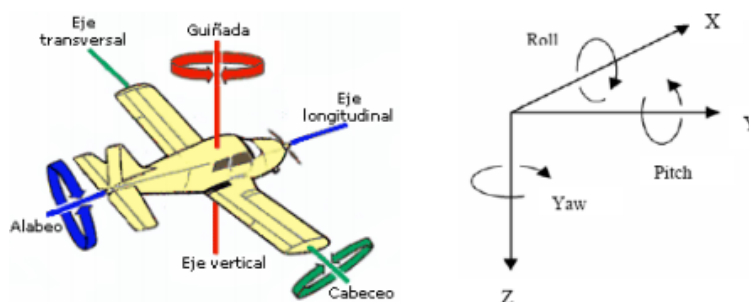


Figura 26 Sistema de Referencia Inercial.

En la figura 26 podemos observar como un giroscopio toma las medidas de cada eje de coordenadas, sirviéndonos para, por ejemplo, obtener los giros en el movimiento de un avión.

Normalmente son utilizados en la industria para medir los desplazamientos y la posición, y se implementan en dispositivos como los robots, navegadores GPS, dispositivos de videojuegos, teléfonos móviles, etc.

4.4.3 Barométricos

Un barómetro es un instrumento capaz de medir la presión atmosférica, es decir, la fuerza por unidad de superficie ejercida por el peso de la atmósfera. El peso del aire ejerce sobre la tierra una presión que es llamada "presión atmosférica". Este fenómeno fue descubierto por Evangelista Torricelli.

La presión atmosférica es variable, y dependiendo de la altura y de la humedad del aire varía:

- Variación de la altura: A mayor altitud menor presión. (A mayor altura, menos masa de aire existente).
- Humedad o sequedad del aire (El aire húmedo es menos pesado que el aire seco).

La unidad con que se mide la presión del aire se llama "Pascal", y normalmente se expresa como HectoPascal en las unidades atmosféricas.

1hP = 1 miliBar.

Un indicador de velocidad es en realidad y básicamente un medidor de diferencias de presión, que transforma esa presión diferencial en unidades de velocidad.

Los barómetros se pueden utilizar para medir la velocidad del aire indicada (AIS), muy importante en los aviones para controlar su rotación, saber el momento en el que puede levantar el vuelo en el despegue y comprobar si el ángulo de inclinación cuando va a aterrizar es demasiado pronunciado.



Figura 27 Avión Despegando.

Colocando dos barómetros podemos saber el ángulo de inclinación de un objeto, por ejemplo, en un avión, poniendo un barómetro en la cola y otro en la parte delantera. Ello nos indicará la altura a la que se encuentra éste y haciendo una operación de diferencias entre los dos barómetros, sabremos qué parte está más alta y cual está más baja.

4.4.4 Magnéticos

Los sensores magnéticos están diseñados para detectar de manera precisa la dirección y magnitud de los campos magnéticos externos, bien sea el campo magnético terrestre, o un campo magnético creado a partir de un imán o circuito magnético (Bobina, interruptores magnéticos, etc.).

El campo de aplicación de los sensores magnéticos es amplio, pudiéndose usar tanto para las unidades de telemetría, como para el automatismo de máquinas e interruptores, al poder pasar la señal magnética a través de varios obstáculos que no sean magnéticos.

Con un sensor magnético de 3 ejes, podemos hacer una brújula que nos indique los polos de la tierra, pudiendo así ser utilizados para la orientación de la ubicación.

Los sensores magnéticos más usados son los de tipo Hall y los transformadores lineales variables:

Tipo Hall

Consisten en la producción de una caída de voltaje a través de un conductor o un semiconductor con corriente, bajo la influencia de un campo magnético externo. Para ello es necesario que la dirección del campo sea perpendicular a la dirección del flujo de la corriente.

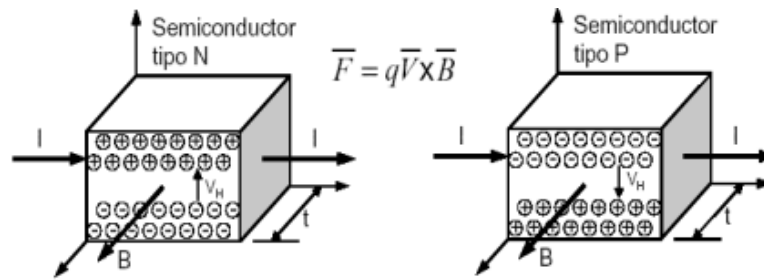


Figura 28 Sensor Hall.

El campo magnético actúa sobre las cargas positivas y negativas, desplazándolas y, al crear una acumulación de cargas, se origina una diferencia de potencial entre las caras opuestas. Este desequilibrio variará dependiendo de la fuerza del campo magnético.

Transformadores lineales variables

Proporcionan una lectura de la posición, usando la inductancia mutua entre dos bobinados. Un núcleo magnético móvil acopla el voltaje de excitación en corriente alterna a los dos secundarios. La fase y la magnitud del voltaje secundario varían de acuerdo con la posición del núcleo.

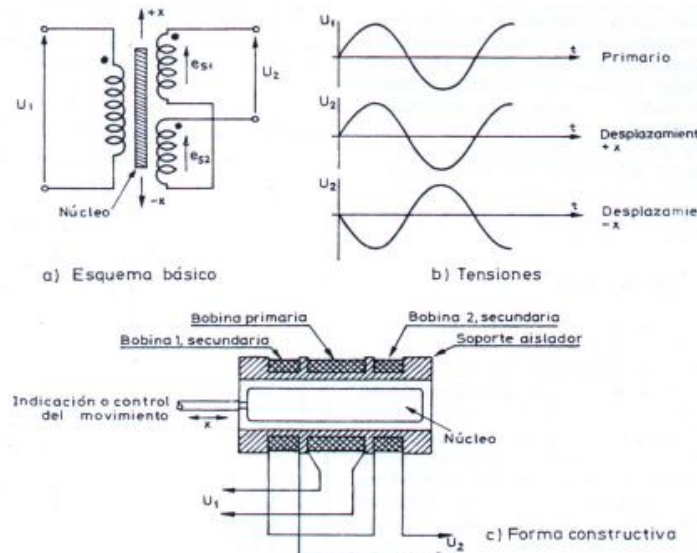


Figura 29 Transformador Lineal Variable.

Cuando el núcleo está en medio de los bobinados, los voltajes de ambos están 180 grados desfasados y son de igual magnitud, por lo que el voltaje neto es cero. Cuando el núcleo se mueve hacia la escala positiva o negativa, la señal en fase con la onda de entrada crece y viceversa, obteniendo un voltaje positivo o negativo.

5.0 COMPONENTES USADOS EN EL DISEÑO

En este apartado hablaremos de lo que se nos pedía para realizar el proyecto y de los materiales que se han utilizado para realizar el diseño de nuestra unidad de telemetría.

5.1 Requisitos del proyecto

Los requisitos del proyecto se definieron al empezar este proyecto fin de carrera. Se concretó que el tamaño de nuestra PCB debería de ser lo más pequeño posible para que ocupara poco espacio y pudiera ser acoplada en cualquier vehículo.

Otro requisito fue el que se eligieran los componentes más económicos. Con esto nos referimos a elementos tales como condensadores, resistencias y otros dispositivos de los que existe una gran variedad en el mercado. Generalmente, el precio dependerá de la marca. Los demás requisitos, los detallaremos en los siguientes apartados.

5.1.1 Alimentación

Para alimentar nuestra unidad de telemetría, se decidió utilizar un voltaje de 3.3v. Este voltaje es el que se utiliza en muchos equipos informáticos y equipos electrónicos de esta clase.

También se tuvo en cuenta la posibilidad de alimentarlo con baterías, pero se finalmente se decidió utilizar la alimentación que nos proporcionaba el USB (5v) y utilizar un regulador de tensión para bajar de los 5v hasta los 3.3v que utilizamos.

Se desechó la utilización de baterías, debido a que queríamos una placa de reducidas dimensiones, la cual se pudiera instalar en sitios pequeños, como un helicóptero teledirigido.

5.1.2 Sensores inerciales (acelerómetros y giroscopio)

Como nuestra idea era conocer los movimientos en donde estuviera instalado este sensor, se planteó la posibilidad de utilizar un sistema de acelerómetros y otro de giroscopio para poder conocer la aceleración con la que se desplazaba el objeto y el movimiento angular que podría tener.

Para ello necesitábamos un sensor que nos pudiera proporcionar los 6 ejes, tres ejes del acelerómetro y tres del giroscopio. Pero como no existe un IC en los que vengan los 6 ejes, se instalaron dos integrados que hicieran este trabajo.

Se eligió la opción de colocar los seis ejes porque con ellos podíamos ver cualquier movimiento lineal y angular del objeto en el que se encuentre nuestra placa. Y eso precisamente era lo que necesitábamos.

5.1.3 Sensores Barométricos

Se planteó utilizar dos sensores barométricos, para utilizarlos a modo de altímetro, es decir, utilizar la diferencia de presión atmosférica que existe entre las diferentes alturas, para saber a qué altura estaba nuestro objeto.

Como nuestra placa PCB está pensada para ir dentro de un avión o un helicóptero, utilizaríamos los barómetros para saber la inclinación que tendría nuestro aparato, ya que colocaríamos un sensor en la parte delantera del mismo y otro en la parte trasera. De esta manera, sabríamos cual de los dos barómetros nos da menos o más altura y sabríamos la inclinación que tendría nuestro avión o helicóptero.

En un avión es muy importante saber la inclinación que tiene, pues los aviones utilizan el sistema AIS (velocidad del aire indicada) para realizar los despegues y aterrizajes. Hasta que no tienen una cierta velocidad, no se pueden inclinar para empezar a despegar, y del mismo modo, para aterrizar necesitan

saber la inclinación que tienen para tocar tierra de manera que el impacto sea lo menos notable posible.

5.1.4 Sensor Magnético

Se decidió poner un sensor magnético para conocer el rumbo que tomaría nuestro avión o helicóptero. Para ello utilizaríamos un integrado que fuera capaz de medir campos magnéticos.

El principal campo magnético que tenemos nosotros es nuestra tierra, por lo tanto nuestro sensor sería capaz de detectar el norte magnético si ningún objeto magnético se cruza en nuestro camino.

En un principio se pensó en utilizar el sensor magnético HMC5843 de Honeywell llamado 3-axis digital compass, pero debido a que dejaron de fabricar este dispositivo, se tuvo que cambiar de integrado, eligiendo otro más moderno y mejor.

5.1.5 Comunicación inalámbrica

Otro factor que tuvimos en cuenta fue poner alguna comunicación en la que no tuviera que estar cableado el dispositivo para comunicarse, ya que nuestra PCB estaría en un objeto móvil y necesitábamos algo inalámbrico.

La mejor opción de las planteadas fue implementar un radiotransmisor en la banda de los 868MHz, debido a que este tipo de transmisión consume considerablemente poco, y sin necesidad de amplificadores es capaz de alcanzar unos 20m con un solo integrado.

Se eligió la banda de los 868 MHz en vez de la banda universal de 2.4 GHz, porque para esta longitud de onda se acorta la distancia en la que es capaz de transmitir, debido a que las ondas a esta frecuencia son más débiles que las ondas a menor frecuencia.

Además se optó por esta banda, pues sólo necesitábamos enviar poca cantidad de datos; a diferencia de la banda de 2.4 GHz empleada para transmitir una mayor cantidad de información a menor latencia.

5.1.6 Conectividad USB

A parte de necesitar comunicación vía radio, es decir, inalámbricamente, se llegó a la conclusión de que debíamos incluir también comunicación con un equipo informático. Por lo tanto, se añadió a nuestro proyecto un puerto USB, el cual a través de un puerto serie virtual, ofrece la posibilidad de comunicarse con un PC.

Se eligió que fuera un conector mini-USB para que nos ocupara un espacio reducido en nuestra placa y así conseguir el objetivo que nos habíamos fijado desde el principio del proyecto, es decir, que la placa tuviera unas dimensiones reducidas.

Se optó por el dispositivo CP2102 debido a que éste hace de interface entre el puerto USB de nuestro PC y el puerto USART de nuestro microcontrolador.

5.2 Alimentación (TPS60500)

Para alimentar nuestro dispositivo se utilizó el regulador de tensión TPS60500. Este regulador puede ser configurado para obtener tensiones de 3.3v, 1.8v, 2.5v o regulable desde cualquier tensión entre 1.8v a 6.5v.

Además, es capaz de entregar a su salida hasta 250mA de corriente. Ésta es más que suficiente para alimentar a todos nuestros dispositivos. Al no necesitar tanto suministro de corriente, se diseñó para que entregara 150mA a su salida, permitiendo así un ahorro en la alimentación.

Con este dispositivo, logramos dos objetivos importantes que son:

- Obtener una eficiencia de hasta el 90%, dependiendo de las alimentaciones de entrada y de salida.
- Integrar toda la alimentación de nuestro sistema en un solo chip, de dimensiones considerablemente pequeñas, ocupando éste muy poco espacio en nuestra placa y logrando así una de las características principales que se nos exigía.

El TPS60500 incluye una protección frente a cortocircuitos y a excesos de temperatura. Si la tensión es inferior a 0.6v, limita el consumo de corriente a 300mA. Por otra parte, si nuestro chip alcanza una temperatura superior a los 160°C, corta la corriente a 0mA.

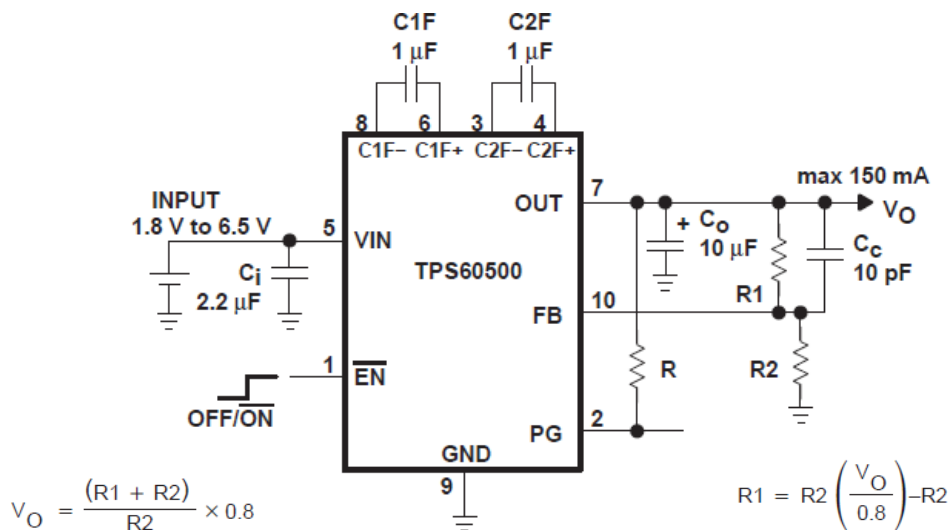


Figura 30 Regulador de tensión.

Mediante el esquema de la figura 30 y las fórmulas, obtenemos la tensión que deseamos en la salida V_o , regulando las resistencias $R1$ y $R2$. Fijando una de las dos resistencias a un valor concreto, podemos descubrir el valor que necesitamos de la otra resistencia.

Como también se puede apreciar en la figura 30, el TPS60500 tiene la posibilidad de apagarse o encenderse dependiendo si hay o no hay tensión en la patilla /EN.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

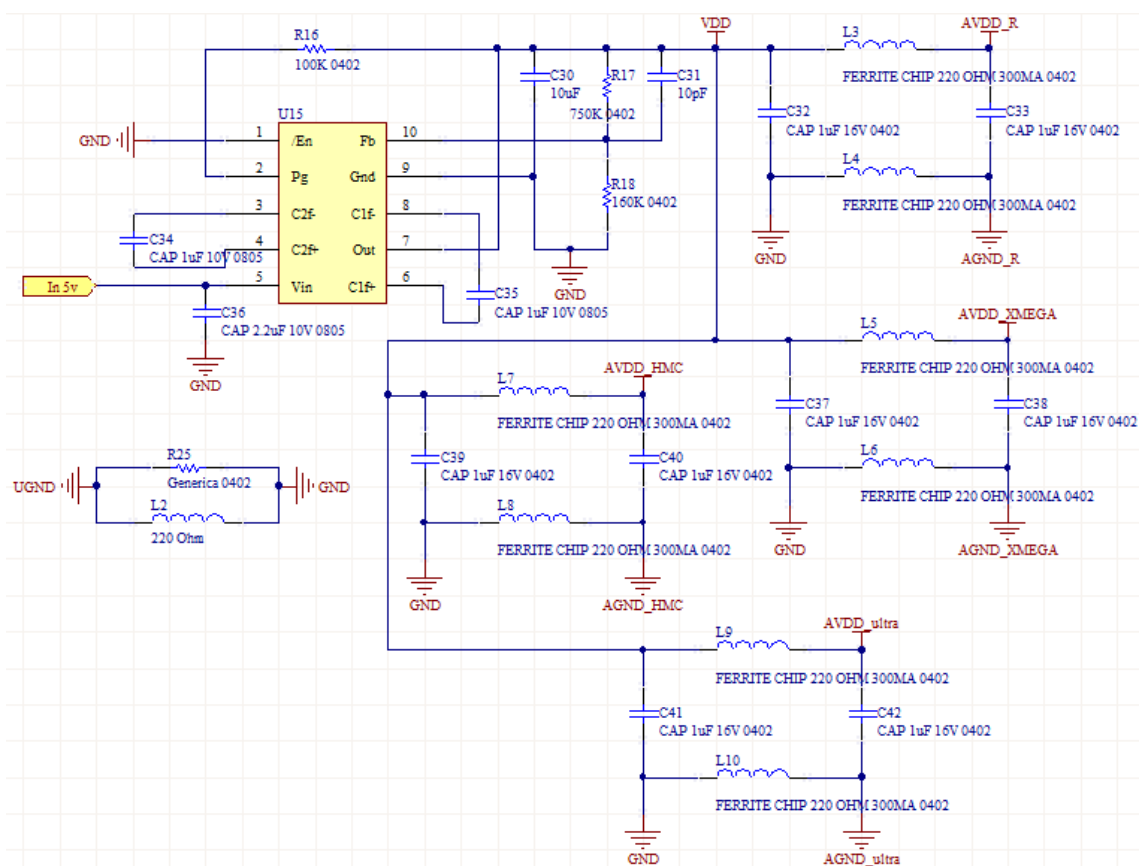


Figura 31 Diagrama de Aplicación del TPS60500.

En nuestro diseño, se utilizó la salida de nuestro regulador para a partir de la tensión digital que se conseguía, obtener las señales analógicas para los dispositivos que necesitaban una señal más estable y precisa.

Para lograr pasar una alimentación digital a una analógica, se utilizaron las Ferrite chips (bobinas) de 220Ω , y condensadores de $1\mu\text{F}$. Las Ferrite chip se colocaron en serie con la tensión positiva y en serie con la tensión negativa, mientras que los condensadores se conectaron en paralelo.

Con esta configuración, logramos una señal analógica muy estable, evitando así picos de tensión y ruidos en la alimentación, indeseables en cualquier diseño electrónico.

La alimentación V_{dd} se utilizó para suministrar tensión a los dispositivos que usaban alimentación digital y no tenían problemas con el ruido o la inestabilidad que tiene esta señal. Por otra parte, para el resto de dispositivos, esto es, el

microcontrolador, la radio, el sensor magnético, y el sensor ultrasónico, se implementó una señal analógica para cada uno de ellos por separado.

5.3 Microcontrolador (ATXMEGA128A3)

El microcontrolador es el dispositivo más importante en el diseño de cualquier aplicación, ya que es éste el encargado de realizar todas las funciones de cálculo, codificación, conversión, activar o desactivar señales, etc.

Nuestro micro es concretamente el ATXMEGAa128A3, que proviene de la familia XMEGA, la cual es bastante reciente en el mercado, siendo uno de los últimos controladores diseñados por ATMEL. El diseño AVR de ATMEL difiere de los demás microcontroladores de 8 bits en tener mayor cantidad de registros (32) y un conjunto ortogonal de instrucciones. AVR es mucho más moderno que su competencia.

ATMEL fabrica los microcontroladores de la familia AVR. Esta nueva tecnología proporciona todos los beneficios habituales de arquitectura RISC y memoria flash reprogramable eléctricamente. La característica que identifica a estos microcontroladores de ATMEL es la memoria flash y EEPROM que incorpora.

Gracias a que tiene un consumo reducido de potencia y a que es capaz de ejecutar las instrucciones en un solo ciclo de reloj, es uno de los mejores si comparamos su consumo con su velocidad de ejecución. Tiene 32 registros conectados directamente a la ALU, permitiendo que puedan acceder a la vez dos registros independientes a una sola instrucción y en un ciclo de reloj. Cuando una instrucción está empezando a ejecutarse, la siguiente instrucción ya se encuentra pre-buscada en la memoria del programa. Gracias a esto, es más eficaz y logra mayor rendimiento que los micros convencionales.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

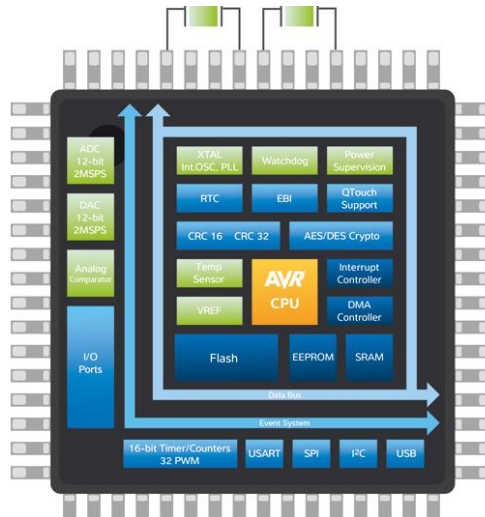


Figura 32 Esquema Xmega.

Para ahorrar energía, este micro permite que pueda ponerse en un modo en el que la CPU se desconecte, pero que continúen funcionando la SRAM, el control de DMA, las interrupciones y el control de los periféricos.

Para programarlo se usa el Program and Debug Interface (PDI), mediante el Jtag que es un estándar IEEE 1149.1. El Xmega soporta los compiladores C, Macro ensambladores, programas de depuración y simulación, etc.

Para acceder a la lectura o escritura de los registros, nuestro micro utiliza registros de 16 bits. Si se desea utilizar sólo 8 bits, se utiliza sólo la parte baja del registro.

Nuestro controlador también cuenta con el registro de estado SREG. Gracias a éste, podemos conocer todos los sucesos que ocurren al trabajar con los registros.

Bit	7	6	5	4	3	2	1	0	
+0x0F	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 33 Registro de Estado (SREG)

5.3.1 Diagrama de Bloques

En la siguiente figura podremos observar todas las conexiones entre los bloques del micro. Asimismo, se aprecian las entradas de los puertos y los diferentes tipos de conexiones a cada puerto.

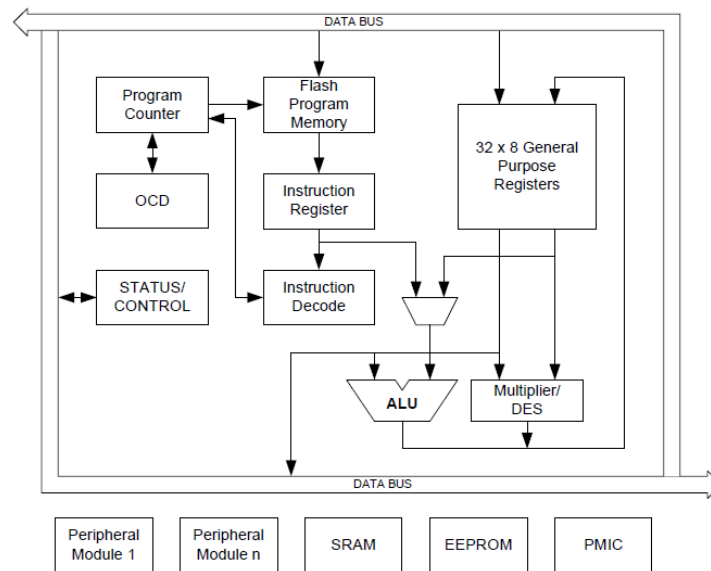


Figura 34 Arquitectura del Xmega.

Se puede observar cómo los registros de propósito general del micro están conectados directamente a la ALU, logrando así una mayor rapidez y eficiencia.

En la siguiente figura 35, vemos todos los bloques del microcontrolador y las funciones que tienen sus puertos.

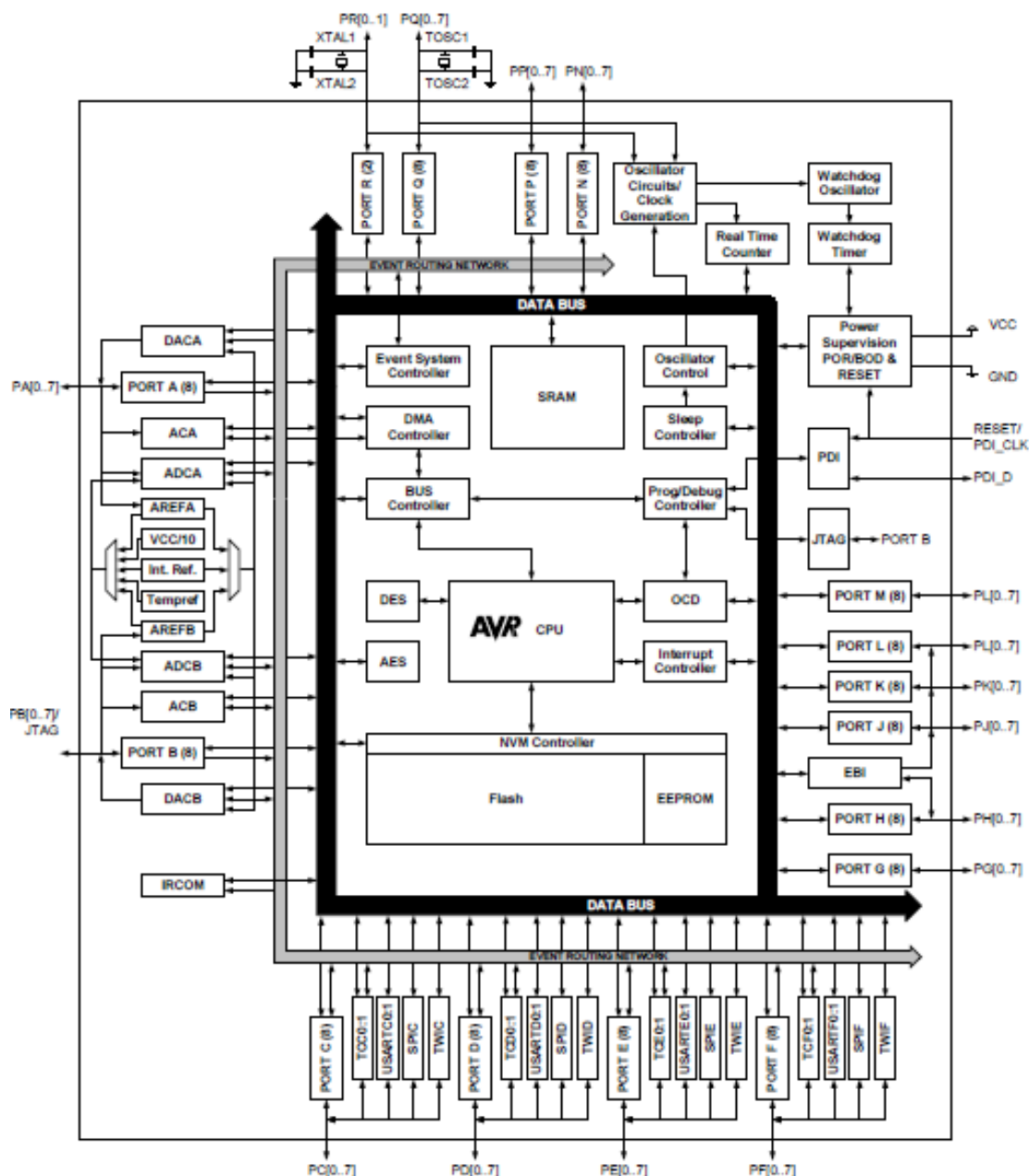


Figura 35 Diagrama de Bloques del microcontrolador.

5.3.2 Características Principales

El XMEGA elegido consta de 64 pines con 7 puertos. Los puertos desde el PA hasta el PF son de 7 pines y se pueden configurar de diferentes funciones (SPI, USART, TWI, ADC, DAC, etc), mientras que el puerto PR es especial con 4 pines para conectar un reloj externo y el programador PDI.

Las características principales son:

- Cuatro canales con control DMA
- Siete Timer/Counter de 16bits
- Siete buses USART
- Dos buses I2C
- Tres buses SPI
- Dos convertidores ADC
- Un convertidor de DAC
- Cuatro comparadores analógicos
- Interrupciones Externas
- WhatchDog programable

De las prestaciones que tiene nuestro micro nosotros utilizamos los buses principales que son el SPI, la USART y el I2C. A continuación detallaremos cada uno de estos 3 buses.

5.3.3 Interrupciones

Las interrupciones en el micro son utilizadas para que el microcontrolador deje de hacer lo que está ejecutando en ese momento, y haga caso a algún dispositivo que necesita ser atendido. El dispositivo emite una interrupción y el micro va a tener en cuenta a dicho dispositivo de manera inmediata, parando la tarea que estaba realizando.

De esta manera evitamos que el microcontrolador tenga que estar preguntando de manera continuada a los dispositivos para averiguar si necesitan ser atendidos o no, perdiendo tiempo en esta operación.

Hay dos tipos de interrupciones: las interrupciones por software y las interrupciones por hardware.

Cuando se produce este hecho, el programa salta y se va al vector de interrupción propio de la interrupción que se produjo. El vector ha de contener un

salto al programa que realice la tarea de atender al dispositivo que provocó la interrupción.

Nuestro microcontrolador tiene tres niveles de interrupción, que son: alto, medio y bajo. Esto es así para dar prioridad a aquellas interrupciones que son más importantes y dar menos preferencia a otras que son menos urgentes.

Para atender una interrupción, la CPU espera cinco ciclos de reloj para ejecutarla. Después de los cinco ciclos, se atiende a la interrupción, saltando el vector de la interrupción.

Si se ejecuta una interrupción mientras nuestro micro está en modo Sleep, se ejecutan los cinco ciclos de reloj y se atiende a la interrupción. Cuando se produce otra interrupción mientras se está atendiendo a una, se termina la que estamos atendiendo y se ejecuta la siguiente cuando se termine.

Cuando la interrupción que ya ha sido atendida se ha terminado, el programa vuelve a donde estaba antes de producirse la interrupción; siguiendo así el programa normal que había dejado para atender la interrupción.

Cada interrupción tiene un flag de registro que se habilita cuando se produce. Si no queremos que se produzca una interrupción, podemos enmascararla para que el programa no haga caso a la misma.

5.3.4 Puertos

El XMEGA tiene la flexibilidad de utilizar el registro GPIO, que se encarga de almacenar datos de propósito general o variable que estén en los pines de los puertos en un espacio de memoria accesible. Un puerto se compone de un máximo de 8 pines que van desde el pin 0 a 7, donde se puede configurar cada pin como entrada o como salida.

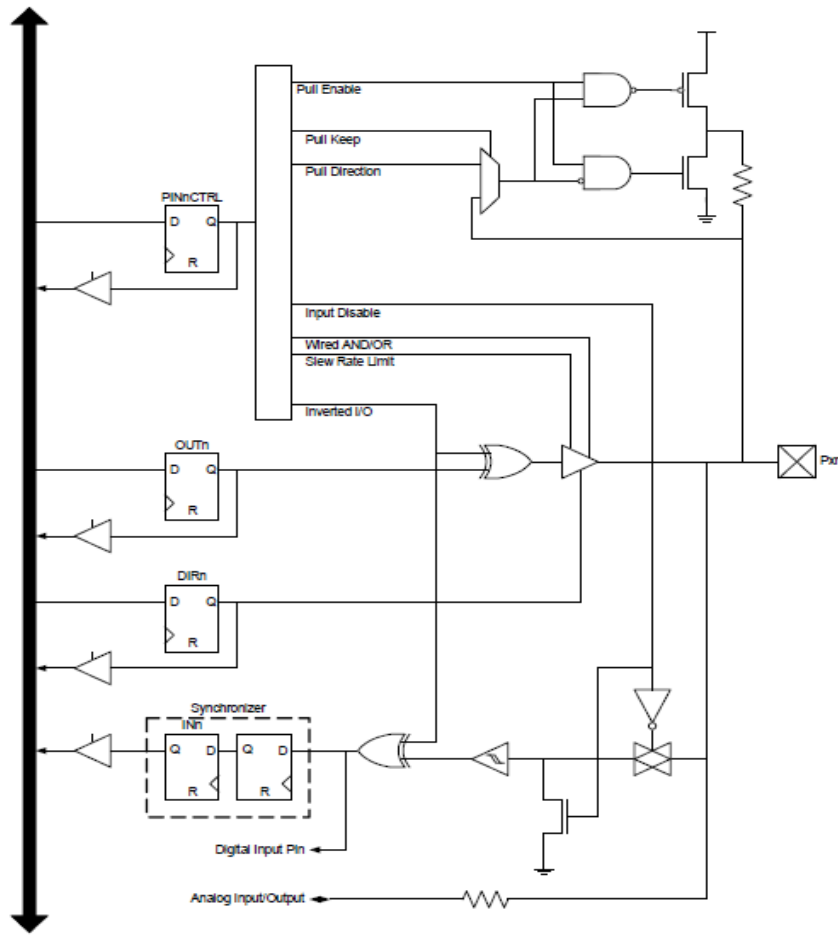


Figura 36 Funcionalidad general de los pines.

Cada pin es posible configurarlo por separado, bien como sólo entrada, sólo salida, entrada/salida, con pull-up, con pull-down. La configuración general de los pines la podemos ver en la siguiente figura.

Para configurar los pines se emplean los registros DATA DIRECTION (DIR) y DATA OUTPUT VALUE (OUT) para los datos de salida, y DATA IN VALUE (IN) para leer los datos de los pines del puerto.

Configuración de los puertos:

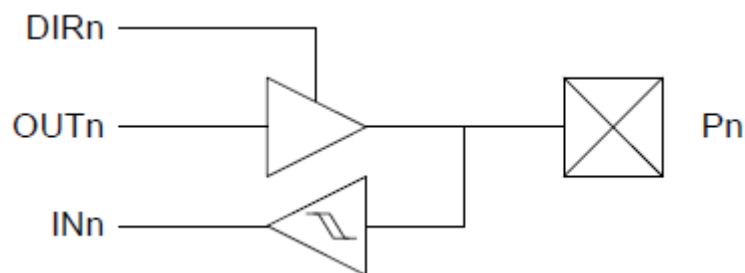


Figura 37 Push-Pull.

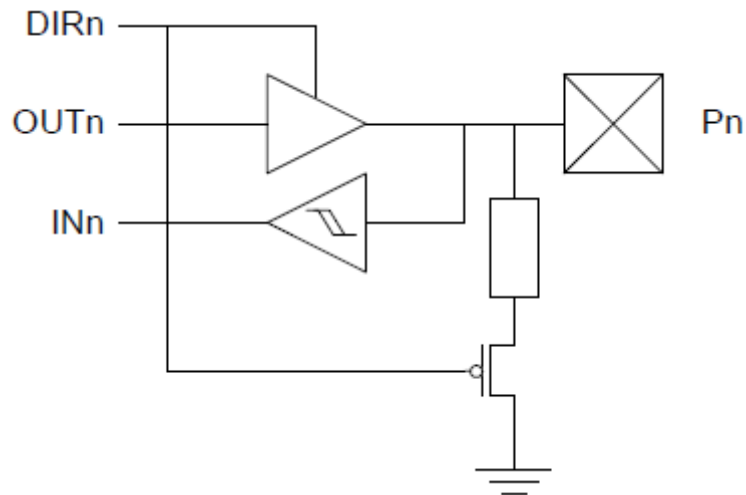


Figura 38 Pull-Down.

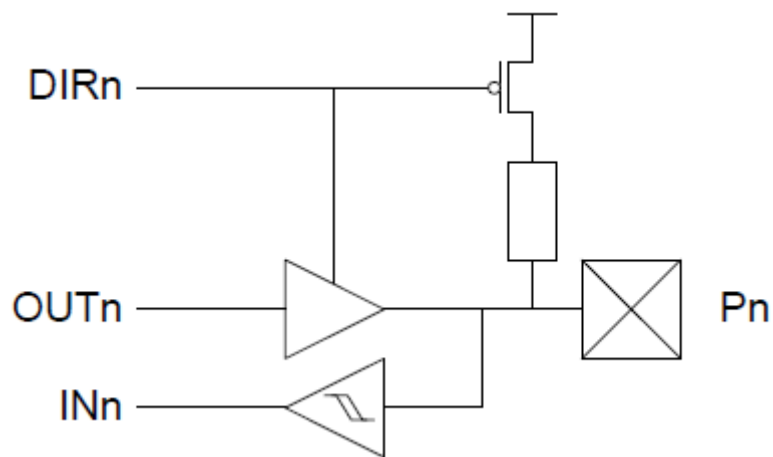


Figura 39 5 Pull-Up.

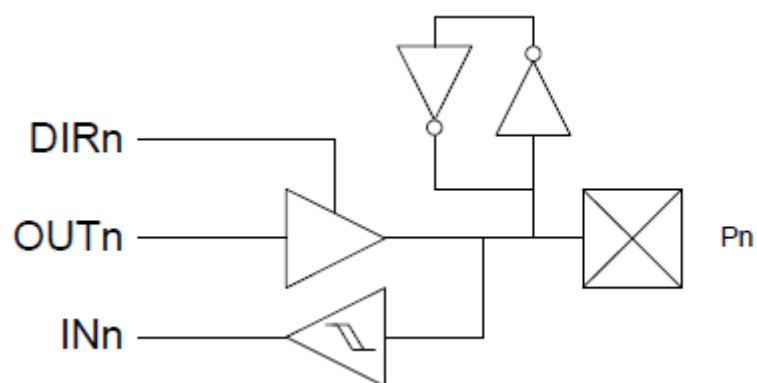


Figura 40 Bus-Keeper.

La configuración Keeper funciona como un Pull-Up si antes había un nivel lógico alto, y como un Pull-Up si antes había un nivel lógico bajo.

Cada pin tiene dos vectores de interrupción, uno síncrono y otro asíncrono. Y cada uno, a su vez, se puede configurar para que se active cuando detecte un flanco de subida/bajada, o un nivel alto o bajo.

5.3.5 Bus SPI

El bus SPI (Serial Peripheral Interface) es un bus muy utilizado en las comunicaciones series por los dispositivos, puesto que permite la comunicación full-duplex a alta velocidad, se pueden utilizar varios dispositivos en un solo bus, pudiendo transmitir y recibir con un solo puerto a varios terminales. Admite que los dispositivos puedan utilizar la comunicación Maestro-esclavo. El dispositivo que está como Maestro es el que inicia la transferencia de datos hacia el que está como esclavo.

El bus consta de dos registros de desplazamiento y un reloj. Los registros reciben y envían los datos, y el reloj es el que sincroniza las señales.

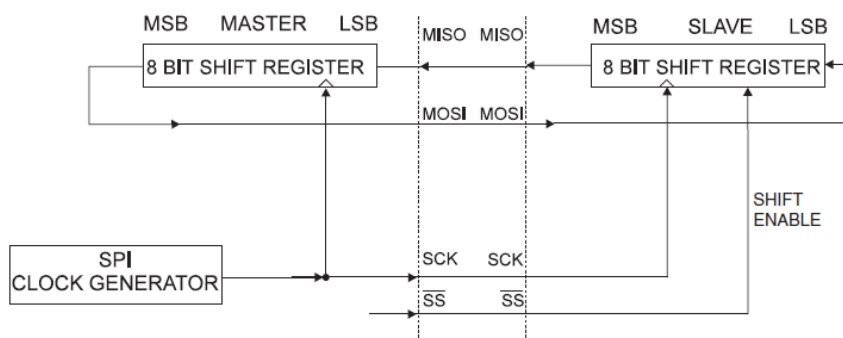


Figura 41 Interconexión Maestro-Esclavo del bus SPI.

Además, el bus tiene otra señal que es la línea /Ss y es la que inicia la transmisión de la señal cuando el dispositivo Maestro la activa.

Señal	Descripción
MOSI	Master Output – Slave Input, Salida del microcontrolador (Maestro) y Entrada en el sensor (Esclavo).
MISO	Master Input – Slave Output, Entrada al microcontrolador (Maestro) y Salida del dispositivo (Esclavo).
CLK	Señal de reloj, el Maestro puede configurar la frecuencia del reloj.
/Ss	Línea /Ss que se utiliza para seleccionar el dispositivo. Se activa a nivel bajo.

Tabla 3 Señales del Bus SPI.

Cuando el Maestro quiere comunicarse con el esclavo, realiza la siguiente operación:

1. Coloca el dato que desea transmitir en la línea Master Output – Slave In (MOSI).
2. Activa la línea /Ss del dispositivo al que desea transmitir.
3. El esclavo, al activarse la línea /Ss, sabe que va a recibir un dato y lo introduce en el registro de desplazamiento.
4. Al esclavo le entran los 8 bits por la línea MOSI. Luego, éste desplaza los 8 bits de su registro de desplazamiento por la línea Master Input - Slave Output (MISO), transmitiendo los datos que tenía al Maestro.
5. Cuando se han transmitido los 8 bits, el maestro desactiva la línea /Ss y es ahí cuando se da por terminada la comunicación.

El maestro configura el modo de transmisión que utilizará para comunicarse con el esclavo.

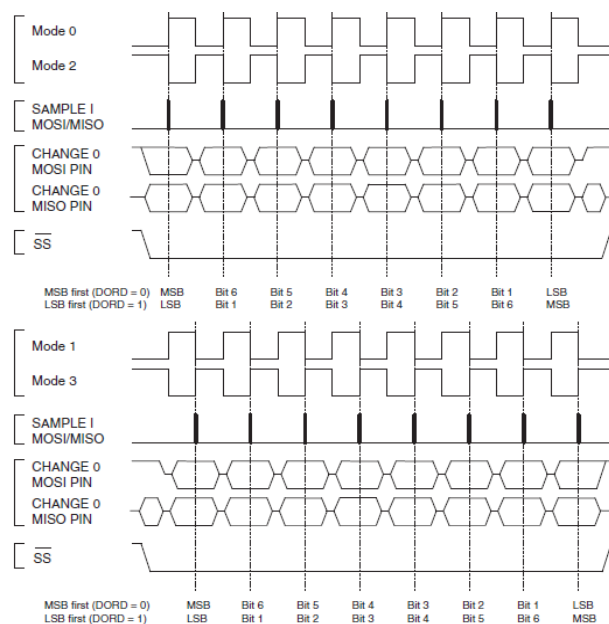


Figura 42 Modos de transferencia del SPI.

Hay 4 modos de transferencia, dependiendo del momento en que se detecta el flanco de la señal de reloj. Por ejemplo, una vez activada la señal /Ss en el Modo 1, el dato es recibido en el flanco de bajada del primer pulso de reloj, mientras que en el modo 0, se recibe el dato en el primer flanco de subida de la señal de reloj.

5.3.6 Bus USART

El bus Universal Síncrono y Asíncrono serie de Recepción y Transmisión (USART), permite la comunicación Full Duplex, la transmisión y la recepción síncrona y asíncrona, tamaño de las tramas de 5, 6, 7, 8 y 9 bits con 1 o 2 bits de stop.

Este bus permite una alta flexibilidad en lo que a transmisión serie se refiere, permitiendo cambiar el tamaño de la trama de datos, transmitir o recibir en modo continuo, e incluso usarse como un bus SPI.

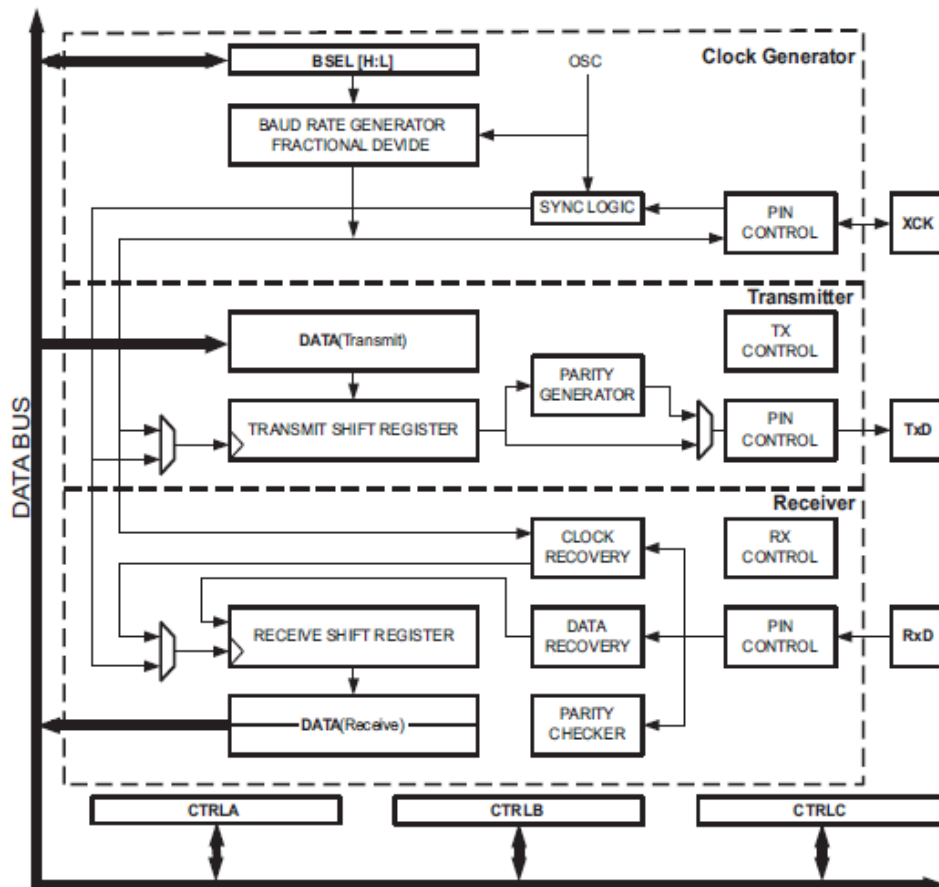


Figura 43 Diagrama de Bloques de la USART.

El reloj se puede configurar para que tenga un Baud Rate en el que se puedan comunicar los dos dispositivos. Este Baud Rate lo configuramos mediante los registros de control del micro. Mediante la tabla siguiente, podemos calcular los números que tenemos que introducir en los registros del micro.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

Operating Mode	Conditions	Equation for Calculation Baud Rate ⁽¹⁾	Equation for Calculation BSEL Value
Asynchronous Normal Speed mode (CLK2X = 0)	BSCALE ≥ 0 $f_{BAUD} \leq \frac{f_{PER}}{16}$	$f_{BAUD} = \frac{f_{PER}}{2^{BSCALE} \cdot 16(BSEL + 1)}$	$BSEL = \frac{f_{PER}}{2^{BSCALE} \cdot 16f_{BAUD}} - 1$
	BSCALE < 0 $f_{BAUD} \leq \frac{f_{PER}}{16}$	$f_{BAUD} = \frac{f_{PER}}{16((2^{BSCALE} \cdot BSEL) + 1)}$	$BSEL = \frac{1}{2^{BSCALE}} \left(\frac{f_{PER}}{16f_{BAUD}} - 1 \right)$
Asynchronous Double Speed mode (CLK2X = 1)	BSCALE ≥ 0 $f_{BAUD} \leq \frac{f_{PER}}{8}$	$f_{BAUD} = \frac{f_{PER}}{2^{BSCALE} \cdot 8 \cdot (BSEL + 1)}$	$BSEL = \frac{f_{PER}}{2^{BSCALE} \cdot 8f_{BAUD}} - 1$
	BSCALE < 0 $f_{BAUD} \leq \frac{f_{PER}}{8}$	$f_{BAUD} = \frac{f_{PER}}{8((2^{BSCALE} \cdot BSEL) + 1)}$	$BSEL = \frac{1}{2^{BSCALE}} \left(\frac{f_{PER}}{8f_{BAUD}} - 1 \right)$
Synchronous and SPI Master mode	$f_{BAUD} < \frac{f_{PER}}{2}$	$f_{BAUD} = \frac{f_{PER}}{2 \cdot (BSEL + 1)}$	$BSEL = \frac{f_{PER}}{2f_{BAUD}} - 1$

Tabla 4 Ecuaciones para el cálculo del Baud Rate.

En nuestro caso, utilizamos el modo de operación asíncrono normal con un BSCALE < 0 para lograr un Baud Rate de 115200 kb/seg.

$$Baud\ Rate = \frac{2.000.000}{16((2^{-7} * 11) + 1)} = 115.108 \frac{kb}{seg}$$

Éste es el Baud Rate más aproximado que alcanzamos modificando los registros BSCALE y BSEL de nuestro microcontrolador.

Para configurar el modo de sincronización que utilizaremos, debemos de configurar los registros UCPHA e INVEN.

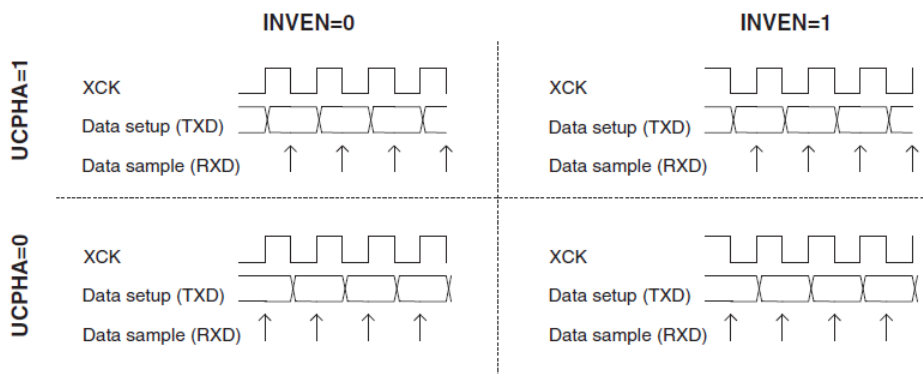


Figura 44 UCPHA y INVEN.

Como se puede observar en la figura superior, lo que hacemos cuando configuramos los registros UCPHA e INVEN es controlar el momento en el que se leen los datos dependiendo de la señal de reloj.

El formato de la trama es el siguiente:

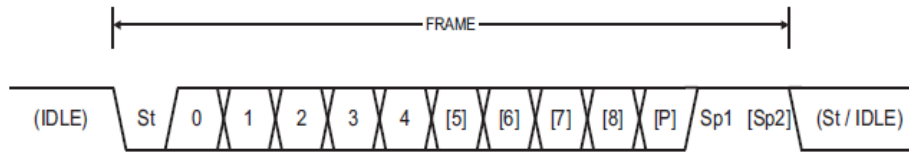


Figura 45 Formato de la trama USART.

St: Bits de comienzo, siempre abajo.

(n): Bits de datos (0 - 8).

P: Bits de paridad.

Sp: Bits de parada.

IDLE: no hay transferencia y debe estar siempre alto.

El bit de paridad puede configurarse para que nos dé el número de “1” que hay en la trama de datos (EVEN) o para que nos informe del número de “0” (ODD).

La USART se inicializa de la siguiente manera:

1. Se fija el pin de TX a valor alto, y opcionalmente el pin del reloj a nivel bajo.
2. Se configura la velocidad de transmisión y recepción.
3. Se fija el formato de trama que se va a utilizar.
4. Se habilita el modo de operación.
5. Se habilita la transmisión y/o recepción, dependiendo de qué se va a utilizar.

Transmisión

Para realizar una transmisión se escribe un dato en el registro del buffer (DATA), que es un registro de desplazamiento, para ser enviado. Al escribirse en este registro, el micro comprueba que el bus esté vacío (IDLE) y que puede enviar el dato; si esto no es así, el micro espera a que la línea de transmisión esté disponible. Una vez esté disponible el canal, el micro envía el dato y completa la

transmisión activando un flag (TXCIF). La transmisión se completa cuando están todos los datos enviados y no queda ninguno en el buffer.

Sólo se podrá escribir en el registro DATA si el flag de registro vacío (DREIF) está fijado. Ello significa que el bus está preparado para enviar un nuevo dato.

Cuando se deshabilita la transmisión, y el buffer de transmisión no contiene datos a transmitir, el pin TX se fija como entrada.

Recepción

La recepción de un dato comienza cuando se detecta un bit de comienzo válido. Esto permite que se reciban en el registro de desplazamiento los bits a la velocidad del reloj y se paran en la primera señal de parada (stop).

Cuando el stop es recibido y la trama completa se encuentra en el registro de desplazamiento, el contenido del registro es movido al buffer de recepción. Al pasar esto, se activa el flag (RXCIF) que indica que se ha completado la recepción.

Cuando se usan menos de 8 bits, los menos significativos dentro de la trama son tomados como "0". Mientras que cuando se utilizan tramas de 9 bits, el noveno bit debe ser leído en RxB8 antes de leer el registro DATA que contiene los 8 bits.

Flags

Para tener información de si se reciben o no los datos correctamente, el micro dispone de una serie de flags que se activan cuando tenemos algún error en la trama (FERR), cuando hay Overflow del buffer (BUFOVF) y cuando no coincide la paridad (PERR). Estos flag se leen en el registro de estado.

Si utilizamos el modo asíncrono, la lógica de recuperación de reloj se utiliza para sincronizar las tramas serie asíncronas entrantes en el pin RxD a la frecuencia de reloj generada internamente en baudios.

Las muestras de recuperación de datos, utilizan un filtrado bit a bit de la trama que hace mejorar la recepción frente a ruidos del receptor. El rango de recepción operativa asíncrono depende de la precisión del reloj interno, de la tasa de baudios y de la tasa y tamaño de las tramas de entrada (tamaño y número de bits).

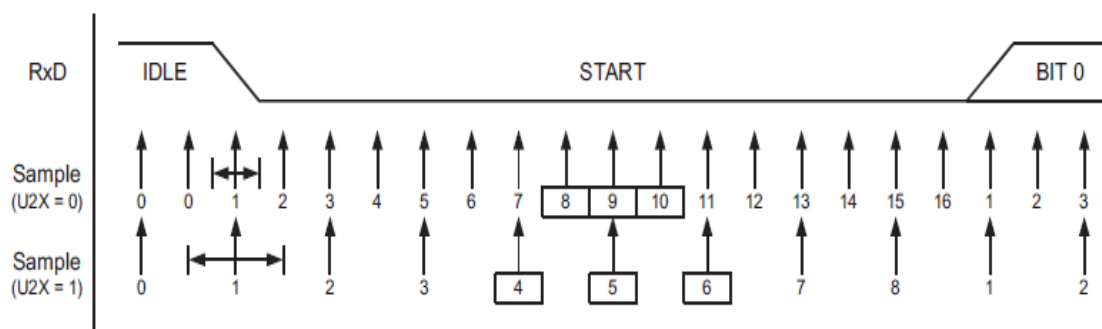


Figura 46 Ejemplo de la trama de comienzo en modo asíncrono.

Cuando la lógica de recuperación de reloj detecta que pasa de IDLE a Start, comienza la secuencia de inicialización. Se utiliza el U2X = 0 si se está realizando la comunicación a una velocidad simple, o se utiliza el U2X = 1 si se está haciendo la recepción a una velocidad doble. Sabiendo ya la velocidad con la que se está transmitiendo, el dispositivo se sincroniza sin utilizar la línea de reloj.

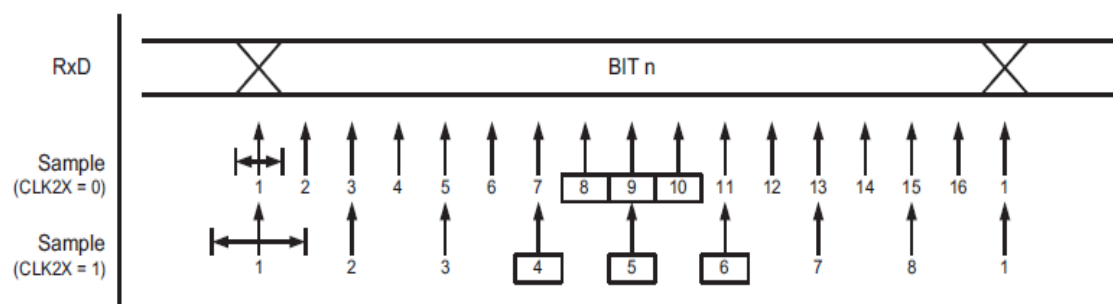


Figura 47 Proceso de recuperación de datos.

Como ya está sincronizado, el proceso es igual al anterior, filtrándose la información que hay en cada flanco de reloj interno. Si no coincide la distancia de la trama y se encuentra el bit de stop antes de los ciclos debidos al reloj, se activa el flag de error.

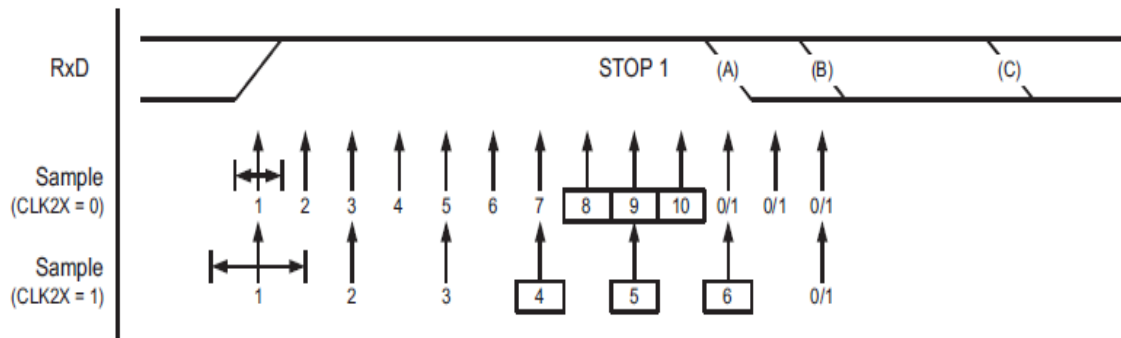


Figura 48 Ejemplo del bit de Stop y del siguiente bit de Start.

Cuando la línea de nuevo está a nivel alto, se vuelve a quedar el sistema esperando la llegada de un nuevo bit de comienzo, para que la línea se ponga a nivel bajo.

El rango de operación en este modo depende de cuál sea el tipo de transmisión/recepción que utilizamos, calculándose mediante las siguientes fórmulas:

$$R_{SLow} = \frac{(D + 1)S}{S - 1 + D * S + S_F}$$

$$R_{Fast} = \frac{(D + 2)S}{(D + 1)S + S_M}$$

D: Suma de caracteres de datos y paridad (D=5 a 10 bits).

S: Muestras por bit, S=16(Modo normal), S=8(Modo doble velocidad).

S_f: Primer número de muestras, S_f=8(Modo normal), S_f=4(Modo doble).

S_m: Número de muestras utilizadas, S_m=9(Modo normal), S_m=5(Modo doble).

R_{slow}: Radio más bajo que puede ser aceptado en relación con el Baud Rate.

R_{fast}: Radio más rápido que puede ser aceptado en relación con el Baud Rate.

5.3.7 Bus I2C

El bus I2C es un bus de comunicaciones que utiliza 2 cables para transportar la información desde el Maestro hasta el Esclavo.

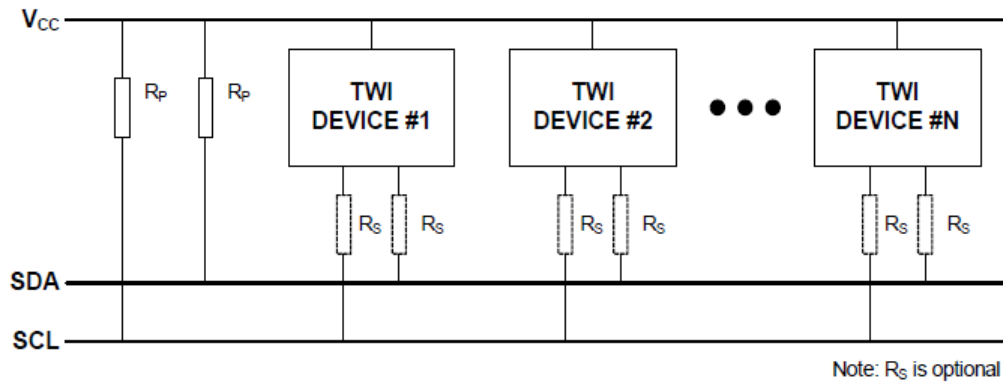


Figura 49 Topología del Bus I2C.

Utiliza la señal SDA para enviar y recibir los datos, al ser una línea bidireccional. Y la señal SCL es la señal de reloj, que puede alcanzar velocidades de entre 100KHz y 400KHz.

Las dos líneas son del tipo colector abierto, y dependiendo de la configuración que utilicemos del micro, son necesarias u opcionales.

Para poder colocar los dispositivos sobre un único bus I2C se utiliza un sistema de direcciones, en el que cada dispositivo tiene una dirección de 7 o 10 bits, pudiéndose colocar hasta 127 o 1024 dispositivos I2C en un solo canal.

Para inicializar la transacción, el Master envía primero la condición de comienzo (S) y luego envía la dirección del dispositivo con el que quiere comunicarse indicando si es un acceso de lectura o de escritura (R/W).

El dispositivo al que pertenezca la dirección de comienzo se quedará a la escucha de los datos, mientras todos los demás que estén en el bus, se desactivarán de la comunicación, hasta que se produzca la siguiente condición de comienzo (S).

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

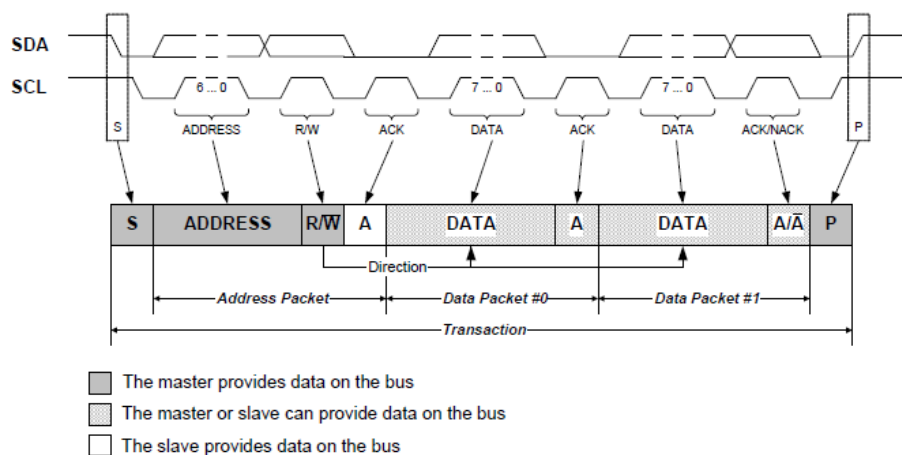


Figura 50 Modo de Transacción del bus I2C.

Por último, los datos (DATA) son transferidos y una vez acabados de transmitir, el Master envía la señal de STOP (P). Cuando se recibe un dato, el esclavo tiene que enviar una señal de reconocimiento (A) o de no reconocimiento (/A) de cada byte recibido.

5.4 Radio (AT86RF212)

Como ya mencionamos, nuestra PCB contiene un enlace inalámbrico para poder cubrir las necesidades de comunicación sin tener que estar fijada o llegar hasta la longitud del cable de comunicación.

Por lo tanto, se decidió utilizar el integrado AT86Rf212 para poder comunicar nuestra placa a un nodo de comunicación inalámbrica.

5.4.1 Características

La radio se ha implementado para poder transmitir vía aérea la información que deseemos, y así dotar a nuestro dispositivo de mayor libertad de movimiento, sin necesidad de un sistema cableado para poder recibir los datos y la información que nos den nuestros sensores.

Según el datasheet de nuestro dispositivo, el AT86RF212 es un conversor SPI en una antena de comunicaciones, es decir, los datos que recibe por el SPI

los modifica y los prepara para enviarlos vía radio a otro dispositivo. Nuestro chip incluye todas las soluciones para transmitir datos, sin incluir la antena, el cristal y los condensadores.

En nuestra transmisión utilizamos el estándar IEEE 802.15.4, y concretamente se transmitirá en la banda de 868MHz, que es la banda europea. Se ha elegido esta frecuencia y este chip porque nos brinda excelentes prestaciones para lo que nosotros necesitamos, que es enviar señales de baja tasa de datos y a una distancia de algunos metros con muy poco consumo de energía.

El AT86RF212 soporta el estándar IEEE 802.15.4-2006 en el que además de modular la señal en BPSK, tiene la opción de poder modular en O-BPSK. Y tiene otras características como son 128 bits de seguridad, y el estándar de seguridad EAS de encriptación/desencriptación.

Nuestro dispositivo soporta una temperatura que oscila desde los -40°C hasta los 85°C, rango muy amplio para poder cubrir todas las condiciones ambientales del terreno en el que esté operando, sin problemas de calentamiento o tener que implementar dispositivos refrigerantes.

Además, opera con una alimentación entre 1.8v - 3.6v, un rango adecuado para la alimentación por baterías o por algún dispositivo que nos proporcione un voltaje dentro de estos límites. En el mercado existen diversos dispositivos que son capaces de ello, puesto que dichos voltajes se han estandarizado en la industria de los equipos eléctricos.

5.4.2 Diagrama de bloque

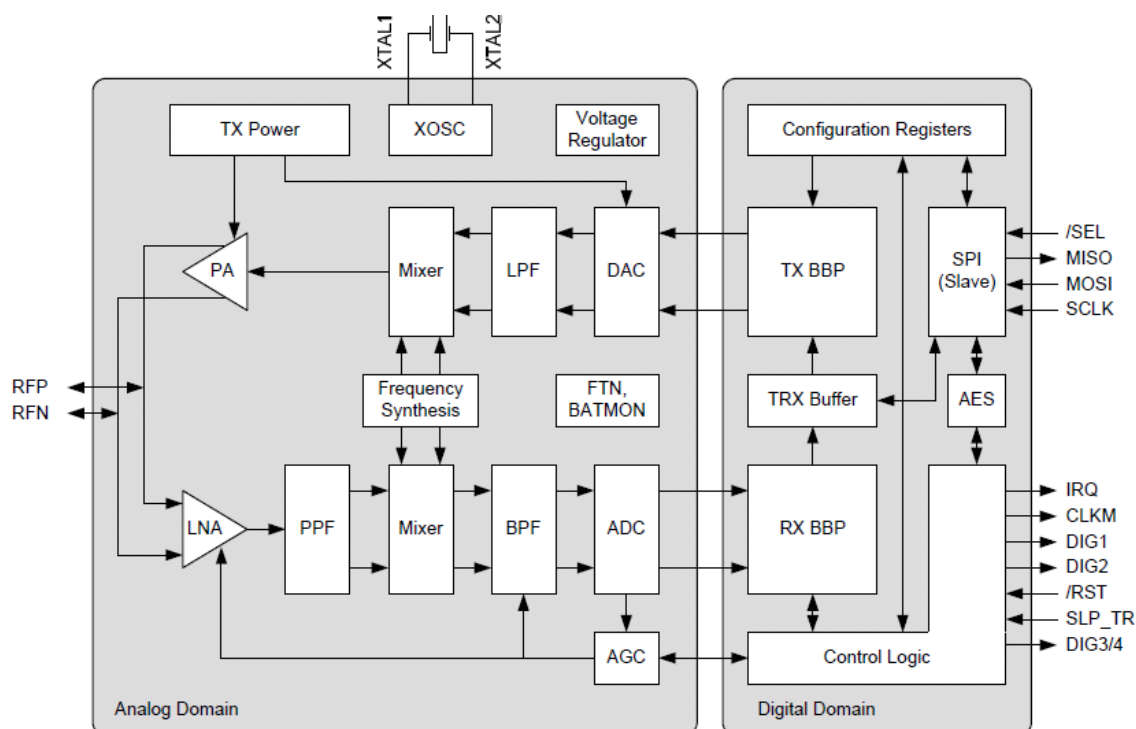


Figura 51 Diagrama de bloques del AT86Rf212.

Como podemos apreciar en el diagrama de bloques, nuestro dispositivo se comunica con nuestro controlador por medio del SPI (/SEL, MISO, MOSI y SCLK), y por medio de otras señales de control (IRQ, CLKM, DIG1, DIG2, /RST, SLP_TR y DIG3/4).

Y internamente contiene todos los bloques necesarios para la conversión de la señal adecuándola a ser transmitida/recibida. Teniendo los registros de configuración y de control, los convertidores de digital a analógico y viceversa, los filtros y los mezcladores de frecuencia, y incluso un amplificador para la señal de salida.

5.4.3 Diagrama de conexión

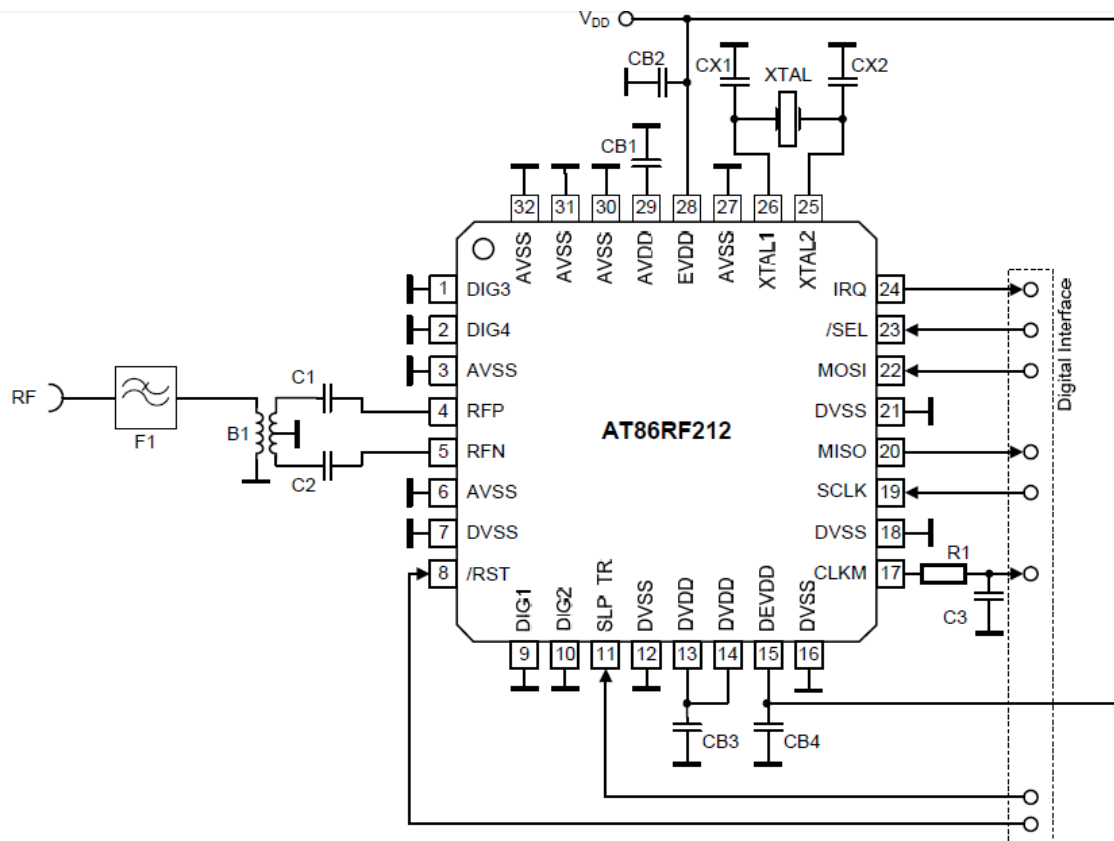


Figura 52 Circuito de aplicación del AT86Rf212.

En la figura superior se puede ver cómo está conectado nuestro AT86Rf212, con los condensadores específicos (Cx y CBx) que nos pide el datasheet del fabricante, con el cristal externo (XTAL) que nos da una frecuencia de 16Mhz, con el filtro o Balun (B1 y F1) a la salida de nuestra señal y nuestro conector de antena (RF).

Es decir con todos los dispositivos externos que necesitamos para la transmisión/recepción de datos, y que el dispositivo AT86Rf212 no incluye.

5.4.4 Conexiones al microcontrolador

Como ya se ha mencionado, la radio se conecta al controlador por medio del puerto SPI. Atraves de este puerto enviaremos y recibiremos los datos de la radio.

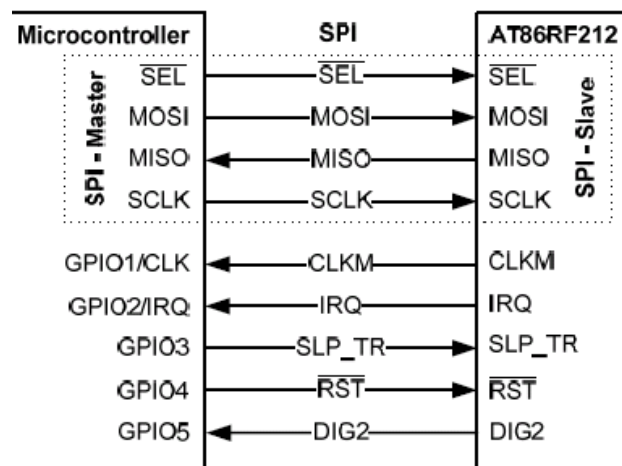


Figura 53 Conexión microcontrolador con el At86Rf212.

Además de las señales del puerto SPI, están las señales de control de la radio, en donde tenemos las siguientes señales:

- CLKM: es una salida de reloj, que es enviada al microcontrolador, para utilizarla como una fuente de reloj.
- IRQ: es una interrupción que utiliza la radio para dar a conocer si entra información a nuestro controlador.
- SLP_TR: es una señal que se utiliza para múltiples propósitos, como son el despertar o el dormir del dispositivo, iniciar el estado de transmisión, habilitar/deshabilitar la señal CLKM.
- /RTS: es la señal de Reset, que se utiliza para inicializar la radio.
- DIG2: es una señal que se encarga de múltiples procesos, entre ellos el control de la recepción de tramas.

CLKM

El pin 17 es el pin de salida del CLKM, y puede ser usado como fuente de reloj, este reloj se utiliza para la comunicación SPI, utilizando la frecuencia del CLKM como si fuera la señal del reloj maestro del SPI (SCLK). Pudiendo obtener una velocidad, que dependiendo si es modo síncrono o asíncrono será de 8MHz o de 7.5MHz.

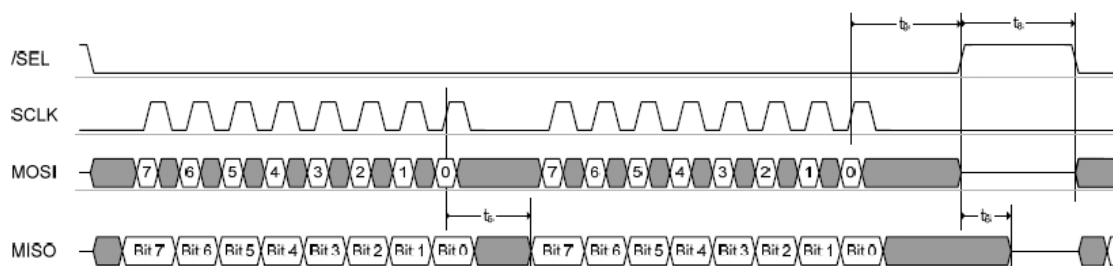


Figura 54 Transmisión del SPI.

Cuando se activa la señal /SEL, activa a nivel bajo, es cuando se produce la comunicación SPI, y cuando se desactiva /SEL, desactivándose con un nivel alto, es cuando se da por finalizada la comunicación.

IRQ

Nuestra radio soporta 8 tipos de interrupciones, las cuales podemos ver en la tabla 5.

Nombre de la IRQ	Descripción
IRQ_7 (Batería baja)	Indica si el voltaje de alimentación no es estable.
IRQ_6 (Error de Transmisión)	Indica que hay una violación en el acceso a la trama del buffer.
IRQ_5 (AMI)	Indica direccionamiento matemáticamente.
IRQ_4 (CCA_ED_DONE)	Tiene múltiples propósitos, pudiendo indicar un cambio de estados de TRX o un cambio de medidas de CCA o ED.
IRQ_3 (TX Finalizada)	TX, RX: indica que han sido enviadas o recibidas las tramas.
IRQ_2 (RX Empezada)	Indica el comienzo de una recepción PSDU; cambia del estado TRX a BUSY_RX.
IRQ_1 (PLL - Desbloqueado)	Indica que el PLL está desbloqueado, si esta en un estado de BUSY_TX o BUSY_TX_ARET, el PA es apagado inmediatamente.
IRQ_0 (PLL - Bloqueado)	Indica PLL Bloqueado.

Tabla 5 Tipos de interrupciones.

La señal externa de interrupción sale a través del Pin de Interrupción (pin 24). Pudiéndose a través de los registros de interrupción, configurar el nivel de interrupción, así como enmascararla o ver su estado.

Después de un Reset, todas las interrupciones están desactivadas, teniendo que activarse colocando un nivel alto en el registro IRQ_MASK.

Además tenemos que configurar la polaridad de la interrupción, es decir, si es activada con un nivel alto o con un nivel bajo, esto lo podemos controlar a través del registro IRQ_POLARITY.

SLP_TR

Es un pin multifuncional, es decir, dependiendo del estado en el que este el proceso de transmisión o recepción, hará que se produzca una acción u otra.

Para entender mejor el proceso, observaremos la tabla 6:

Estado	Función	Transición	Descripción
PLL_ON	Comienzo de TX	L→H	Comienzo la transmisión de la trama.
TX_ARET_ON	Comienzo de TX	L→H	Comienza Tx_ARET transacción.
BUSY_RX_AACK	Comienzo de TX	L→H	Comienza la transmisión ACK durante RX_AACK.
TRX_OFF	Sleep	L→H	Pone la radio en modo sleep, deshabilita CLKM.
SLEEP	Wakeup	H→L	Coloca la radio al estado de TX_OFF.
RX_ON	Deshabilitar CLKM	L→H	Coloca la radio en el estado RX_ON_NOCLK, deshabilitando CLKM.
RX_ON_NOCLK	Habilitar CLKM	H→L	Coloca la radio en el estado RX_ON, habilitando CLKM.
RX_AACK_ON	Deshabilitar CLKM	L→H	Coloca la radio en el estado RX_AACK_ON_NOCLK, deshabilitando CLKM.
RX_AACK_ON_NOCLK	Habilitar CLKM	H→L	Coloca la radio en el estado RX_AACK_ON, habilitando CLKM.

Tabla 6 Funciones del Pin SLP_TR.

Este pin SLP_TR es **fundamental** para controlar y para cambiar los estados de transmisión y recepción.

***Los estados del radio transmisor los explicaremos más adelante en el tema 5.4.6 Estados del Radio Transmisor.

5.4.5 Protocolo SPI del AT86Rf212

El protocolo que utiliza el SPI de la radio es muy simple, en el byte en donde se pone la dirección de los registros, se escribe un nivel alto o un nivel bajo en el bit 6 para indicar si es un acceso de escritura o un acceso de lectura.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access Mode	Access Type
1	0	Register address [5:0]						Register access	Read access
1	1	Register address [5:0]							Write access
0	0	1	Reserved					Frame Buffer access	Read access
0	1	1	Reserved						Write access
0	0	0	Reserved					SRAM access	Read access
0	1	0	Reserved						Write access

Tabla 7 Protocolo SPI.

Aparte también se utilizan los bits 7 y 5, para otras funciones como son el acceso al buffer de tramas y el acceso a la memoria SRAM.

Para realizar una lectura en el puerto SPI, se sigue el siguiente procedimiento:

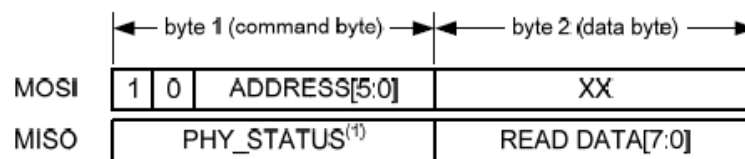


Figura 55 Acceso de Lectura.

Se envía la dirección por el puerto MOSI, configurada con el bit 6 puesto a nivel bajo indicando que es un acceso de lectura. Cuando es enviada, el dispositivo esclavo carga los datos de la dirección que le pusimos y para poder leer estos datos, necesitamos mandar un byte cualquiera (XX), para que así, el dispositivo cargue los datos que vamos a leer en la línea MISO, es decir,

necesitamos hacer 2 lecturas, una con los datos del registro que deseamos leer y otra lectura con datos aleatorios.

Para realizar la escritura se realiza un proceso muy similar:

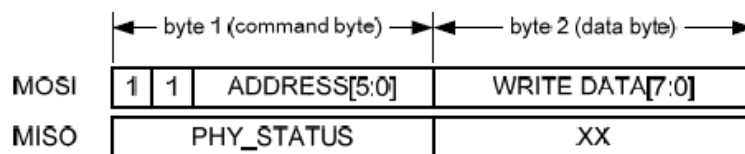


Figura 56 Acceso de escritura.

Ponemos el bit 6 del byte de dirección a nivel alto, para indicar que es un acceso de escritura y luego enviamos este byte por la línea MOSI. Una vez hecho esto, cargamos los datos que vamos a escribir y lo enviamos de nuevo por la línea MOSI.

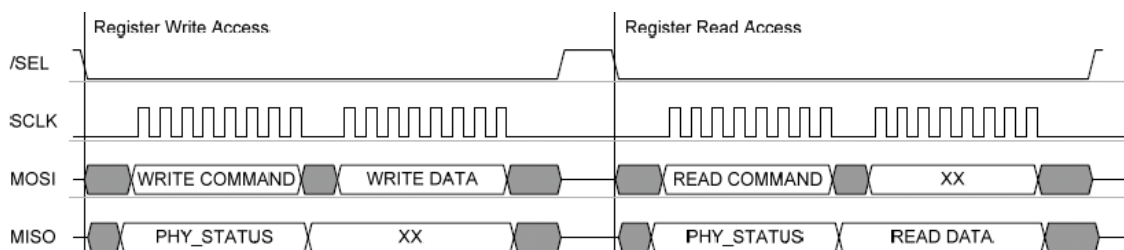


Figura 57 Ejemplo de la secuencia de escritura y de lectura.

La escritura o lectura del buffer de tramas o de la memoria SRAM, realiza un proceso similar al anterior, sólo que en vez de hacer un solo acceso de lectura o escritura cuando se activa la señal /SEL, realiza múltiples accesos de lectura o escritura mientras la señal /SEL está habilitada.

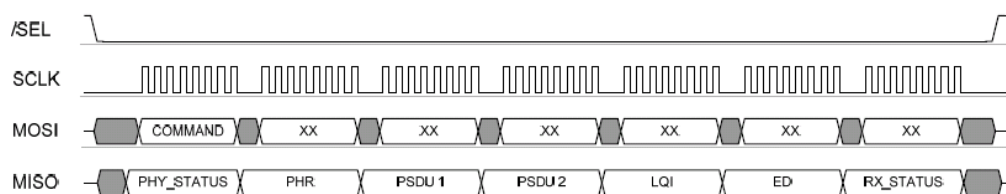


Figura 58 Ejemplo de lectura de una trama del buffer.

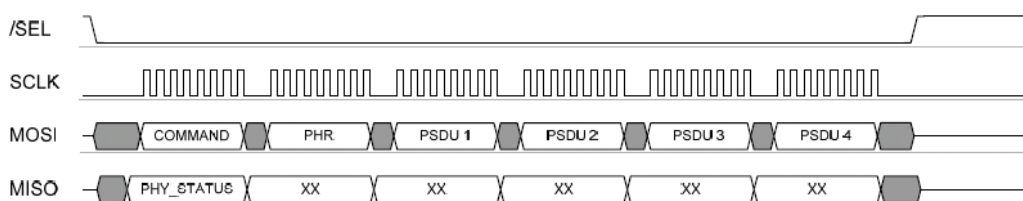


Figura 59 Ejemplo de escritura de una trama del buffer.

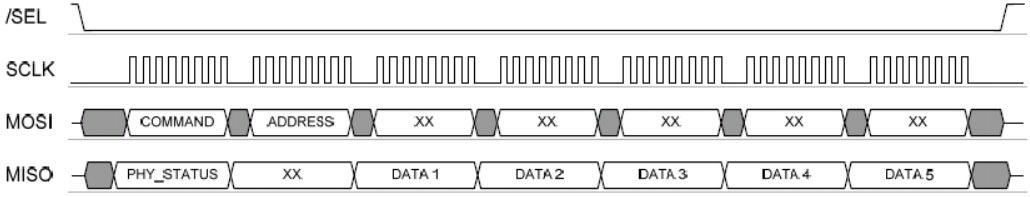


Figura 60 Ejemplo de lectura de la SRAM.

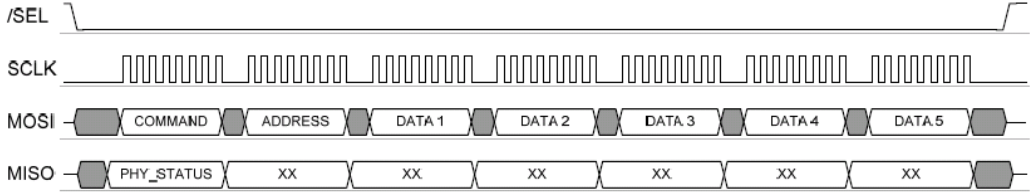


Figura 61 Ejemplo de escritura de la SRAM.

5.4.6 Estados del Radio Transmisor.

A continuación detallaremos los estados en los que se puede encontrar el radio transceptor, que dependiendo del estado en que se encuentre y de las señales que activemos, saltará de un proceso a otro. En la siguiente figura se puede observar las señales que activamos para cambiar de un estado al otro.

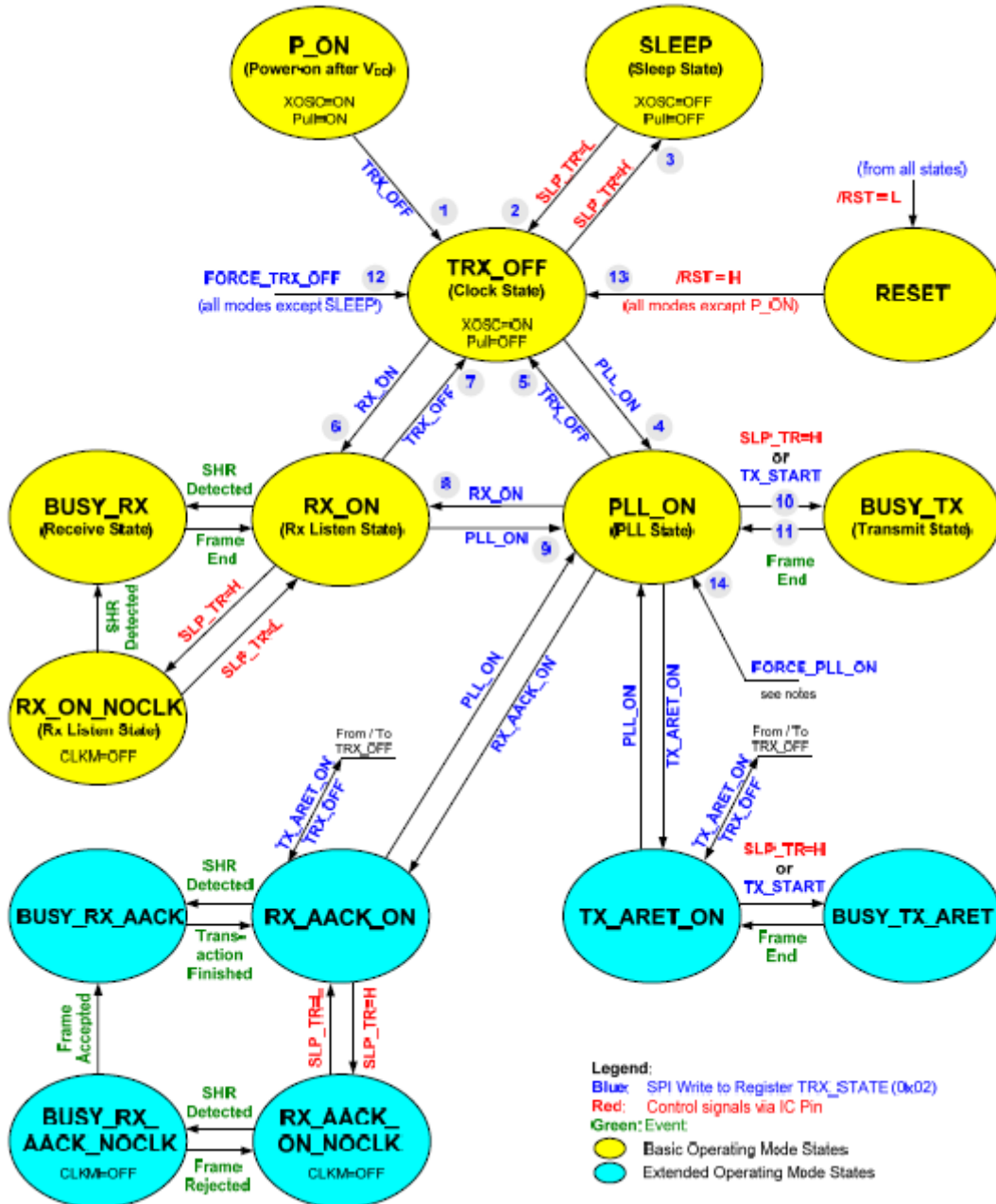


Figura 62 Estados del AT86Rf212.

En cada estado la radio tendrá algunas de sus funciones activadas o desactivadas. Explicaremos cada proceso para saber qué funciones se pueden realizar:

- P_ON: se entra en este estado cuando se le suministra la alimentación eléctrica (Vdd) a la radio. Es habilitado el reloj interno de 1MHz y se estabiliza la señal del CLKM. También se inicializa el SPI. No es necesario hacer un Reset en el pin 8 (/RST), pero es recomendable para la sincronización del software con el hardware.
- SLEEP: cuando está en este modo, todos los circuitos del radio transceptor están deshabilitados. En este modo, el circuito sólo consume una corriente mínima (típicamente 100 nA). Sólo se puede entrar a este estado activando el pin SLP_TR = H, cuando se está en el estado TRX_OFF. Para salir del estado Sleep se hace justamente lo contrario: se pone SLP_TR = L y se entra al estado TRX_OFF.
- TRX_OFF: se habilita el reloj maestro CLKM, el SPI y el voltaje interno digital. Esto hace que los registros de la trama buffer y los parámetros de seguridad AES sean accesibles.
- PLL_ON: al entrar en este estado, primero se activa el regulador de voltaje (AVREG). Dicho voltaje habilita la frecuencia del sintetizador PLL. Esta frecuencia se puede definir en el bit de registro CHANNEL.
- RX_ON y BUSY_RX: en el diagrama de estados del radio transceptor, estos aparecen separados; pero en realidad, no hay diferencias entre estos estados en lo que se refiere al circuito analógico. Los dos reciben la frecuencia del sintetizador PLL. Durante RX_ON, el sistema está listo para recibir una trama válida. Cuando detecta la cabecera de la trama (SHR), automáticamente entra en el estado BUSY_RX. Finalmente termina de recibir la trama, genera una interrupción (IRQ_3) y retorna al estado RX_ON.
- RX_ON_NOCLK: a este estado entra a partir de RX_ON, configurando el pin SLP_TR=H. Funciona de igual forma que RX_ON, sólo que desactiva CLKM. El reloj CLKM se desactivará 35 ciclos después de la activación de

pin SLP_TR, permitiendo al microcontrolador la secuencia de apagado. Cuando recibe una trama, entra en el estado BUSY_RX, y cuando se termina la trama, se ingresa en el estado RX_ON. No regresa al RX_ON_NOCLK a no ser que se vuelva a activar el pin SLP_TR.

- BUSY_TX: se entra en este modo a partir del estado PLL_ON, puesto que necesita que la frecuencia del sintetizador PLL esté activada. Para entrar se puede hacer de dos modos:
 1. Con un flanco de subida del pin SLP_TR.
 2. Escribiendo en el registro TX_START (registro 0x02, TX_STATE).Después de que la transmisión de la trama haya terminado, se produce la interrupción IRQ_3 (TRX_END) y retorna al estado PLL_ON.
- RESET: este estado fuerza a poner todos los registros al valor inicial por defecto. Cuando se produce el Reset, el radio transceptor es obligado a entrar al registro TRX_OFF. Para entrar a este estado, hay que poner el pin /RST=L.
- RX_AACK_ON, RX_AACK_ON_NOCLK, BUSY_RX_AACK_NOCLK, BUSY_RX_AACK, TX_ARET_ON, BUSY_TX_ARET: son estados en modo extendido, que sirven para una mejor comunicación automática, reconociendo directamente la trama ACK. Nosotros no utilizamos el modo extendido, por lo tanto no se va a explicar sus funciones.

5.4.7 Descripción funcional

La descripción del estándar IEEE 802.15.4 fue explicada anteriormente en “El Estándar IEEE 802.15.4”, pero en este apartado hablaremos un poco más de él, pues el At86Rf212 es lo que utiliza para la transmisión de las tramas.

PHY Protocol Data Unit (PPDU)			
Preamble Sequence	SFD	Frame Length	PHY Payload
5 octets Synchronization Header (SHR)		1 octet (PHR)	max. 127 octets PHY Payload PHY Service Data Unit (PSDU)
MAC Protocol Data Unit (MPDU)			

Figura 63 Trama de la estructura de la capa PHY.

La sincronización de cabecera (SHR) la forman cuatro bytes, todos a nivel “0”, seguidos de un único byte llamado SFD, que es el que limita el empezar de la trama.

La transmisión de los 5 bytes de la cabecera SHR, requiere 40 símbolos en la transmisión BPSK y 10 símbolos en la transmisión O-BPSK. La duración de la cabecera SHR depende del Baud Rate que elijamos.

PHY Mode	PSDU Bit Rate (kbit/s)	SHR Bit Rate (kbit/s)	Duration		
			SHR (μ s)	PHR (μ s)	Max. PSDU (ms)
BPSK	20	20	2000	400	50.8
	40	40	1000	200	25.4
O-QPSK	100	100	300	80	10.16
	250	250	160	32	4.064
O-QPSK	200	100	300	80	5.08
	400	100	300	80	2.54
	500	250	160	32	2.032
	1000	250	160	32	1.016

Tabla 8 Tiempos de la PPDU.

Como vemos los tiempos dependen tanto de la modulación como del Baud Rate que utilicemos. No debemos olvidar que la velocidad del SPI es mayor a la velocidad de nuestra comunicación aérea, por lo tanto debemos tenerlo en cuenta para tener libre el buffer de transmisión/recepción.

El PHR sólo mide un byte y sirve para indicar la longitud de la trama PSDU.

El PSDU tiene una longitud de 0 a 127 bytes. Dentro de la PSDU se encuentra el protocolo MPDU, en donde los dos últimos bytes son usados para la secuencia de la trama de chequeo (FCS).

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

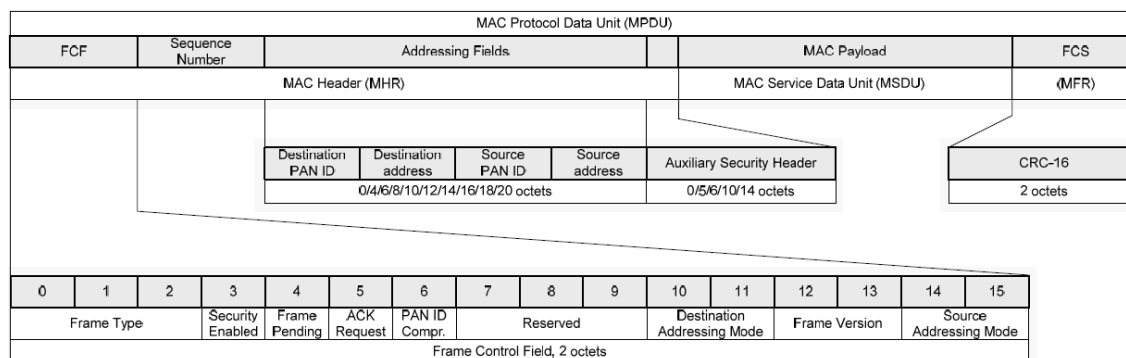


Figura 64 Estructura de la MPDU.

La MHR contiene tres campos: el campo FCF, una secuencia numérica y el direccionamiento de los campos.

El campo FCF está compuesto por 2 bytes de control que se encargan de lo siguiente:

- Indicar el tipo de trama.
- Habilitar la seguridad.
- Comprobar si hay tramas pendientes.
- Solicitar el ACK.
- Indicar en la trama el campo de destino y fuente de la misma (PAN ID).
- Describir el formato de la dirección de destino.
- Indicar la versión a la que se corresponda la trama (IEEE 802.15.4-2003, IEEE 802.15.4-2006).
- Indicar la fuente del modo de direccionamiento de nuestro At86Rf212.

La secuencia numérica es de un byte y nos identifica una trama en particular, para que las tramas duplicadas en la transmisión puedan ser detectadas.

El campo de dirección indica la dirección del campo de destino, seguido del campo de la fuente de la trama. Cada dirección consiste en el campo PAN ID y la dirección del dispositivo.

El campo auxiliar termina la MHR y especifica la información de seguridad, incluyendo cómo está protegida la trama.

Después del campo MHR, viene el campo MSDU (MAC Service Data Unit) que es el que contiene la trama de datos que fue explicada en el TEMA 3 “El Estándar IEEE 802.15.4”.

Y termina con 2 bytes del campo FCS, que se utiliza para comprobar si hay errores en la trama, usando un código de redundancia cíclica (CRC).

5.5 Acelerómetro (ADXL345)

5.5.1 Descripción

El acelerómetro ADXL345 es una de las mejores soluciones que se tiene hoy en día para utilizar en los dispositivos móviles. Es capaz de medir las aceleraciones en estático realizando mediciones de la gravedad. Debido a su alta resolución (3.9mg/LSB), puede realizar medidas de inclinaciones inferiores a 1°.

Nuestro acelerómetro es muy pequeño (3 mm × 5 mm × 1 mm), es de ultra baja potencia (140µA en modo normal y 0.1µA en standby), tiene la capacidad de medir hasta 16 g en los 3 ejes, con una resolución de 13 bits. Su salida es digital, por medio de 16 bits, a través del bus SPI (3 o 4 cables), pudiendo transmitir también a través de la interfaz I2C.

Incluye una memoria FIFO estando preparado para almacenar hasta 32 niveles de medidas. Nosotros no usamos la FIFO, pues lo que queremos es leer el valor que tiene en el momento de la lectura.

Su estructura interna es simple, conteniendo los bloques que observamos en la figura 65.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

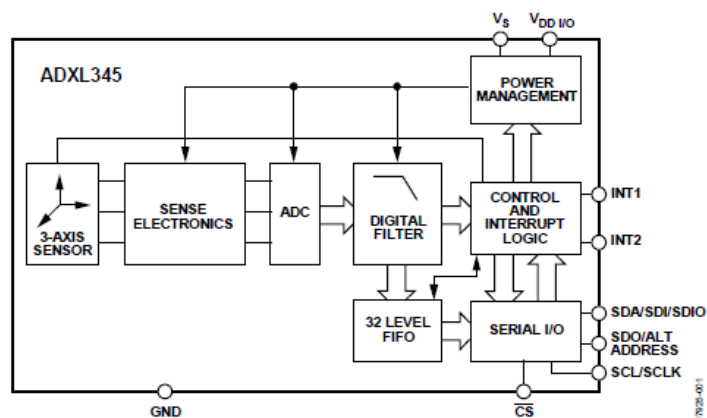


Figura 65 Diagrama de bloques ADXL345.

5.5.2 Diagrama Funcional

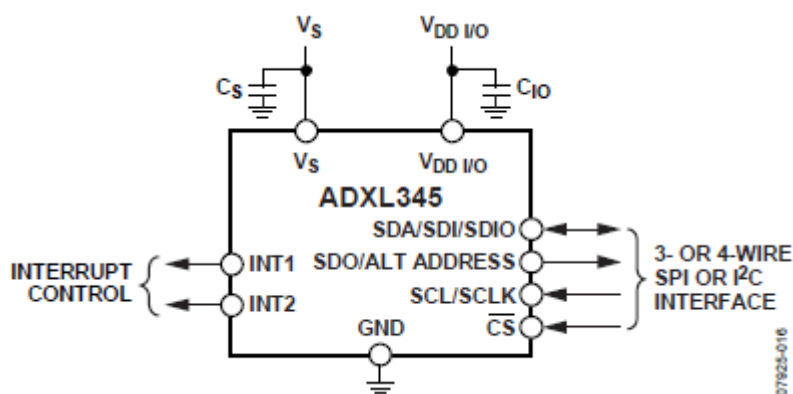


Figura 66 Diagrama del acelerómetro (ADXL345).

ACLARACIÓN: En el diseño de nuestro acelerómetro, se implementó un buffer, debido a que compartían el mismo puerto SPI que los barómetros (MS5540C) y este modelo de barómetros no tiene la capacidad de desconectarse mediante el pin /Cs del SPI.

Se le colocó un condensador de Tántalo, dotando a la tensión V_s de las prestaciones que nos ofrecen dichos condensadores (velocidad de respuesta ante picos de tensión). Asimismo, se colocó en serie un chip de ferrita, para hacer más estable la corriente y evitar los picos de tensión.

5.5.3 Salida de datos

Para darnos las lecturas de los ejes, nuestro ADXL345 tiene las opciones de utilizar el SPI o el I2C. Ante esta facilidad, nos decidimos por utilizar el puerto SPI, pues este puerto tiene mucha más velocidad que un puerto I2C y pudiendo conectar varios dispositivos en el mismo puerto, ahorramos programación.

La configuración de los datos de salida se puede observar en las figuras 67 y 68.

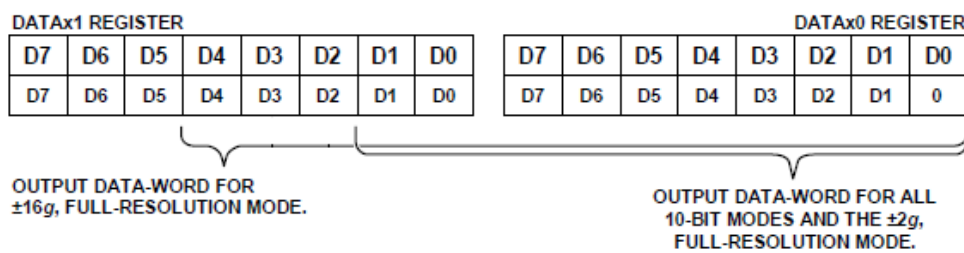


Figura 67 Salida de datos utilizando justificación a la derecha.

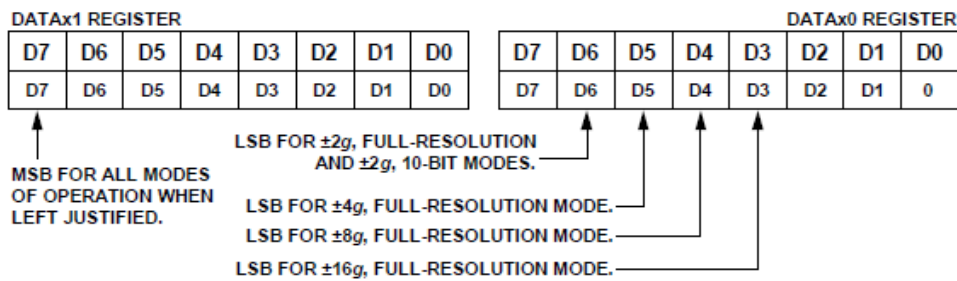


Figura 68 Salida de datos utilizando justificación a la izquierda.

Dependiendo de la configuración que utilicemos, los datos saldrán justificados a la derecha o izquierda. En nuestro proyecto utilizamos la justificación a la derecha, pues nos es más sencillo trabajar con los datos a la hora de la programación.

5.6 Giroscopio (L3G4200D)

5.6.1 Descripción

Este giroscopio es uno de los más nuevos del mercado, y una de sus características es la de poder medir los 3 ejes en un solo chip. Anteriormente había que utilizar 2 chips diferentes: uno que medía dos ejes (X,Y) y otro chip que medía el eje (Z).

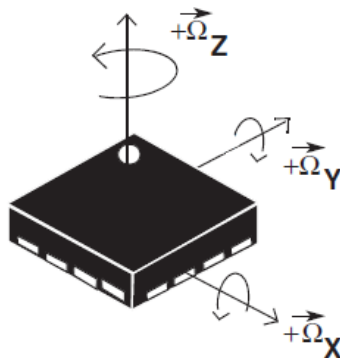


Figura 69 Ejes de medida.

Tiene la capacidad de comunicarse por medio del puerto SPI o I2C, haciendo más fácil la adaptación a nuestro diseño.

Se puede alimentar con tensiones de entre 2.4v y 3.6v, consumiendo una corriente máxima de 6.1mA, con un rango de temperatura que oscila entre los -40°C y los 85°C y una frecuencia máxima del SPI de 10MHz.

Contiene una memoria FIFO la cual se puede configurar de tres formas diferentes:

1. Bypass Mode: se sobrescribe el dato anterior por el dato nuevo, sólo se utiliza un slot de memoria (figura 70).
2. FIFO Mode: se escribe la memoria completa y se sacan los primeros datos que se recogieron, que son los datos más antiguos (figura 71).
3. Stream Mode: se va rellorando la memoria completa como en la FIFO Mode, pero con la diferencia de que a medida que van llegando nuevos datos, se van eliminando los más antiguos si no se han leído (Figura 72).

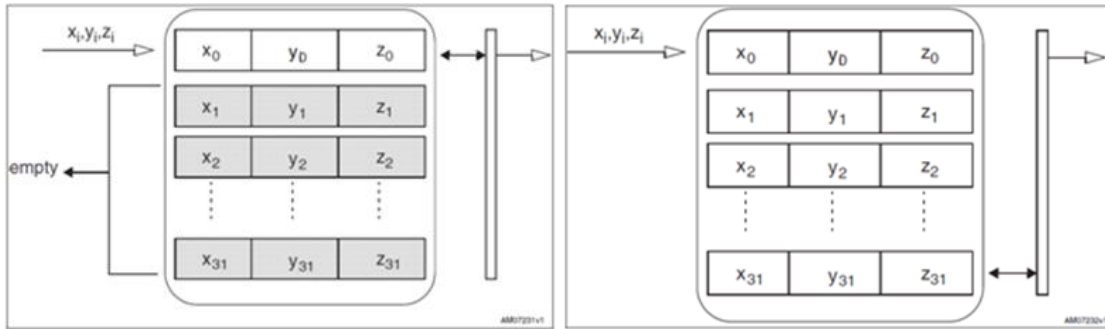


Figura 70 Bypass Mode.

Figura 71 FIFO Mode.

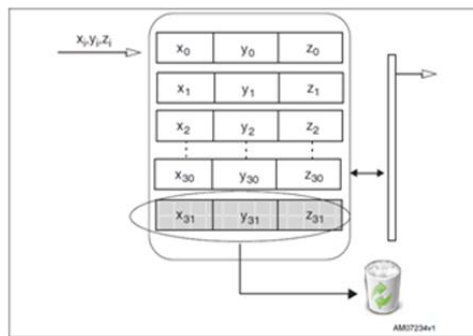


Figura 72 Stream Mode.

5.6.2 Diagrama de bloques

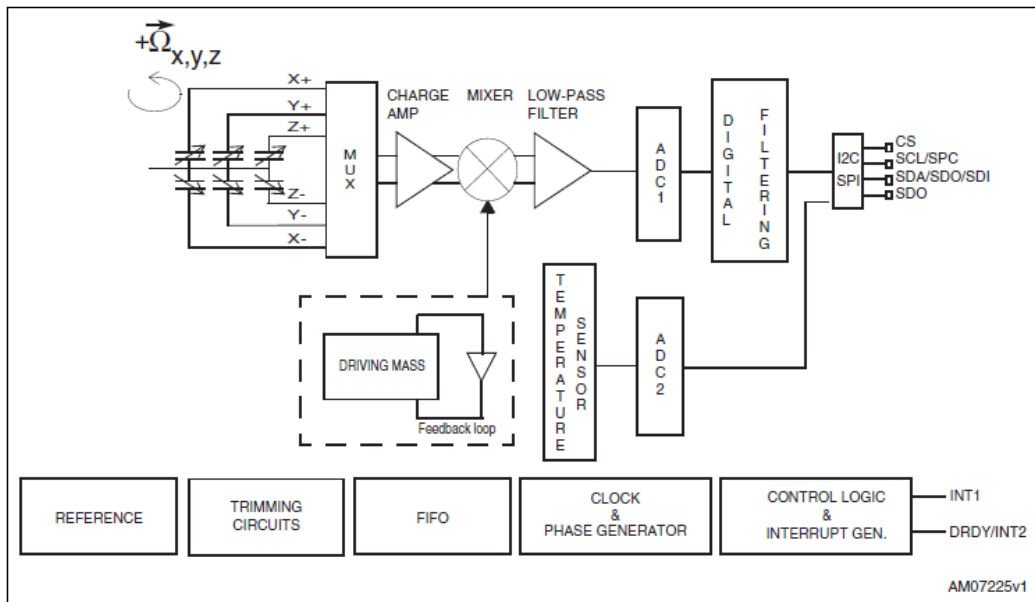


Figura 73 Diagrama de bloques del giroscopio L3G4200D.

En el diagrama de bloques vemos los tres ejes (X, Y, Z) en un solo chip, que es lo fundamental que tiene este chip. Además se pueden observar los otros bloques como son el de la memoria FIFO, los controles lógicos y de interrupción, el reloj interno y el sensor de temperatura.

Se puede observar también el bus de conexión, formado por 4 cables, si es por medio del SPI, o por 2 si se conecta por el bus I2C.

5.6.3 Diagrama funcional

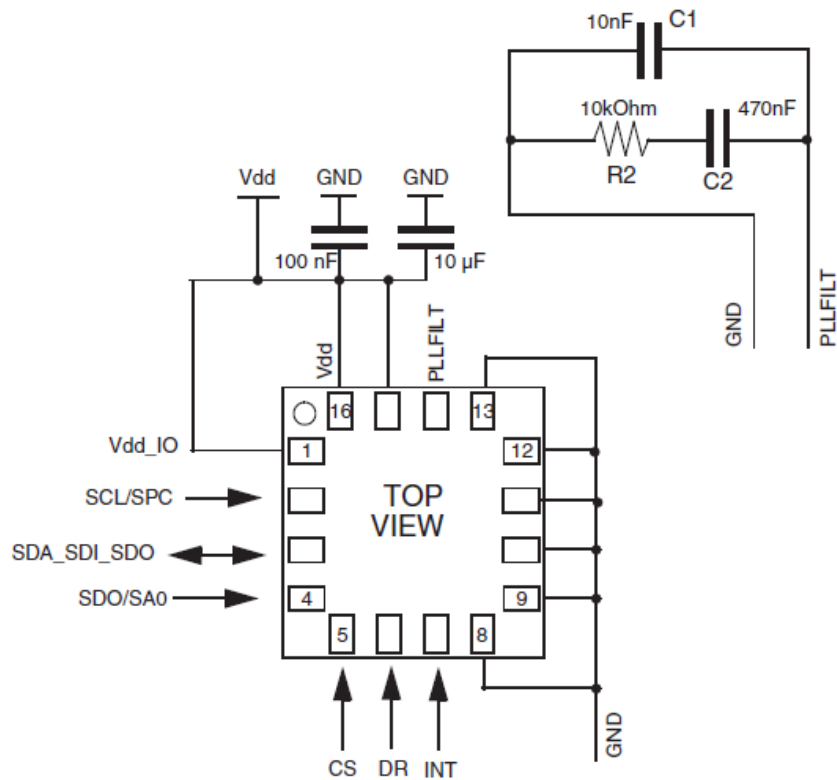


Figura 74 Diagrama Funcional del Giroscopio.

En la figura superior, se aprecia la conexión de cada pin de nuestro giroscopio.

La alimentación del L3G4200D es de tipo digital, no necesitándose ningún requerimiento específico para alimentarlo, basta con 2 condensadores de desacoplo, de tamaños diferentes, para alimentar al dispositivo si hubiera algún pico de tensión.

También vemos un pequeño filtro paso bajo de segundo orden que utiliza el giroscopio para sincronizar el circuito de conducción con el de detección de las interfaces.

5.7 Barómetro (MPL115A1)

En un principio, se diseñó nuestra placa para el barómetro MS5540C, el cual tenía la comunicación SPI de 3 cables, no teniendo la conexión del pin /Ss. Al no tener este pin, tuvimos que implementar un buffer, para poder conectar los 2 barómetros en el mismo puerto SPI, al que también estaban conectados el acelerómetro, el giroscopio y la radio.

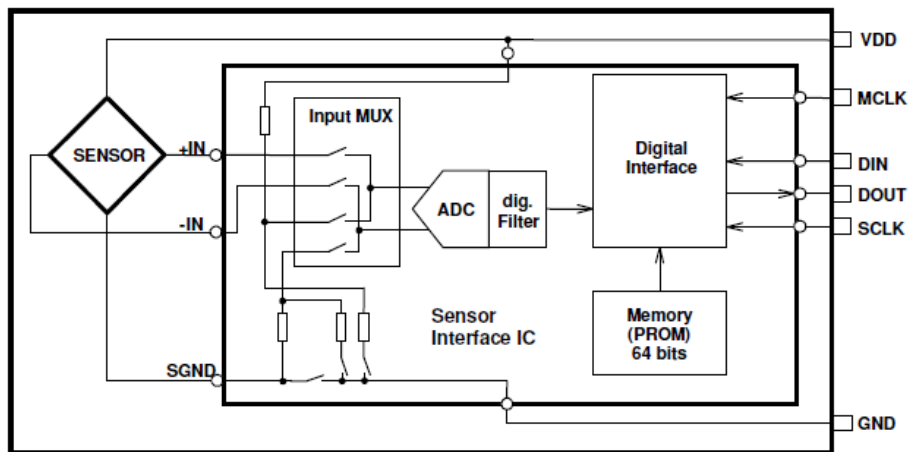


Figura 75 Diagrama de Bloques MS5540C.

Como se aprecia en el diagrama de bloques, las conexiones del puerto SPI sólo tienen los pines de: Datos de Entrada (DIN), Datos de Salida (DOUT) y Reloj (SCLK). Por este motivo se realizó la implementación del buffer, el cual provocaba que se desconectarán los cables de salida de datos y del reloj, de los dos barómetros.

Por motivos del mercado, y de sus mejores prestaciones, se cambió el barómetro, a un nuevo modelo con características superiores. Este nuevo modelo es el MPL115A1.

5.7.1 Características del MPL115A1

Nuestro nuevo barómetro, cuenta con un puerto SPI de cuatro pines, por lo que ya no hace falta ningún buffer para su conexión. Además cuenta con una calibración de fábrica que llega a alcanzar una precisión de ± 1 kPa.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.



LGA Package
5.0 mm by 3.0 mm by 1.2 mm

Figura 76 Dimensiones del Barómetro.

Con unas dimensiones considerablemente pequeñas, como se pueden apreciar en la figura 76, nuestro barómetro es capaz de darnos los datos de presión y temperatura con gran exactitud.

Se alimenta con un rango que va desde los 2.375V a los 5.5. Dicho rango es más que adecuado para nuestro diseño. Tiene un consumo máximo de corriente de $10\mu\text{A}$, y es capaz de medir presiones de entre 50 kPa y 115 kPa y de comunicarse con el puerto SPI a una frecuencia de 8MHz.

5.7.2 Diagrama de bloques y conexiones

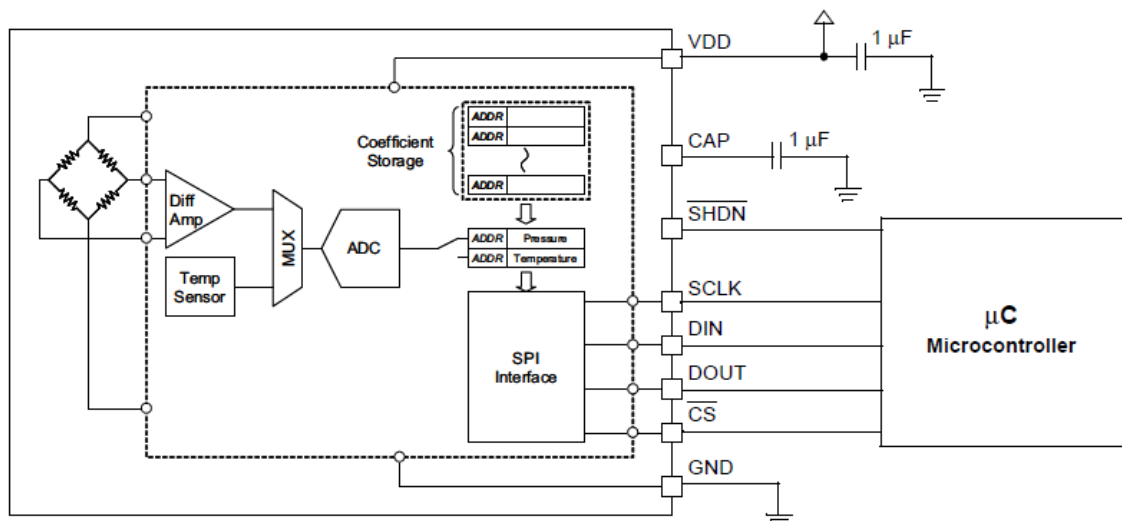


Figura 77 Diagrama de bloques y conexiones del MPL115A1.

Para que nuestro barómetro funcione, sólo le hace falta un condensador de desacoplo y un condensador CAP (pin 2) que se usa como desacoplamiento de un regulador interno principal.

5.7.3 Comunicación y coeficientes

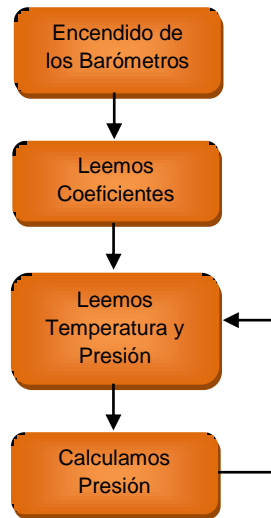


Figura 78 Secuencia de lectura del Barómetro.

Nuestro MPL115A1 viene con unos coeficientes internos de fábrica (a_0 , b_1 , b_2 y c_{12}), que es lo primero que leemos y guardamos, pues estos serán utilizados más adelante para nuestros cálculos de la presión.

Una vez leídos, convertimos esos coeficientes a su valor real, es decir, separamos los bits de números enteros en números fraccionarios, quedando nuestros coeficientes con valores flotantes.

Leemos los datos de presión y temperatura (P_{adc} y T_{adc}) y aplicamos los cálculos necesarios para que al final nos dé el valor de la presión atmosférica en KPa.

Los cálculos a realizar son:

$$c_{12x2} = c_{12} * T_{adc}$$

$$a_1 = b_1 + c_{12x2}$$

$$a_{1x1} = a_1 * P_{adc}$$

$$y_1 = a_0 + a_{1x1}$$

$$a_{2x2} = b_2 * T_{adc}$$

$$P_{Comp} = y_1 + a_{2x2}$$

Siendo el valor de P_{Comp} , el total de la presión expresado en KPa.

5.8 Magnético (HMC5883L)

Al igual que con el barómetro, en un principio se diseñó la brújula con otro IC (HMC5843 de 20 pines), el cual, se dejó de comercializar y se tuvo que cambiar a un nuevo sensor magnético, con mejores prestaciones como es el rango de sensibilidad que es el doble en nuestro nuevo chip, y pasa de ser de $\pm 4g$ en el IC antiguo a tener $\pm 8g$ en el IC nuevo (HMC5883).

5.8.1 Características Principales del HMC5883

Cuenta con una tensión de alimentación de 2.16v a 3.6v y un consumo de corriente de $100\mu A$. Posee un rango de medida que va desde ± 8 gauss, con una resolución de 5 mili-gauss y una sensibilidad de ± 0.2 %FS/gauss.

En el modo continuo de medida, puede tener una frecuencia de hasta 75Hz, y si sólo queremos el modo simple de medida, llega a una frecuencia de 160 Hz.

Al tener el bus de conexión I2C, con una frecuencia máxima de 400 kHz, es mucho más lento que el bus SPI, que puede alcanzar los 10 MHz.

Conectamos nuestro sensor magnético al puerto I2C y lo alimentamos con una tensión analógica especial para este integrado.

5.8.2 Diagrama de Aplicación del IC antiguo (HMC5883)

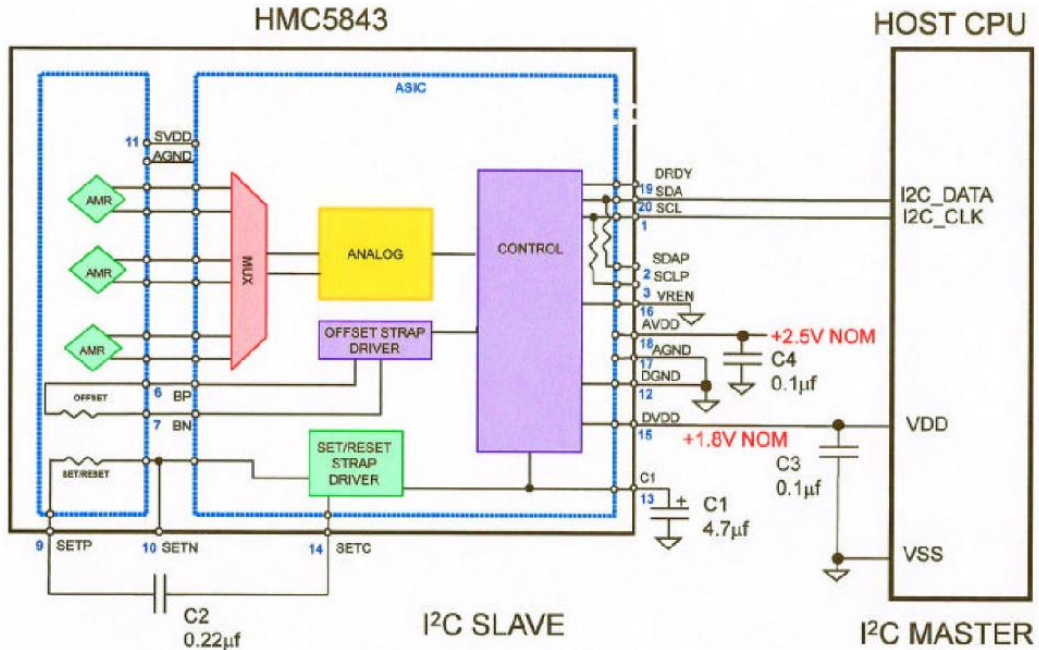


Figura 79 Diagrama de conexiones del HMC5843.

5.8.3 Diagrama de Aplicación del IC nuevo (HMC5883)

El nuevo chip no tiene la misma forma de conexión, como se muestra a continuación en la figura 80:

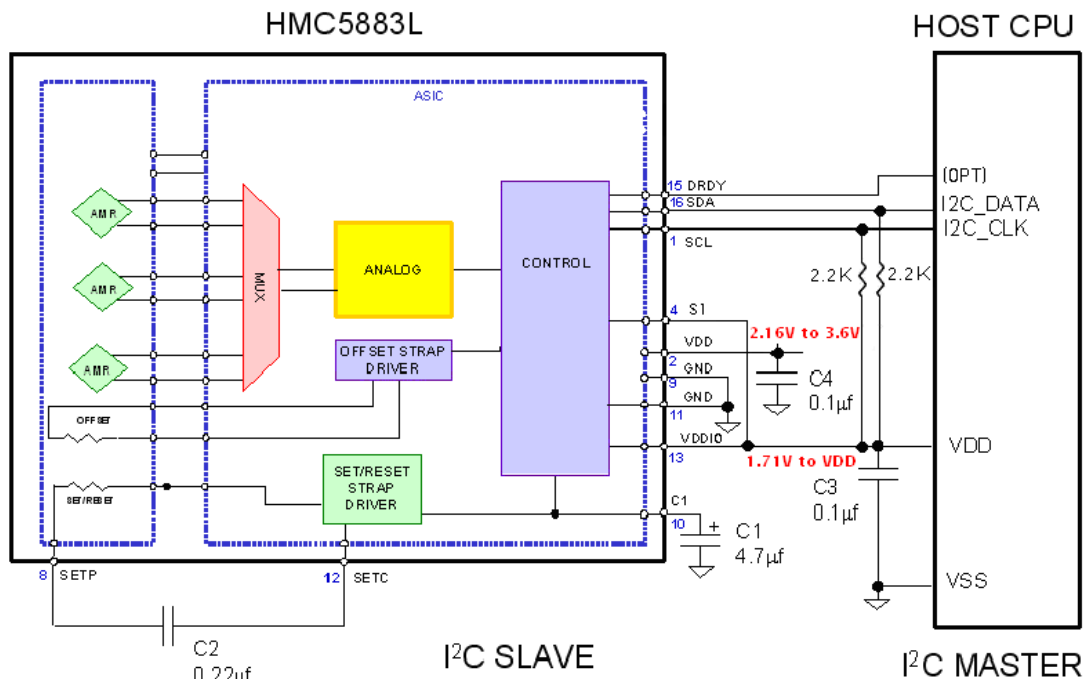


Figura 80 Diagrama de bloques y conexión del HMC5883.

Teniendo que cambiar nuestro diseño y adaptarlo para el nuevo IC. El cual tiene menos pines (16), en comparación con el antiguo que contaba con 20 pines.

5.9 Ultrasonido

Se llama ultrasonido a una onda acústica, cuya frecuencia está por encima del oído humano, sobre de los 20 kHz que es el límite superior de frecuencia que es capaz de percibir una persona.

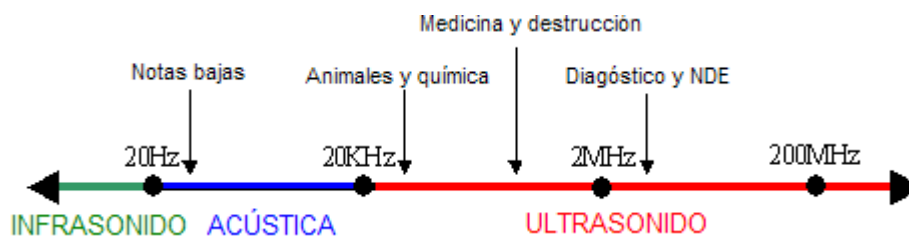


Figura 81 Rango de frecuencias de las ondas sonoras.

Los ultrasonidos son muy conocidos en dos ambientes: en el sector de la medicina y en el ambiente de algunos animales.

En medicina se utiliza para ver el estado de los embriones cuando están en el vientre de la madre, para observar tejidos que están por debajo de la piel, etc.

Los animales como los murciélagos, delfines, ballenas, etc. utilizan estas ondas ultrasónicas para formar una especie de radar, y poder localizar tanto obstáculos como presas. Debido a que estas ondas rebotan en las superficies se puede crear una imagen aproximada, de los obstáculos que interrumpen la onda en el medio.

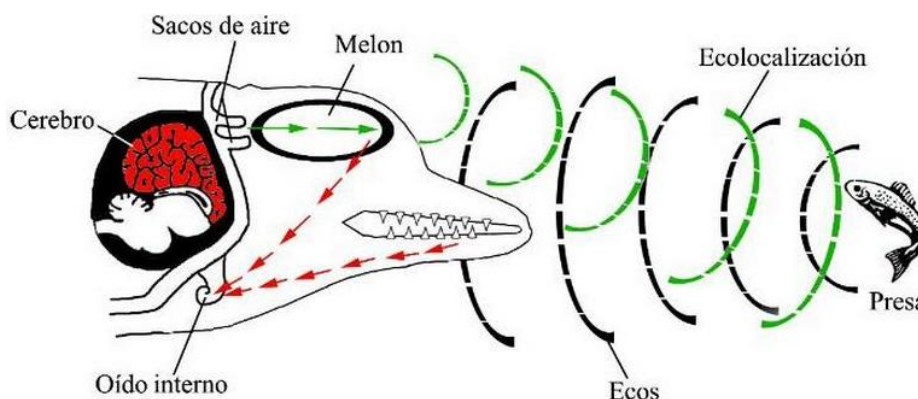


Figura 82 Ultrasonido en delfines.

5.9.1 Características

Las ondas ultrasónicas se crean a través de un transductor, que transforma una frecuencia a señal sonora. Por ejemplo, en medicina que se utiliza la frecuencia de entre 500 kHz y 3 MHz, se emplea un cristal de cuarzo a estas frecuencias acoplado a un transductor que emita el sonido.

Se utilizan las frecuencias más altas, cercanas a los 3 MHz, para tratar obstáculos pequeños, mientras que utilizaremos las frecuencias cercanas a los 500 kHz, para llegar a más profundidad, debido a que tienen menor absorción en el medio.

Dependiendo del medio que atraviese la onda sonora, tendrá una mayor o menor velocidad de propagación, siendo imprescindible saber el medio que traspasa la onda, para poder hacer los cálculos a la hora de recibir el eco. En el aire la velocidad de propagación es de 331 m/seg.

Cuando el haz ultrasónico atraviesa áreas de diferentes densidades (impedancias), se crea una reflexión de parte del haz. Esta reflexión será mayor cuanto más elevada sea la diferencia de impedancias, pudiendo incluso hacer que el haz ultrasónico no funcione correctamente.

En el área de medicina, es imprescindible utilizar geles y aceites, para evitar este desacoplamiento entre la piel y el transductor.

El ultrasonido a altas frecuencias es tan potente, que puede incluso producir un desgarre en el tejido. Por este motivo, los ultrasonidos médicos, tienen una frecuencia máxima que impide que se produzca este fenómeno.

5.9.2 Aplicaciones

Se usan los dispositivos de sonar (ecogoniómetro); los detectores de grietas en los materiales metálicos (opacos a las radiaciones electromagnéticas), que se

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

basan en el hecho de que las grietas reflejan parcialmente la energía ultra sonora incidente.

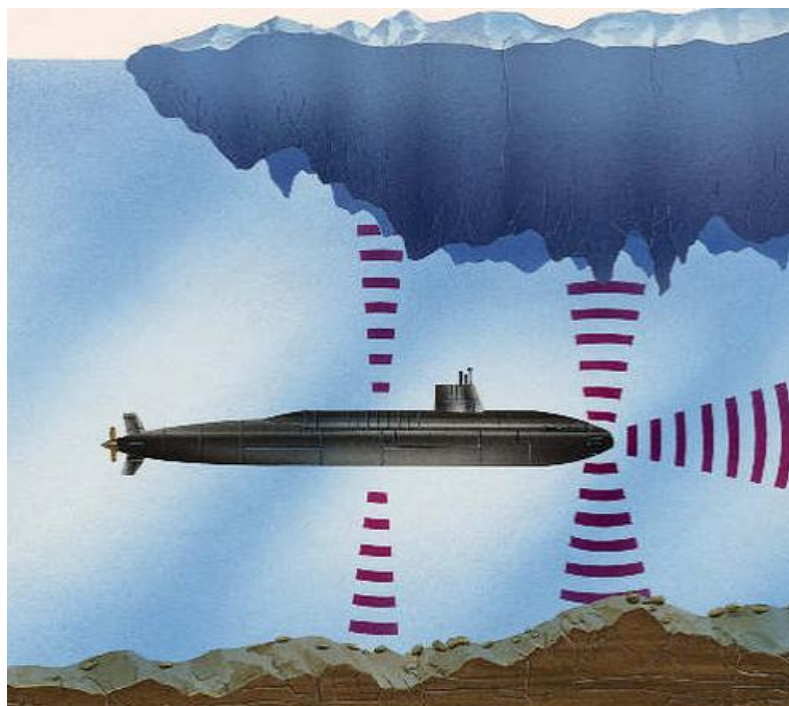
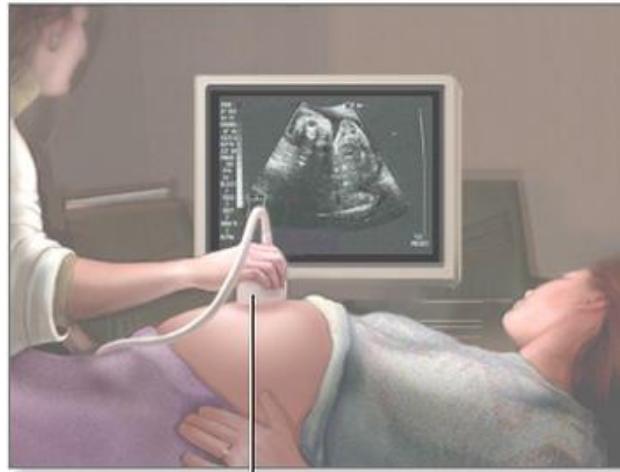


Figura 83 Dispositivos de sonar.

También se emplean en la medida de propiedades elásticas de los sólidos y de materias plásticas artificiales.

Además se aplican en ciertos procedimientos metalúrgicos, como el refinado de los granos cristalinos durante el enfriamiento de las coladas fundidas, la ventilación de metales fundidos mediante irradiación por ultrasonido y la soldadura del aluminio mediante la irradiación ultrasónica de las superficies que se van a soldar, para eliminar la capa superficial de óxido.

En el campo de la salud, se aplican en bacteriología (destrucción de microorganismos), en medicina (curación de enfermedades del oído, neuritis, periartrosis, artrosis y úlceras) y en ginecología se utilizan para ecografías.



Transductor

Figura 84 Ecografía.

En nuestro diseño, utilizamos las ondas ultrasónicas, para detectar obstáculos con los que podamos colisionar y para saber las distancias a la que se encuentra los objetos cercanos.

Para ello, utilizamos una serie de dispositivos IC con los que lograr una especie de radar.

5.9.3 Diagrama de aplicación

- Para transmitir la onda sonora, configuramos nuestro micro para que saque una frecuencia ultrasónica por el pin de salida del ultrasonido, el cual conectamos al MAX3232, que hace de amplificador, al amplificar onda de 3.3v hasta los 5v.
- Para recibir la onda, es un poco más complicado; el principio básico que utilizamos es el de: al recibir la onda del medio, la amplificamos, luego la pasamos por un comparador y la salida del comparador, va a un pin de nuestro microcontrolador, para que se pueda calcular el tiempo que ha tardado la onda en rebotar.

En la figura 85 se puede observar cómo se conecto los diferentes IC en el entorno del simulador MicroSim Schematics. Con este programa, podemos realizar

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

las simulaciones de los circuitos que utilizamos y conseguimos visualizar la señal que obtendremos a la salida. En la figura 86 vemos la simulación de nuestro ultrasonido.

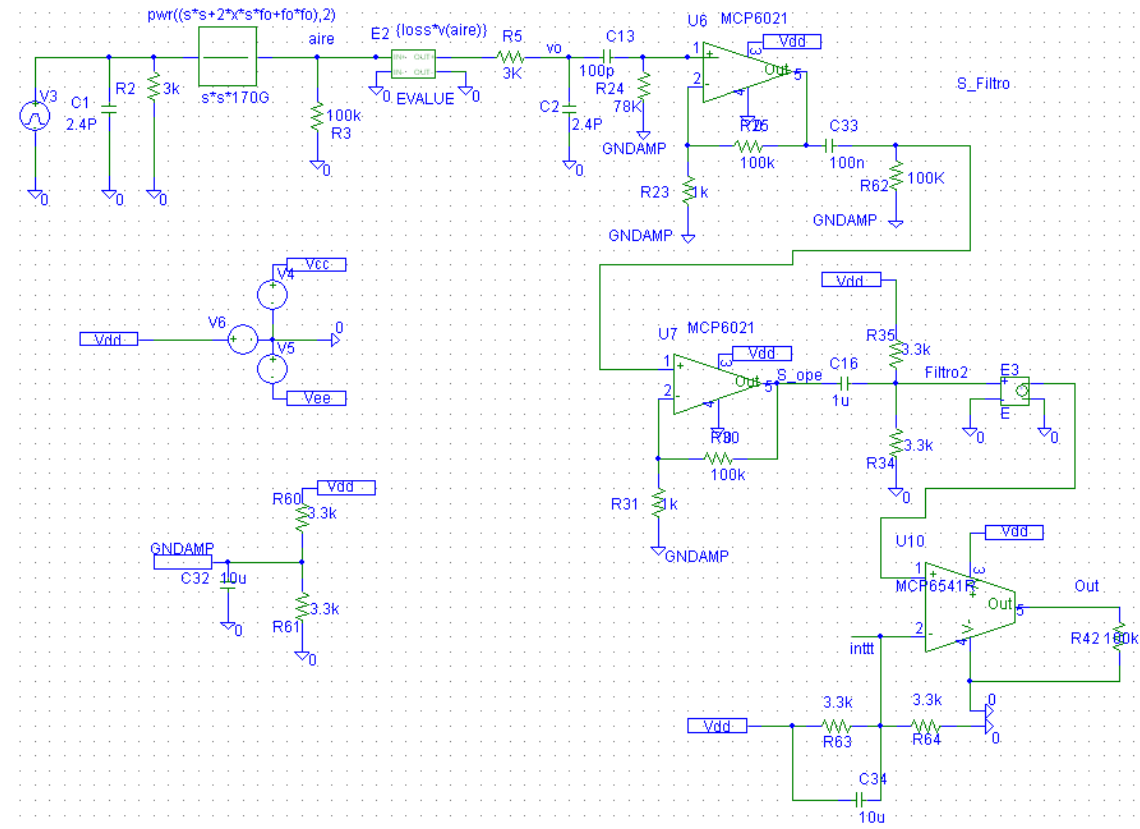


Figura 85 Conexión del Ultrasonido.

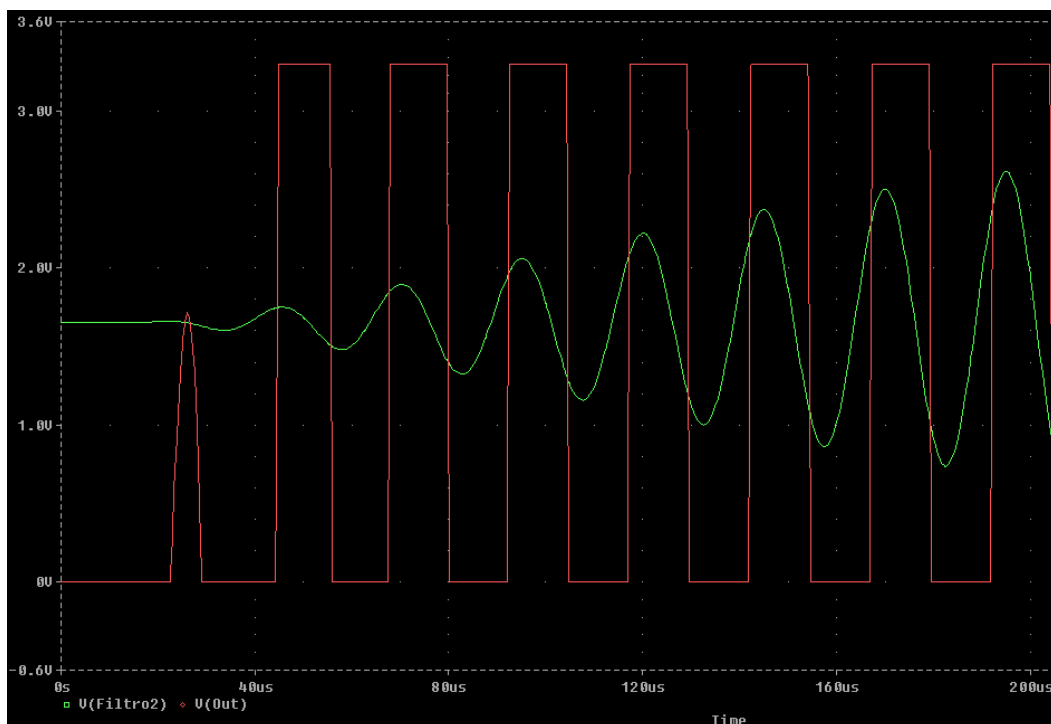


Figura 86 Análisis Transient del Simulador MicroSim.

5.9.4 Especificaciones

Transmisión

Para transmitir la onda ultrasónica, realizamos los siguientes pasos:

1. Configuramos el puerto F como salidas y entradas dependiendo de qué pin sea, poniendo los pines de salida a nivel H.
2. Activamos el puerto F0 = L (pin 46).
3. Al cambiar el pin de nivel, activamos un contador, que nos contara hasta que llegue la interrupción del ultrasonido.
4. El puerto F0 se conecta al MAX3232, el cual complementara la señal y la transformara a 5v.
5. El MAX3232 se conecta al transductor, el cual nos emitirá la onda ultrasónica.

Tras realizar estos pasos, nos quedamos a la espera de que la onda ultrasónica sea captada por nuestro transductor.

Recepción

En la recepción, realizamos los siguientes pasos:

1. Filtramos la señal, para solo recibir las frecuencias centradas en 40 kHz, con un ancho de banda de 2kHz.
2. Amplificamos la señal dos veces, con el amplificador MCP6022, con una ganancia regulable, pues no sabemos si la señal recibida tendrá la suficiente intensidad como para saturar al amplificador, o si vamos a tener una señal muy débil, necesitando la máxima amplificación.
3. Estabilizamos la señal al punto medio de la tensión, para que el comparador se active correctamente.
4. Pasamos la señal por el comparador, para que este nos produzca el flanco de subida, que producirá la interrupción en el puerto F6 (pin 56).

5. Cuando nos produce la interrupción, paramos el contador que activamos en el paso 3 de la transmisión.
6. Hacemos los cálculos necesarios para determinar el tiempo que tardo el haz de sonido en rebotar, sabiendo que la velocidad a la que se deslaza en el aire es de 331 m/seg.

5.10 Dispositivo USB (CP2102)

Para la conexión de nuestra placa a un PC, utilizamos un IC que se encarga de convertir los datos en serie del puerto USART, a datos I2C del puerto USB y viceversa.

El IC que se encarga de esto es el CP2102, el cual en una simple pastilla cuadrada que no llega a medio centímetro (4.8 mm), contiene todo lo necesario para realizar esta conexión, incluyendo las especificaciones del USB 2.0 (full-speed 12 Mbps).

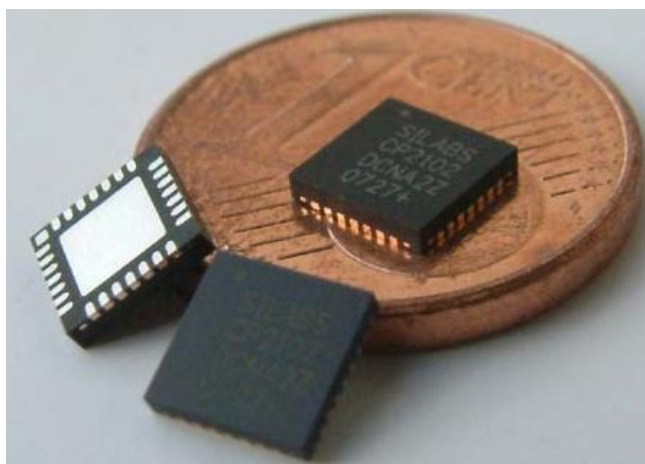


Figura 87 Integrado CP2102.

5.10.1 Características

Las características del CP2102 son las siguientes:

- Formatos de datos soportados de 5, 6, 7 y 8 bits.
- Bits de stop de 1, 1.5 y 2.
- Paridad de niveles bajos, altos y no paridad.

- Un Baud Rate que va desde los 300 bps a 1 Mbps.
- Pin de suspensión.
- Tensión de alimentación de 3.0v a 3.6v.
- Tensión del bus USB de 4.0v a 5.25v.
- Formato del encapsulado QFN.

5.10.2 Diagrama de bloques.

En el diagrama de bloques se puede ver como nuestro IC se conecta al USB por medio del bus I2C, y que por el otro lado, se conecta a nuestro microcontrolador, por medio del puerto USART, es decir, nuestro CP2102 es una interfaz entre nuestro microcontrolador y nuestro PC.

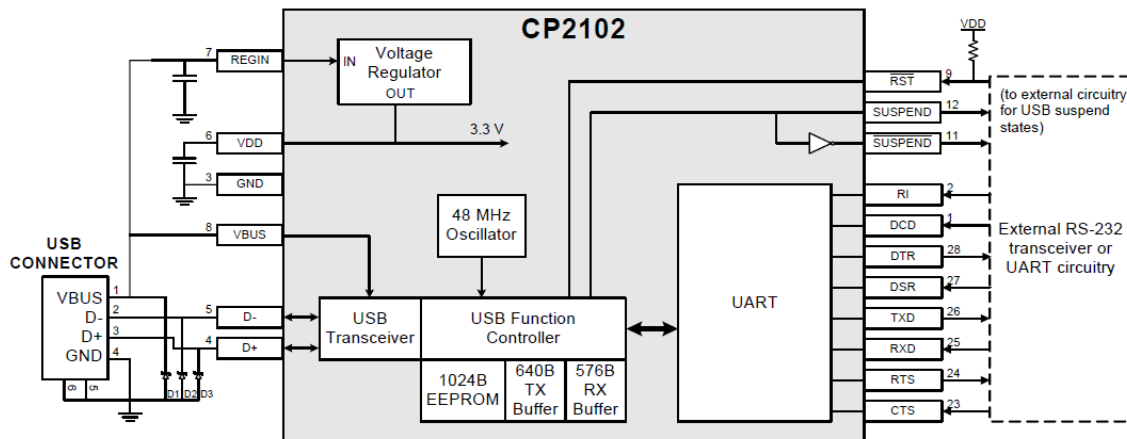


Figura 88 Bloques del CP2102.

El CP2102 cuenta con un oscilador interno de 48 MHz. Además cuenta con una memoria EEPROM de 1Mb. Y un bloque que es capaz de transmitir hasta 640 Bps y otro bloque capaz de recibir hasta 576 Bps.

Se conecta a nuestro PC como si fuera un puerto COM VIRTUAL, tal y como lo haría cualquier dispositivo con un puerto RS-232 o como una puerto serie de transmisión y recepción UART.

5.10.3 Diagrama de conexiones

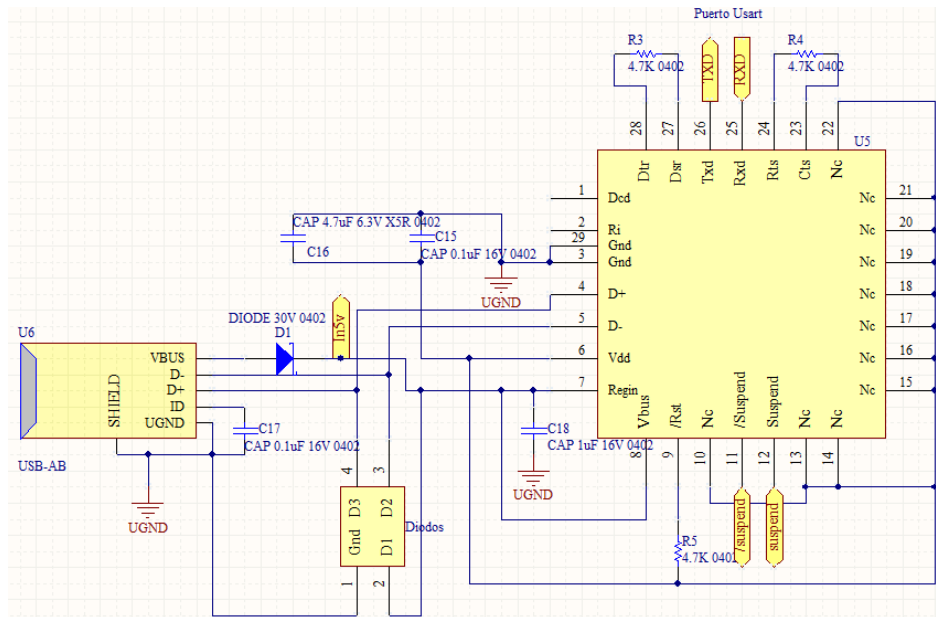


Figura 89 Diagrama de conexión del CP2102.

Se colocó un diodo Shockley (D1), como protección ante conexiones al revés del bus USB, aunque por razones de infraestructura, es francamente difícil conectarlo mal. Pero para prever algún fallo, se utilizó este tipo de diodo, por su velocidad y su poca atenuación de tensión en conducción (0.3v).

6.0 DISEÑO DE LA PCB

Para diseñar nuestro proyecto, se utilizó el programa Altium Designer Release 10, cuyo aspecto se puede apreciar en la 90.

Este software se encarga de la parte de diseño del hardware de nuestro proyecto, teniendo primero que hacer las bibliotecas de los componentes, luego interconectar todos los componentes y finalmente, colocarlos en nuestra placa.

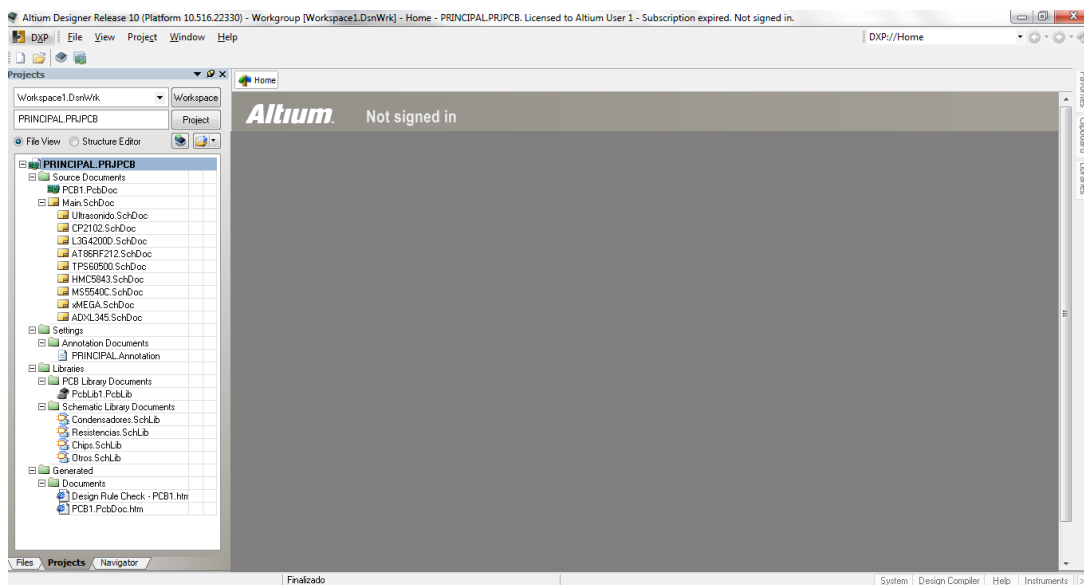


Figura 90 Programa Altium Designer Release 10.

El primer paso que se realiza es el de agregar las bibliotecas de diseño, que vienen incluidas en el programa. Estas bibliotecas contienen los componentes principales que se utilizan como son los condensadores, resistencias, diodos, bobinas, etc. Los componentes que no tenemos en las bibliotecas originales, los tenemos que crear manualmente.

6.1 Diseño de las bibliotecas

Hay dos tipos de bibliotecas que debemos diseñar, las bibliotecas graficas y las bibliotecas de los footprint.

6.1.1 Bibliotecas graficas

Estas bibliotecas son las que utilizamos para hacer nuestros esquemáticos. En estas bibliotecas haremos el dibujo o mejor dicho, el símbolo de los componentes que utilizamos. Cada elemento que utilizamos en el diseño tendrá que tener su representación.

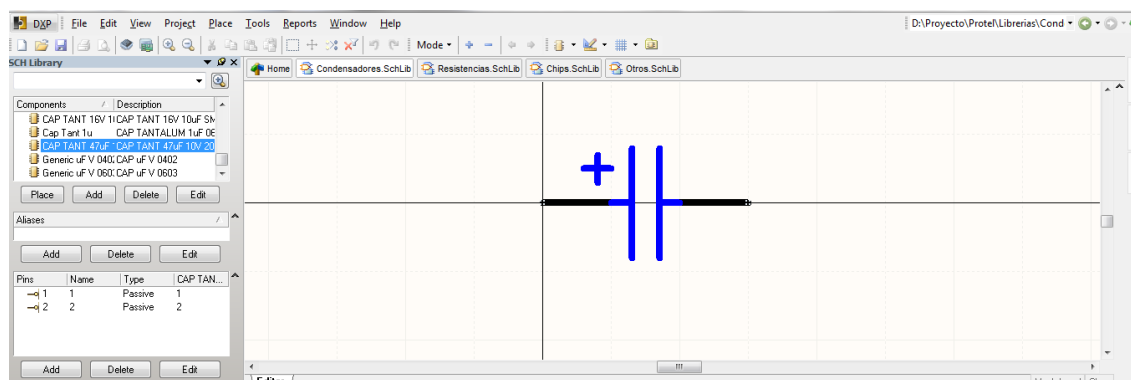


Figura 91 Símbolo de un condensador de Tántalo.

En la figura superior se ve el distintivo de un condensador con su polo positivo. Aparte de dibujar el símbolo, se le añade los pines que tendrá el elemento y se le puede añadir distintivos para a la hora de la soldadura saber la posición que tiene que tener el dispositivo en la PCB.

Para la creación de los componentes que no están incluidos en la biblioteca por defecto (Miscellaneous), se tiene que crear una nueva biblioteca, que será la que utilizaremos para los elementos específicos de nuestro proyecto.

En nuestro diseño creamos cuatro bibliotecas diferentes:

- Biblioteca de condensadores: incluimos todos los tipos condensadores.
- Biblioteca de resistencias: introducimos todas las resistencias empleadas.
- Bibliotecas de Chips: fijamos todos los IC que utilizamos para la realización del proyecto.
- Bibliotecas otros: en la que incluimos elementos como los buffer que utilizamos, el balun, la antena, etc.

Para completar los símbolos y poder pasarlos a los esquemáticos sin problemas, tenemos que hacer la otra biblioteca (biblioteca de footprint), para a cada símbolo, añadirle su correspondiente marca de soldadura.

6.1.2 Biblioteca de FootPrint

En la biblioteca de footprint realizamos la huella de los componentes que tendrá nuestro diseño. Cada componente tendrá su propio footprint, el cual es específico para algunos de ellos, teniendo un aspecto particular. Mientras que para otros este será genérico, compartiendo la misma huella.

Por ejemplo, los chips tienen una huella diferente, puesto que no todos tienen las mismas patillas, ni los mismos tamaños, ni el mismo tipo de encapsulado; sin embargo, los condensadores y resistencias, sí pueden compartir el mismo footprint, pues para ellos se ha estandarizado una serie de medidas, las cuales utilizan la mayoría de este tipo de componentes.

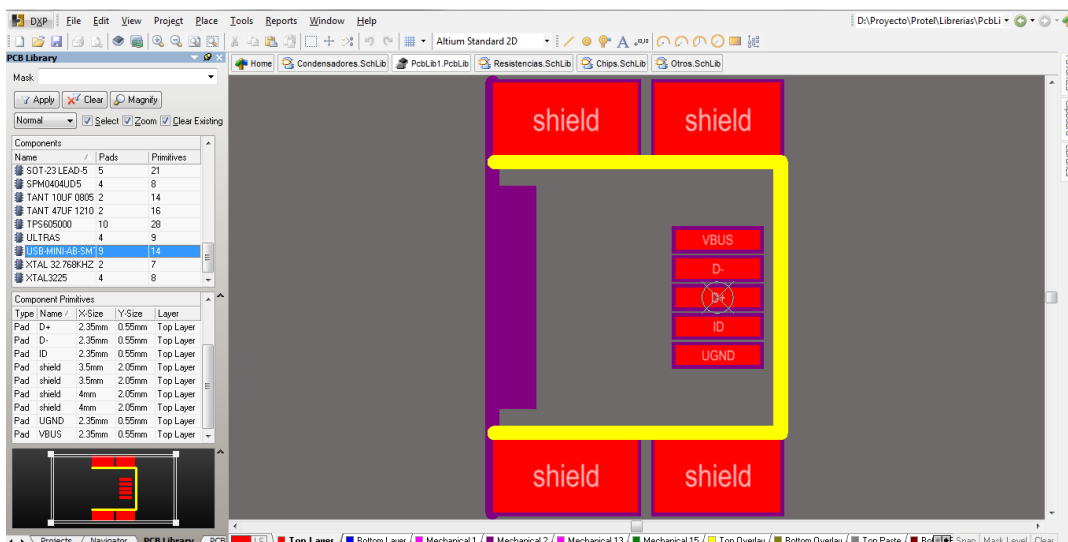


Figura 92 Footprint de un conector mini USB.

En la figura anterior, los recuadros rojos representan los pad que se verán en nuestra placa y en ellos luego se colocará el estaño, para soldar los pines de nuestro elemento.

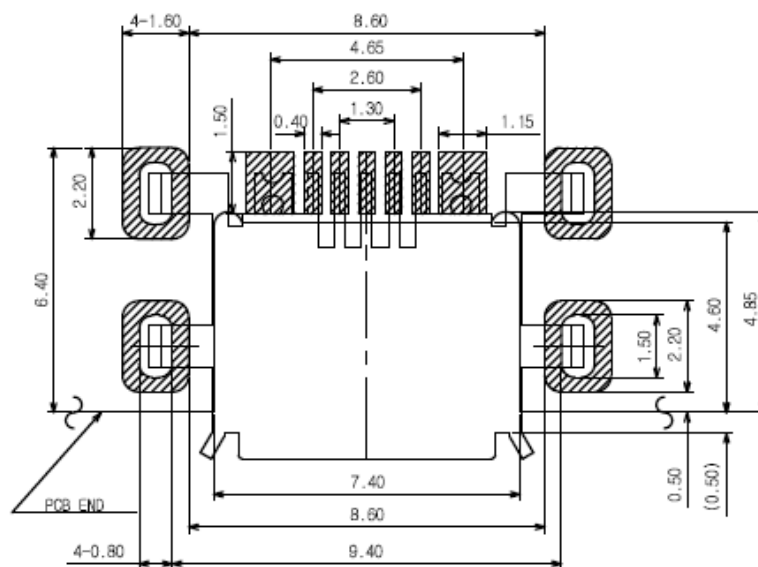


Figura 93 Medidas del conector mini USB.

En la figura 92 se aprecian los pad de un conector mini USB y se ve donde irían soldados los pad de este conector. Y en la siguiente figura 93, se observan las medidas que tienen que tener los pad y las distancias entre ellos, para que a la hora de soldar, coincida el footprint que nosotros diseñamos, con las patillas del elemento al que le diseñamos dicha huella.

Esta parte del diseño es muy especial, y hay que tener mucho cuidado a la hora de realizar el footprint, pues las medidas y los tamaños que diseñaremos de los pad, tienen que ser muy precisos. Si nos equivocamos, aunque sea mínimamente, del sitio donde debería ir el pad, puede que a la hora de soldar, tengamos que desechar nuestra PCB por no poder realizar tal operación.

Para que la soldadura nos sea un poco más factible, se deja el pad algo más largo de lo que en un diseño para fábrica se podría dejar. Esto es debido a que nuestra soldadura es manual y los elementos son extremadamente pequeños, complicándonos esta labor. El tamaño del pad se aumenta, para que al ponerle estaño, este fluya hasta la patilla del IC.

En un diseño para fabricación a gran escala, se puede reducir el tamaño de los pad, al mínimo necesario para soldar la patilla, pues el proceso con el que se sueldan los IC es mediante maquinaria industrial y se puede ajustar con más exactitud los tamaños de los pad.

Una vez realizados los footprints de los componentes, se interconectan con los símbolos creados en la biblioteca grafica, quedando la biblioteca completa de la manera que se ve en la figura 94.

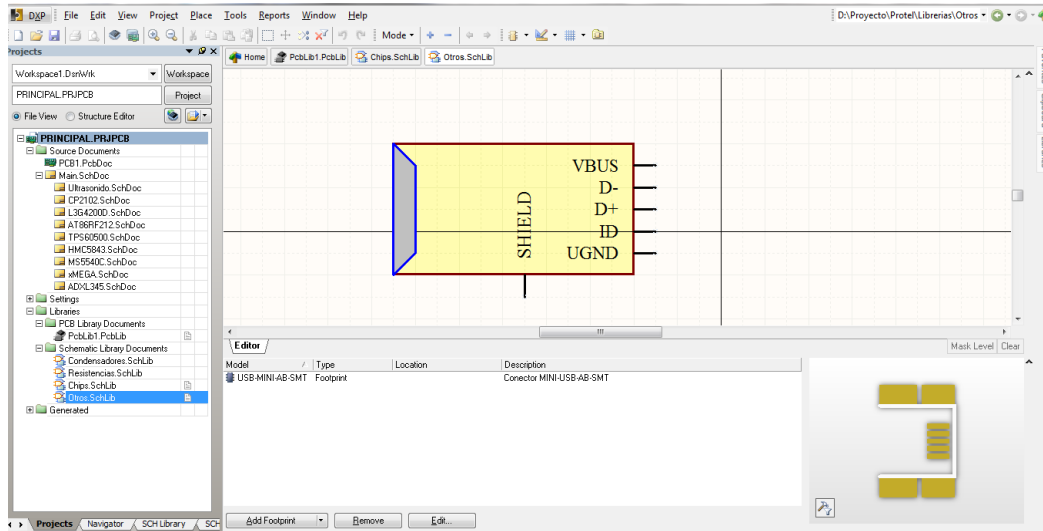


Figura 94 Biblioteca grafica incluyendo el footprint.

Una vez hecho esto, se pasa a realizar las conexiones de los componentes en los esquemáticos.

6.2 Esquemáticos

En los esquemáticos, realizamos las siguientes operaciones:

1. Colocamos el o los elementos que necesitamos utilizar y que hemos definido en nuestras bibliotecas.
2. Hacemos las interconexiones que necesita este elemento para funcionar correctamente.

Para colocar los elementos de las bibliotecas, basta con crear una página en donde realizar nuestro esquemático, Project → Add New Project → Schematic, este proceso creará una página en blanco, en donde cargar nuestros IC, condensadores, resistencias, etc.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

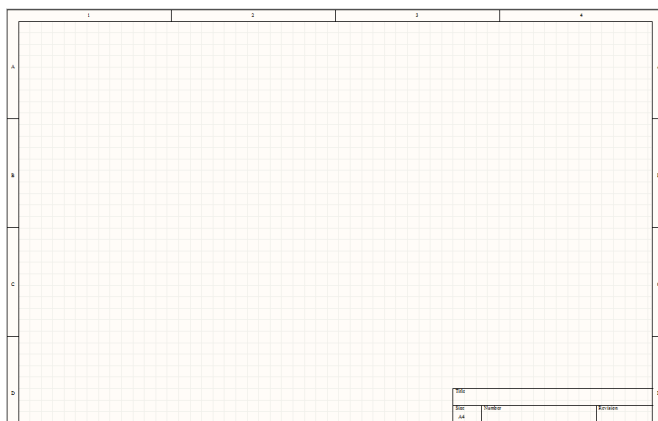


Figura 95 Página en Blanco de esquemáticos.

En la nueva página que hemos creado, introducimos los elementos que necesitamos haciendo Place → Part → Choose.

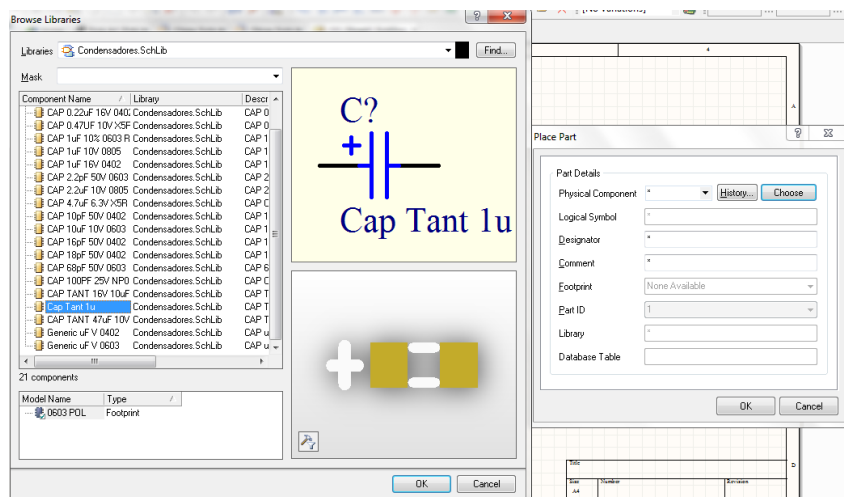


Figura 96 Introducción del componente.

Cuando se introducen todos los elementos que necesitamos, conexionamos los símbolos en Place → Wire o en los iconos que nos aceleran el trabajo.

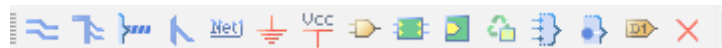


Figura 97 Iconos.

Los pines que van conectados a otros esquemáticos se unen mediante el PLACE PORT que es una especie de enlace entre todos los esquemáticos, y que con sólo ponerle el mismo nombre, se conectan directamente entre ellos.

Una vez conectados todos los dispositivos que van a ir en el esquemático, quedarían de la siguiente forma:

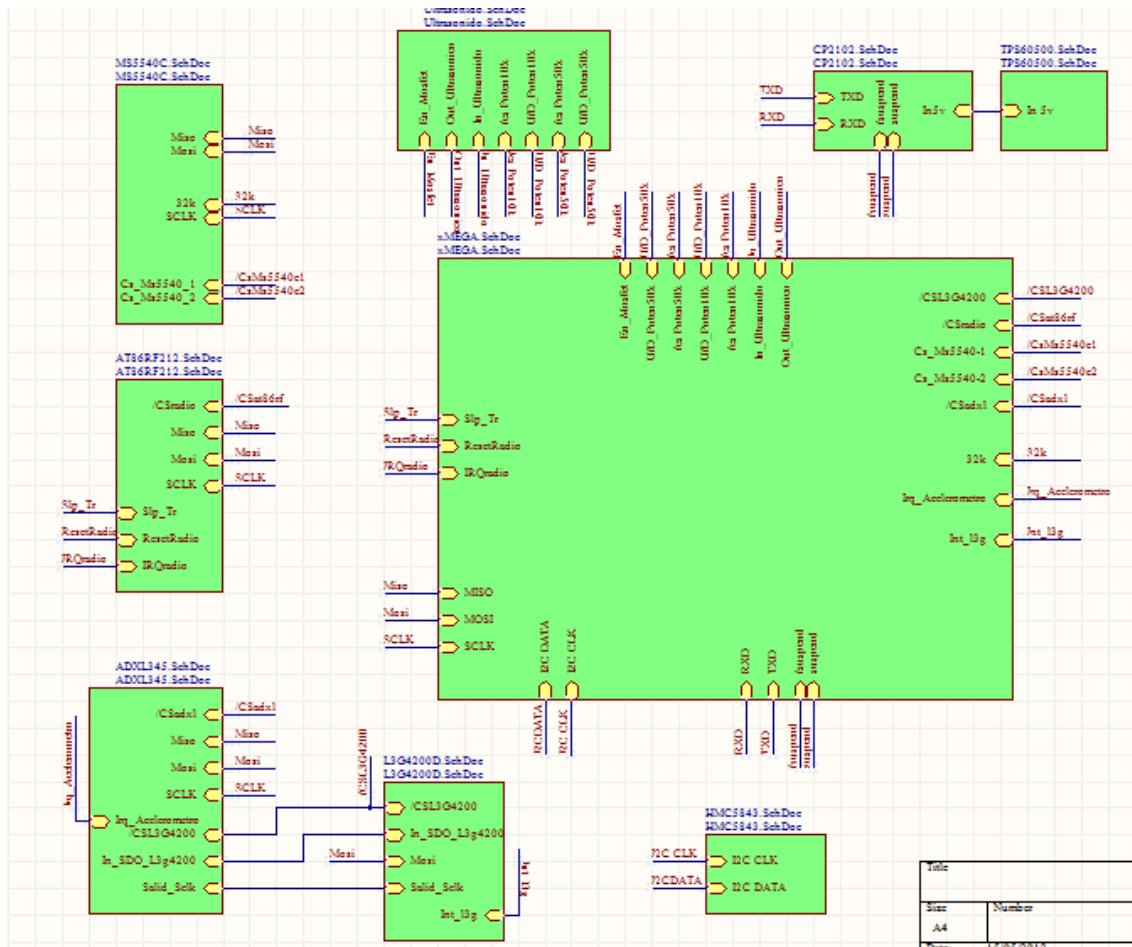


Figura 98 Esquemático General.

6.3 Paso de los esquemáticos a la PCB

Cuando tengamos todos los esquemáticos completados, pasamos a introducir todos los footprint de los elementos que hemos conectado a nuestra PCB. Para ello realizamos los siguientes pasos:

Vamos a PROYECT → ADD NEW TO PROYECT → PCB

Con esto tendremos nuestra placa PCB, con las capas TOP Y BOTTOM. Seguidamente, tenemos que importar los footprint de los esquemáticos a nuestra placa de la siguiente manera:

DESING →IMPORT CHANGE FROM “nombre de nuestro proyecto”.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

En la figura 99 se puede ver cómo quedan todos los elementos que formarán parte de nuestra PCB, interconectados de la manera en que lo están en los esquemáticos diseñados anteriormente.

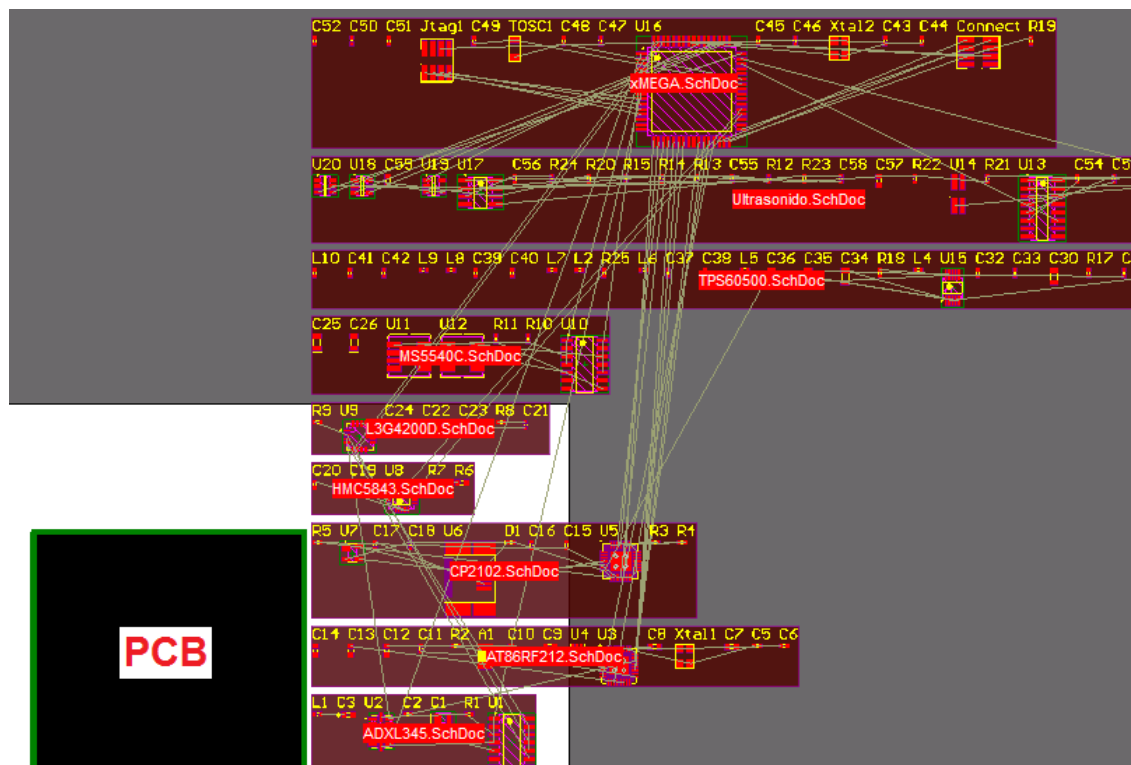


Figura 99 Footprints importados a la PCB

Antes de ponernos a colocar cada componente en el interior de la PCB y de empezar su conexión mediante los enrutamiento y las vías, tenemos que fijarnos en las reglas de diseño que hay que cumplir para que los fabricantes de placa puedan construirla de manera satisfactoria.

6.3.1 Reglas de diseño

Con estas reglas nos aseguramos que cumplimos las limitaciones tecnológicas impuestas por el fabricante de la PCB. Dependiendo del fabricante y de la tecnología que utilicemos, las reglas de diseño cambiarán. Las reglas de diseño que nos afectan a nosotros vienen marcadas por el fabricante de nuestra PCB, como dijimos anteriormente, y son las siguientes:

Características	Disponibilidad
Capas	1 – 10 Capas
Material	FR4 (Fibra de vidrio y polipropileno)
Tamaño máximo	600x700mm
Min. Diámetro de la Via	0.2mm
Min. Ancho de Línea	0.2mm
Min. Espacio Entre Líneas	0.2mm
Min. Recubrimiento de Pad	0.4mm

Tabla 9 Reglas de Diseño.

Una vez cumplidas todas las reglas de diseño, pasamos a posicionar los IC, elementos pasivos y elementos activos. Este paso es decisivo, pues la situación y la colocación de cada elemento en su sitio, es extremadamente compleja.

6.3.2 Posicionamiento y Enrutamiento

Para ubicar los componentes debemos de empezar por los más grandes y con más conexiones y luego ir colocando los más pequeños y con menores conexiones. Así conseguimos ocupar el menor espacio posible en la PCB, reduciendo costes de esta manera, pues cada mm de más que desperdiciemos, incrementa el coste.

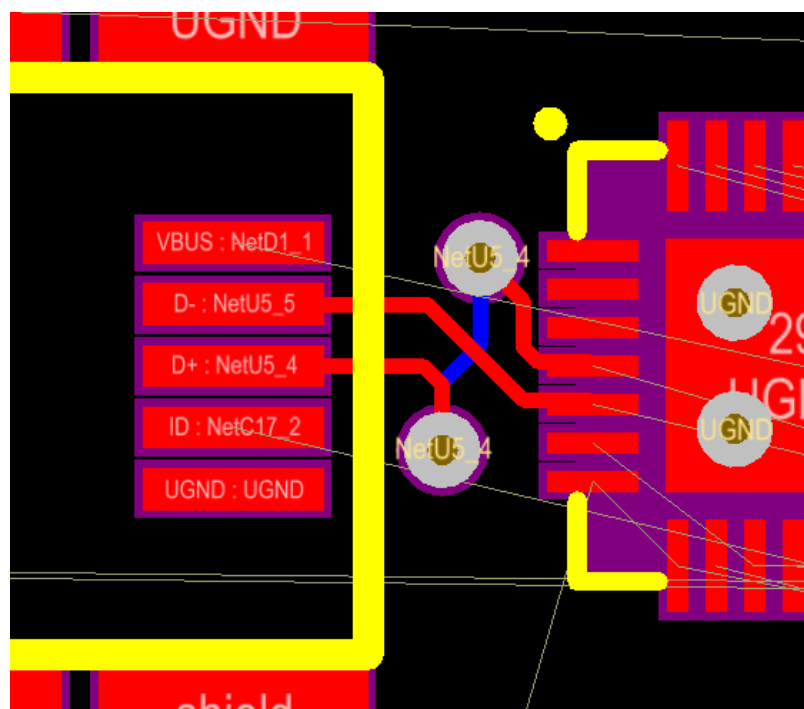


Figura 100 Ejemplo de conexión Bus USB.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

En la figura 100 se observa un pequeño ejemplo de conexión del puerto mini USB a nuestro CP2102. Como se aprecia, las conexiones están cruzadas, siendo imposible conectarlas sin utilizar las vías.

Las vías las utilizamos para pasar de la capa TOP a la capa BOTTOM y viceversa, pudiendo así cruzar líneas sin que se produzcan cortocircuitos. Debemos de utilizarlas lo menos posible, pues el número de vías incrementa también el coste, además de ocupar más espacio en nuestra PCB.

Lo más lógico para posicionar los elementos de nuestro diseño, es colocar los IC y los elementos difíciles de soldar en la capa TOP, pues en esta capa se utilizará la soldadura al horno, que sólo se puede realizar en una cara. La capa BOTTOM se deja para elementos como resistencias, condensadores, bobinas y el resto de elementos que no pudieron ser posicionados en la capa TOP.

Aparte de las capas TOP y BOTTOM, podemos crear más capas, las cuales llevarán el plano de masa y alimentación, figura 101.

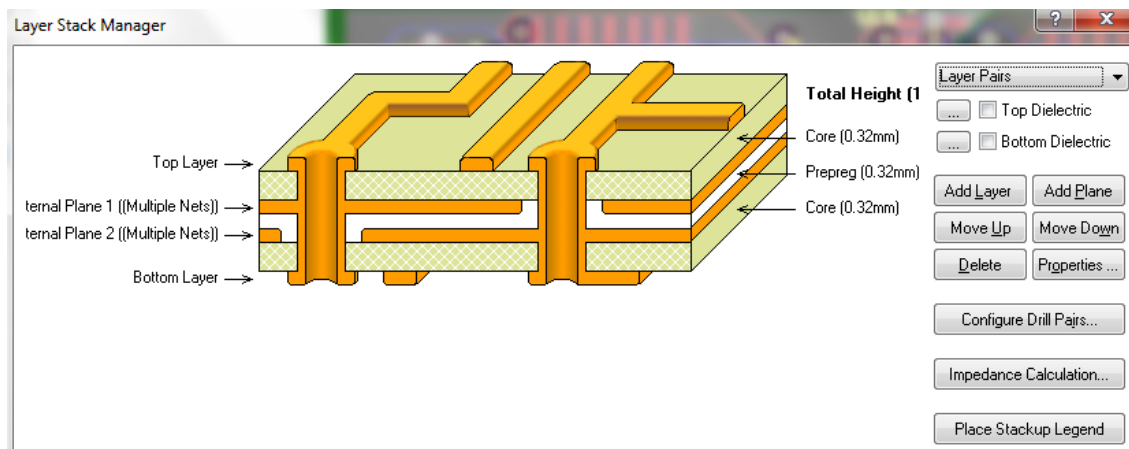


Figura 101 Capas del diseño.

Estas capas son internas y llegamos a ellas a través de las vías, las cuales se conectarán al plano de alimentación o de tierra si es necesario; sino atravesarán los planos internos sin conectarse a ellas. Gracias a estos planos internos, podemos llevar la alimentación o la masa a cualquier sitio de nuestra placa.

Una vez que tengamos todos nuestros componentes colocados y enrutados tratando de ocupar el menor espacio posible, pasamos a recortar y dejar la placa

lista para pasar a su fabricación. Para ello vamos a DESIGN → BOARD SHAPE → REDEFINE BOARD SHAPE, con esto limitamos el área de nuestra placa quedando con el tamaño adecuado y la forma que deseemos darle.

Nuestra PCB quedó con un tamaño considerablemente pequeño, pues sus medidas son de 42.79mm x 39.531mm. El diseño final que se envió a fábrica se puede ver en la figura 102.

Se puede apreciar en nuestro diseño, lo comprimido que han quedado todos los componentes, sin dejar huecos y espacios vacíos que harían que nuestra PCB fuera mucho más grande y los gastos de producción de la misma se incrementarían considerablemente.

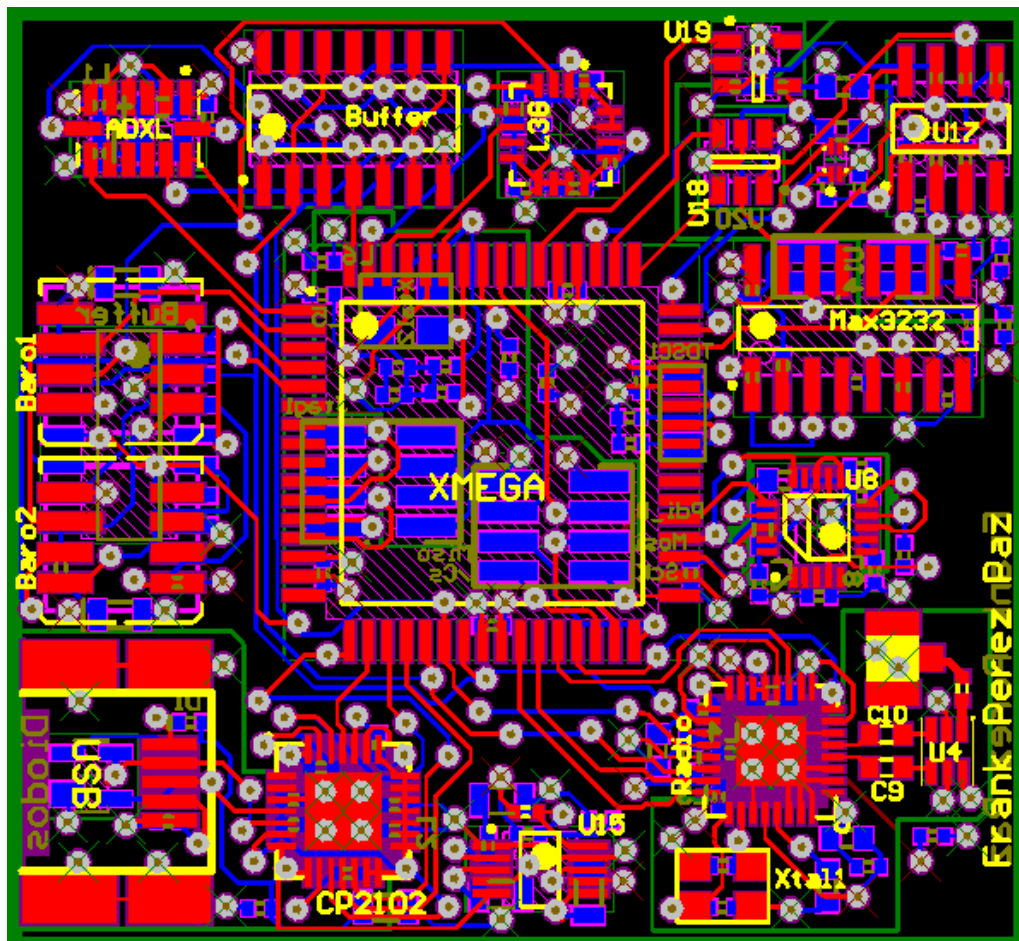


Figura 102 Diseño de nuestra PCB.

7.0 FABRICACIÓN DE LA PCB

China es el país en el que se fabrican gran parte de los productos electrónicos, debido a la calidad que ofrecen en sus productos y a los precios considerablemente bajos.

Gracias a los grandes avances tecnológicos de fabricación que tiene China, les es más fácil y menos caro fabricar y ensamblar los diseños que les mandan a construir las grandes marcas del mercado.

Las grandes marcas en electrónica como SONY, SAMSUNG, PHILIPS, etc. fabrican íntegramente o los elementos principales de sus equipos electrónicos. Por ejemplo SONY una de las empresas más conocidas, producen sus paneles de televisión allí.

Nuestra placa PCB se mandó a fabricar a China pero como es un prototipo, solo se fabricó la placa, no se ensamblaron los componentes ya que encarecería muchísimo el presupuesto. Posteriormente nosotros nos tuvimos que encargar de su ensamblaje completo, soldando todos los elementos que componen el proyecto.

7.1 Soldadura

Para soldar los IC así como otros componentes de nuestro diseño, se utilizaron diferentes métodos de soldadura, pues dependiendo del tipo y del tamaño del encapsulado de los componentes, se puede emplear un método u otro, según convenga.

Para soldar la capa TOP manejamos varias máquinas de colocación del componente y luego utilizamos soldadura al horno. Para soldar la capa inferior

(BOTTOM) se utilizó un soldador, así como los equipos necesarios para poder ver los componentes que se estaban soldando.

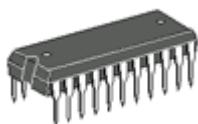
7.2 Tipos de encapsulados

Para cubrir el chip de agresiones externas como la humedad, el polvo, la corrosión, etc., a los IC se les coloca un encapsulado que los aíse y a la misma vez les permita la conexión al exterior. Además de hacerlos más robustos para su manipulación, favorecen la disipación del calor.

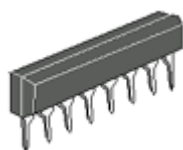
Existen 2 clasificaciones para los encapsulados. Los encapsulados IC, que son para los circuitos integrados, y los encapsulados discretos que son para las resistencias, condensadores, diodos, transistores, etc.

Tipos de Encapsulados				
IC		Discreto		
Inserción	Montaje Superficial	De inserción	Montaje Superficial	
DIP	SOP	SP-8	SC-59	TO-254
SIP	TSOP	SST	SC-62	TO-263
PGA	QFP	TO-3	SC-70	HVSON
	SOJ	TO-92	SC-74	HWSON
	QFJ	TO-126	SC-75	XSOFF
	QFN	Is. TO-220	SC-84	SOP8
	TCP	TO-220AB	SC-88	TSSOP
	BGA	TO-251	SC-89	MLP
	LGA		SC-95	EFLIP

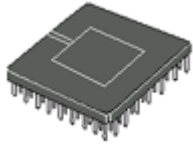
Tabla 10 Tipos de encapsulados.



DIP: Los pines se extienden a lo largo del encapsulado (en ambos lados) y tiene como todos los demás, una muesca que indica el pin número 1. Este encapsulado básico fue el más utilizado hace unos años.



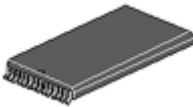
SIP: Los pines se extienden a lo largo de un solo lado del encapsulado y se monta verticalmente en la placa.



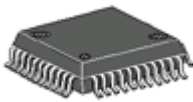
PGA: Los múltiples pines de conexión se sitúan en la parte inferior del encapsulado. Este tipo se utiliza para CPUs de PC y era la principal opción a la hora de considerar la eficiencia pin-cápsula-espacio antes de la introducción de BGA.



SOP: Los pines se disponen en los 2 tramos más largos y se extienden en una forma denominada “gull wing formation”, este es el principal tipo de montaje superficial.



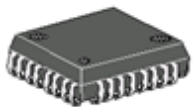
TSOP: Simplemente una versión más delgada del encapsulado SOP.



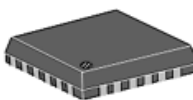
QFP: Es la versión mejorada del encapsulado SOP, donde los pines de conexión se extienden a lo largo de los cuatro bordes. Este es en la actualidad el encapsulado de montaje superficial más popular, debido a que permite un mayor número de pines.



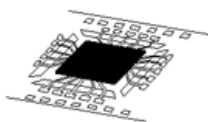
SOJ: Las puntas de los pines se extienden desde los dos bordes más largos dejando en la mitad una separación como si se tratase de 2 encapsulados en uno. Recibe éste nombre porque los pines se parecen a la letra “J” cuando se mira desde el costado.



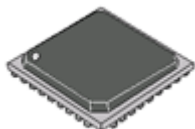
QFJ: Al igual que el encapsulado QFP, los pines se extienden desde los 4 bordes.



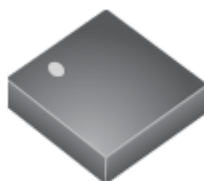
QFN: Es similar al QFP, pero con los pines situados en los cuatro bordes de la parte inferior del encapsulado.



TCP: El chip de silicio se encapsula en forma de cintas de películas. Se puede producir de distintos tamaños y el encapsulado puede ser doblado.



BGA: Los terminales externos, en realidad esferas de soldadura, se sitúan en formato de tabla en la parte inferior del encapsulado. Este encapsulado puede obtener una alta densidad de pines.



LGA: Es un encapsulado con electrodos alineados en forma de array en su parte inferior. Es adecuado para las operaciones donde se necesita alta velocidad debido a su baja inductancia. Además, en contraste con el BGA, no tiene esferas de soldadura por lo cual la altura de montaje puede ser reducida.

7.3 Proceso de Soldadura.

Para fabricar la máscara de soldadura se utiliza una máquina que realiza huecos en la misma. Los huecos los abre en donde van a ir soldados los pad, para que de esta forma, al poner la pasta de soldadura, esta quede impregnándolos. Debido a esto, donde no haya pad, no se depositará la pasta.

Una vez tengamos la máscara, el primer paso es alinearla con nuestra PCB. Esto se hace manualmente, y se trata de conseguir un alineado perfecto para que la pasta quede justamente en los pad. Una vez que hayamos conseguido esto, debemos fijar nuestra máscara por un lado, de manera que una vez impregnada por la soldadura, se pueda levantar para sólo dejar la pasta en los pad de la PCB. En la figura 103 se puede ver cómo está la máscara en la PCB de forma alineada y cómo se puede levantar esta sin inconveniente.

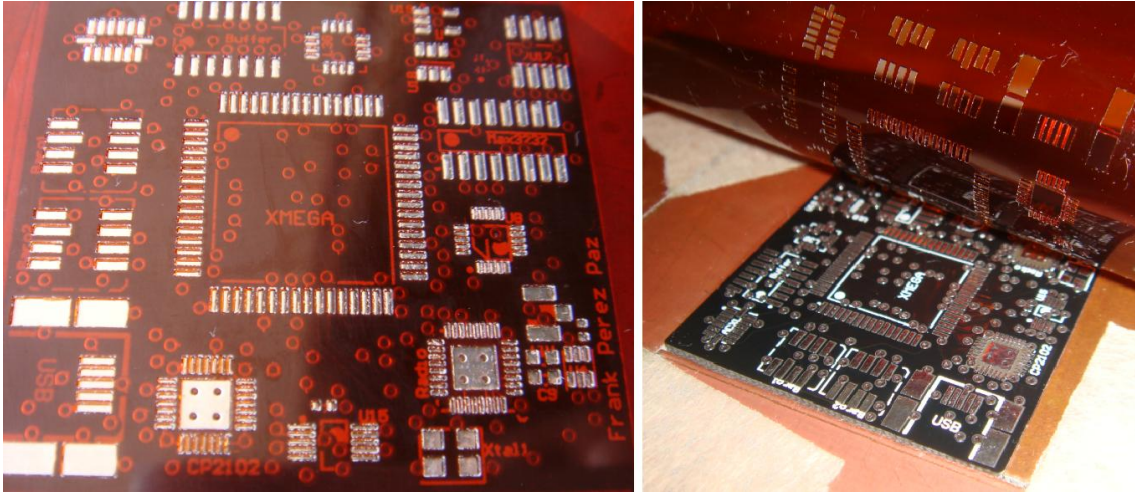


Figura 103 Mascara de Kapton y placa PCB.

Para realizar esto, se colocó nuestra PCB a una placa más grande, y la máscara de soldadura se fijó por la parte superior con cinta adhesiva. Con esto logramos que cuando empastáramos, pudiéramos levantar la máscara y comprobar que todos los pad hubieran cogido pasta. Si esto no ocurría o la cantidad de pasta era insuficiente, podíamos realizar de nuevo el proceso, sin necesidad de volver a alinearla.

Cuando todos los pad estuvieron empastados correctamente, pasamos a colocar los IC y componentes que iban en nuestra capa TOP. Para ello nos servimos de diferentes aparatos que nos ayudaron a la correcta colocación.

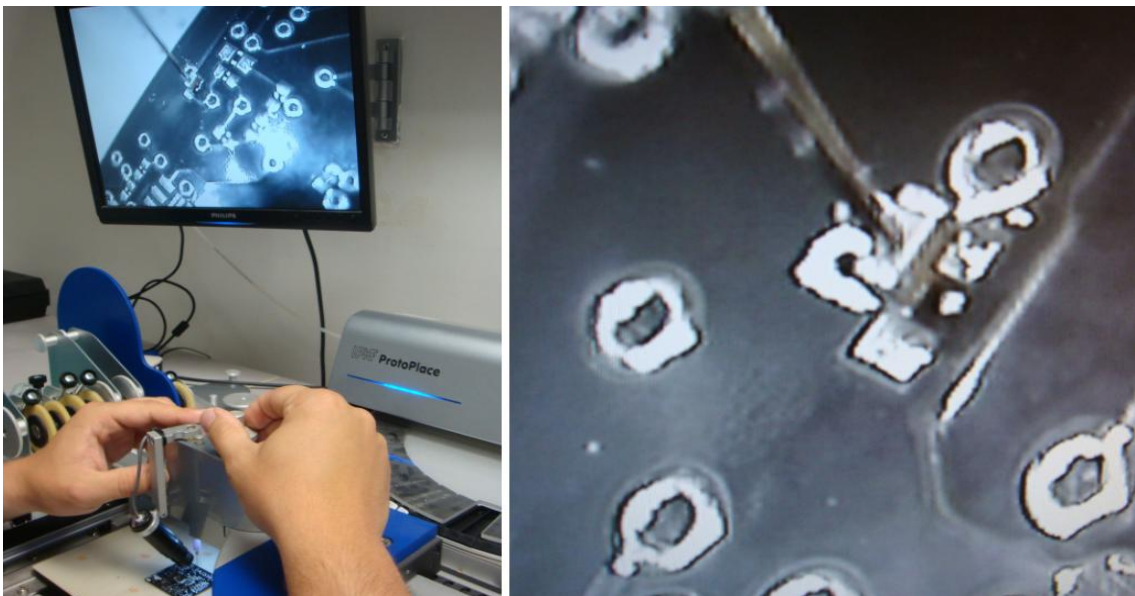


Figura 104 Maquina de succión.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

En la máquina que se ve en la figura 104 se observa cómo una aguja sujeta un condensador por medio de la succión del aire, y cómo nosotros manualmente, llevamos nuestro componente hasta donde va a ser soldado.

Estas máquinas conseguían depositar los IC en el lugar que correspondía, puesto que al tener pasta los pad, el componente tiene que ser depositado sin ningún movimiento lateral para no desplazar la pasta en este proceso.

Para algunos componentes cuyo encapsulado era del tipo QFN, LGA o MLP, se necesitó una máquina que por medio de un haz de luz, nos indicaba dónde estaban los pines de estos encapsulados y donde se colocarían en la PCB. Esta máquina es semiautomática, pues sólo teníamos que colocar la placa PCB debajo de ésta, moviéndola hasta que coincidieran los pad con el haz de luz. Una vez conseguido esto, se quedaba fija la placa y el componente descendía y se depositaba justamente encima de los pad, donde se encontraba la pasta de soldadura. Figura 105.

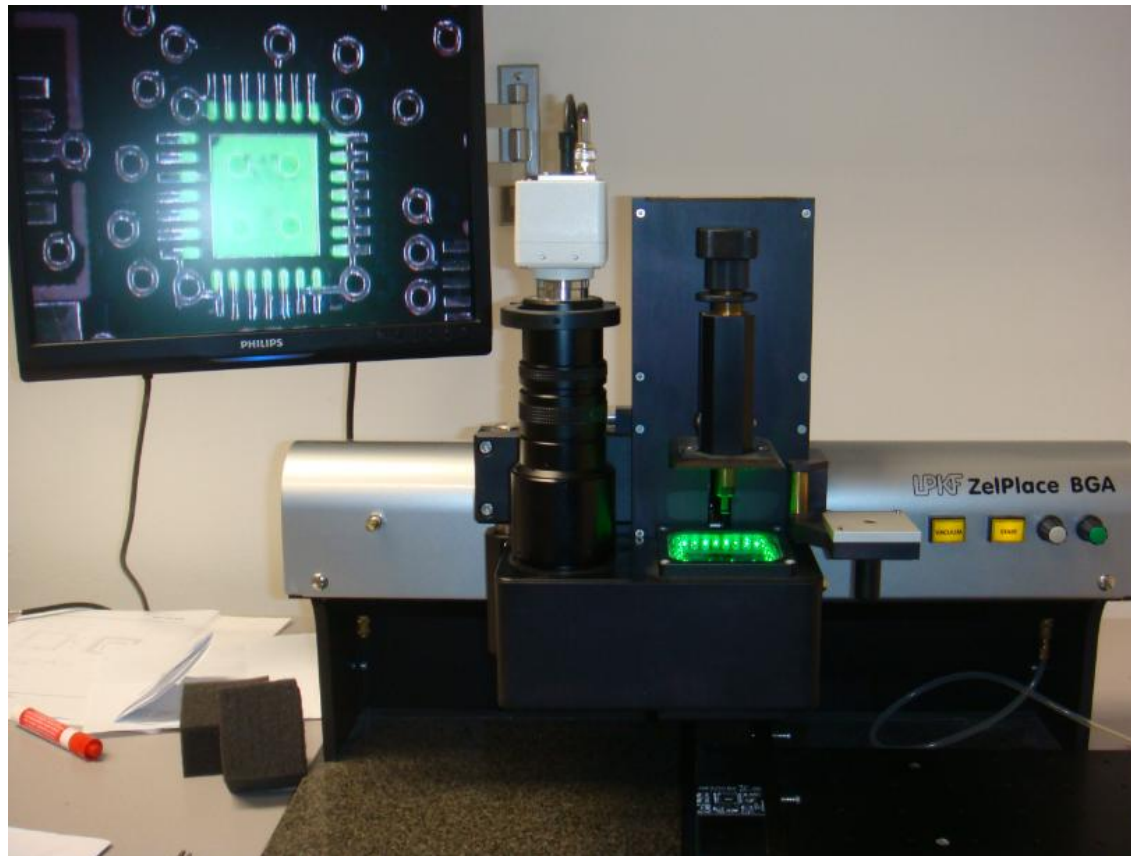


Figura 105 Máquina LPKF ZelPlace BGA.

Esta máquina utiliza un sistema de alineación óptica con cámaras CCD, pudiendo nosotros saber dónde depositará nuestro componente SMD en la placa PCB. Además tiene un plato mecanizado en donde poner el componente para que sea sujetado por el sistema de colocación, el cual tiene un brazo neumático que baja el componente al sitio indicado.

Una vez todos los componentes que van soldados en la capa TOP estaban colocados, se pasa con extremo cuidado la PCB hasta un horno. Este horno (LPKF ProtoFlow) se puede calentar hasta los 320 °C, para que la pasta de soldadura se derrita y suelde los pad a las patillas de los componentes. En unos pocos minutos se saca la placa del horno y se observa cómo la capa TOP está completamente soldada.



Figura 106 LPKF ProtoFlow.

Si quedan los componentes sin soldar correctamente o la pasta no permanece el suficiente tiempo en el horno, se puede volver a introducir en el mismo nuestra placa. Si algún pad queda sin soldar con la patilla del componente, se puede calentar un poco y estañarlo para soldarlo mejor manualmente.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

Para soldar la capa BOTTOM, en donde irán situados la mayoría de los componentes pasivos de nuestro diseño y algún IC que no se colocó en la TOP, debemos emplear un soldador manual.

Para soldar, siempre se ha utilizado el plomo pues tiene una temperatura inferior que el estaño para derretirse y de esta manera los pines de los componentes no sufren mucho a la hora de su soldadura. Sin embargo, con la introducción de la directiva RoHS/WEEE se prohibió la utilización de elementos con plomo, por lo tanto la soldadura se realiza con estaño sin plomo, siendo más problemático este elemento a la hora de derretirse, pues necesita mayores temperaturas.

Como estos componentes son muy pequeños, necesitamos un equipo de microscopio para poder observar mejor la soldadura, y comprobar si el componente se ha soldado correctamente o no. En la siguiente figura 107 se puede ver el proceso de soldadura manual.

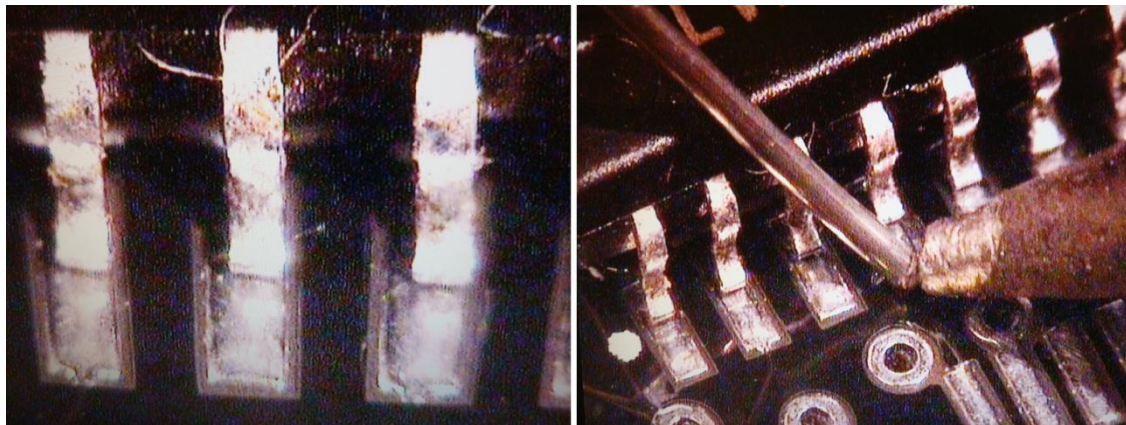


Figura 107 Vistas del microscopio.

Cuando conectamos algún componente y queremos desoldarlo, utilizamos una maquina de aire caliente. Este aire hace que el estaño se derrita y que todas las patillas del componente adquieran idéntica temperatura. Finalmente con este proceso, se desprenden los pad. El aire caliente adquiere gran importancia a la hora de quitar componentes de muchos pines o componentes tipo QFN o LGA en donde no podemos acceder a las patillas.

8.0 PROGRAMACIÓN

Hoy en día la programación forma parte de la mayoría de los dispositivos electrónicos que rodean nuestras vidas. Desde un ordenador, un robot e incluso hasta una lavadora contiene programación. Todas las máquinas que contengan un microcontrolador o un procesador, necesitan de un software para funcionar aparte del hardware. Es más, la programación es el software que controla el hardware de nuestro dispositivo.

La programación es lo que hacemos las personas para poder controlar las acciones de los controladores y procesadores. Para ello utilizamos distintas clases de programación como son el C, C++, Java, lenguaje ensamblador, vhdl, etc. dependiendo hacia qué tipo de máquina nos orientamos a programar, se utilizará un programa u otro.

Al programar, escribimos un código (instrucciones) en el cual, estructuramos una o varias acciones que tiene que realizar nuestro controlador, para que una vez depuremos el programa, ensemblemos y compilemos, nuestro micro entienda este código y haga la acción que nosotros le hemos programado.

8.1 Programa Utilizado

Para programar nuestro ATXMEGA128A3 se utilizó el programa de programación AVR STUDIO. Este está especialmente diseñado para los microcontroladores AVR y nos permite no sólo programar sino además admite editar, ensamblar y simular el código fuente.

El entorno del programa se puede ver en la figura 108.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

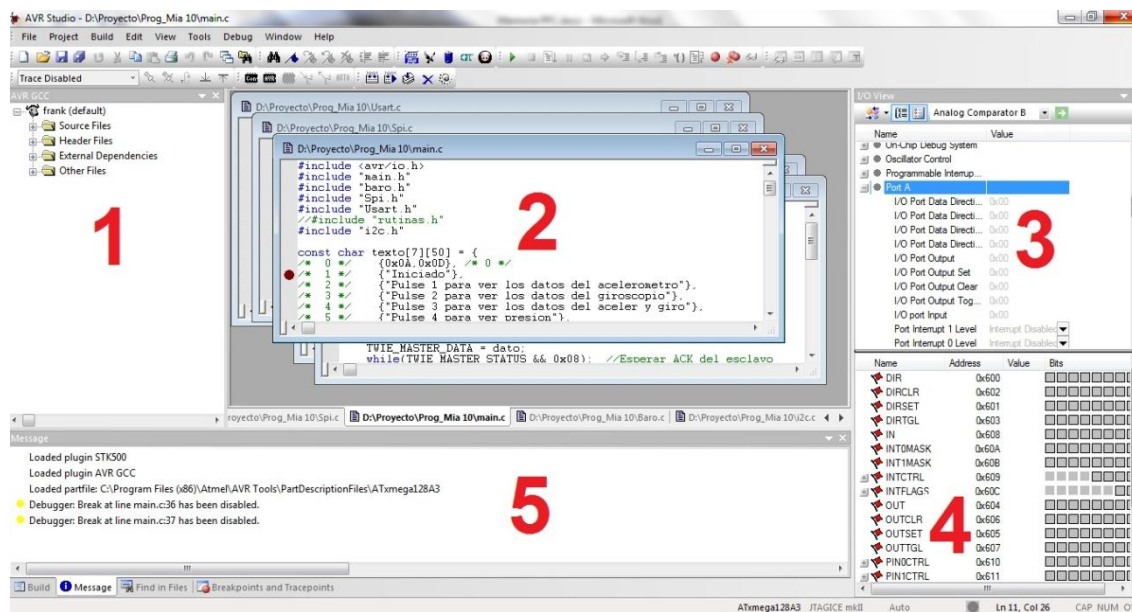


Figura 108 Entorno programa AVR-STUDIO.

Se puede apreciar cómo tiene cinco ventanas. Cada ventana hace una función diferente. Describiremos cada ventana a continuación:

1. La ventana uno es en donde se cargan los ficheros que utilizamos en la programación. Aquí se suben los archivos “.c, .h”; también se pueden cargar otros archivos como son los “.arm”, que se utilizan para otros micros diferentes.
2. En la ventana dos escribimos nuestro código de programación. En ella podemos visualizar los archivos que tenemos en la ventana uno.
3. Esta ventana carga todos los puertos, relojes, buses, interrupciones, convertidores ADC y DAC, etc. que tenga nuestro microcontrolador. Podemos ver todas las configuraciones que tienen los puertos, las velocidades con la que tenemos que cargar el reloj para una determinada frecuencia de funcionamiento, etc.; es decir, podemos controlar el funcionamiento de nuestro micro mediante esta ventana.
4. Ésta está relacionada con la ventana anterior, pues es una extensión de los registros y configuraciones que tenemos en la ventana tres. Es decir, si

miramos los puertos en esta última, en la ventana cuatro tenemos los registros de configuración de los puertos.

5. En esta ventana, tendremos información sobre errores, avisos y puntos de ruptura en nuestro programa. A la hora de compilar será la que nos diga si nuestro programa tiene algún fallo de programación, en dónde tenemos esos fallos y los puntos de ruptura, si los hubiese.

Además, este programa contiene todos los registros de los microcontroladores que trabajan con el entorno AVR y al seleccionar el controlador que estamos utilizando nos incluye directamente todos los registros y puertos.

Para configurar nuestro microcontrolador en el programa, debemos de incluirle las bibliotecas con las que trabajaremos. Estas bibliotecas se observan en la figura 109.

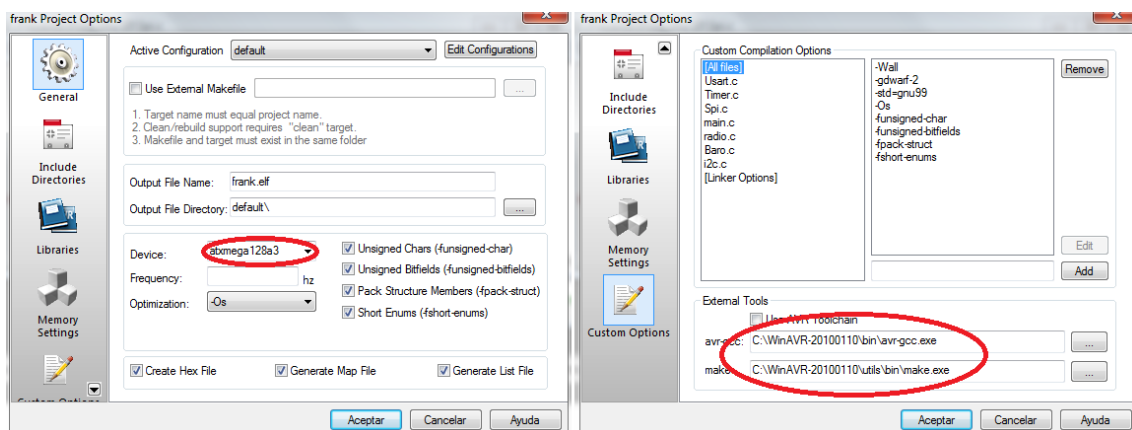


Figura 109 Opciones generales y librerías.

Una vez que tengamos nuestras librerías, y hayamos realizado nuestra programación, lo compilamos para que se cree el programa que entenderá nuestro microcontrolador para operar.

Lo que se hace al compilar es crear una interface, la cual, a partir del programa con el que trabajamos nosotros, que es el C, crea el programa que

entienden los controladores. Al compilar se crea el archivo “.hex”, que contendrá toda nuestra programación, pero en el lenguaje en el que trabaja el controlador.

8.2 Dispositivo JTAG



Figura 110 Programador JTAG.

El dispositivo que utilizamos para comunicar nuestro programa AVR-STUDIO con nuestro microcontrolador fue el programador KEE JTAGICE mkII.

Este programador cuenta con todo el protocolo de comunicación JTAG, con lo cual es capaz de crear una interface entre nuestro AVR y nuestro puerto JTAG del microcontrolador.

Soporta un amplio rango de tensión de alimentación que va desde los 1.8v hasta los 5.5v, incluyendo una protección frente a alimentaciones incorrectas o sobre voltajes.

También es compatible con los dispositivos de depuración XMEGA a través de la interfaz PDI que implementan estos micros.

En la foto 110 observamos nuestro programador JTAG MkII, al cual se le adaptó un pequeño conector para acoplarlo a nuestro header, pues este tiene una separación entre pines (pitch) de 1.27mm.

El protocolo del JTAG se basa en el estándar 1149.1-2001, el cual define las instrucciones IDCODE, BYPASS, SAMPLE/PRELOAD y EXTEST. Éstas sirven para testear y comprobar el funcionamiento de nuestro microcontrolador.

Consta de diez pines, los cuales se utilizan para la comunicación (sólo son útiles ocho pines, los otros dos no se conectan). Los pines se pueden ver en la tabla 11.

Pin	Descripción
TDI	Entrada de datos
TDO	Salida de datos
TCK	Reloj
TMS	Selector de modo
TRST	Reset

Tabla 11 Pines del puerto JTAG.

Los tres pines que faltan son dos pines de tierra, y un pin de alimentación. La tensión para alimentar nuestro dispositivo JTAG la recibe directamente del puerto USB.

En la siguiente figura se puede ver el conector del JTAG y el adaptador que tuvimos que utilizar para que se acoplara a los pines de nuestro proyecto.

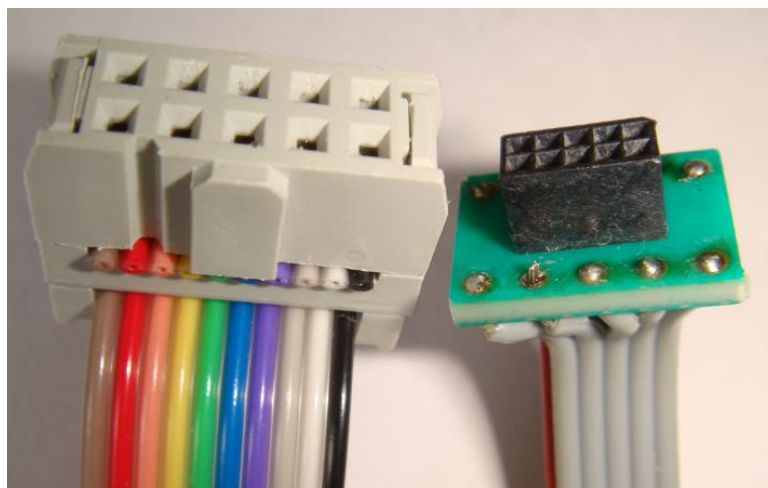


Figura 111 Conector de JTAG de 10 Pins

8.3 Pruebas Pre-programación

Para comprobar que nuestra placa de circuito impreso está lista para la programación y para ser conectada, primero comprobamos que todos los componentes están correctamente soldados y que todos los integrados reciben la tensión correcta de trabajo.

Para ello antes de conectar la alimentación se comprobó si existían cortocircuitos en los planos de masa y alimentación, así como en los sitios en los que pudiéramos sospechar que la soldadura hubiese quedado en mal estado.

Para realizar las pruebas cogimos un polímetro y medimos la continuidad de un plano a otro, para comprobar que no estaban conectados y por lo tanto no existían cortocircuitos.

Una vez comprobado lo anterior, conectamos la placa a la alimentación del USB y se observó que la alimentación para la que fue diseñada nuestra PCB concordaba con la alimentación que teníamos en la realidad. Si la tensión variaba ± 0.1 v se consideraba que estaba correcta.

Además se analizó la temperatura de cada integrado para comprobar que todos tenían la temperatura normal de trabajo. Si había alguno a una temperatura anormal, quería decir que no estaba operando correctamente, teniendo que pasar a ser revisado minuciosamente.

Cuando las tensiones y temperatura estuvieron correctas, se procedió a instalar los driver de nuestro dispositivo USB para ver si nuestro PC reconocía nuestro integrado CP2102.

Una vez que nuestro PC reconoció nuestro dispositivo, ya se procedió a conectar el programador y a programar nuestro microcontrolador.

8.4 Descripción del Software

Para que nuestro dispositivo funcionara, se desarrolló un software, el cual realiza de forma cíclica la tarea de leer los sensores y de mandar los datos por el puerto USART.

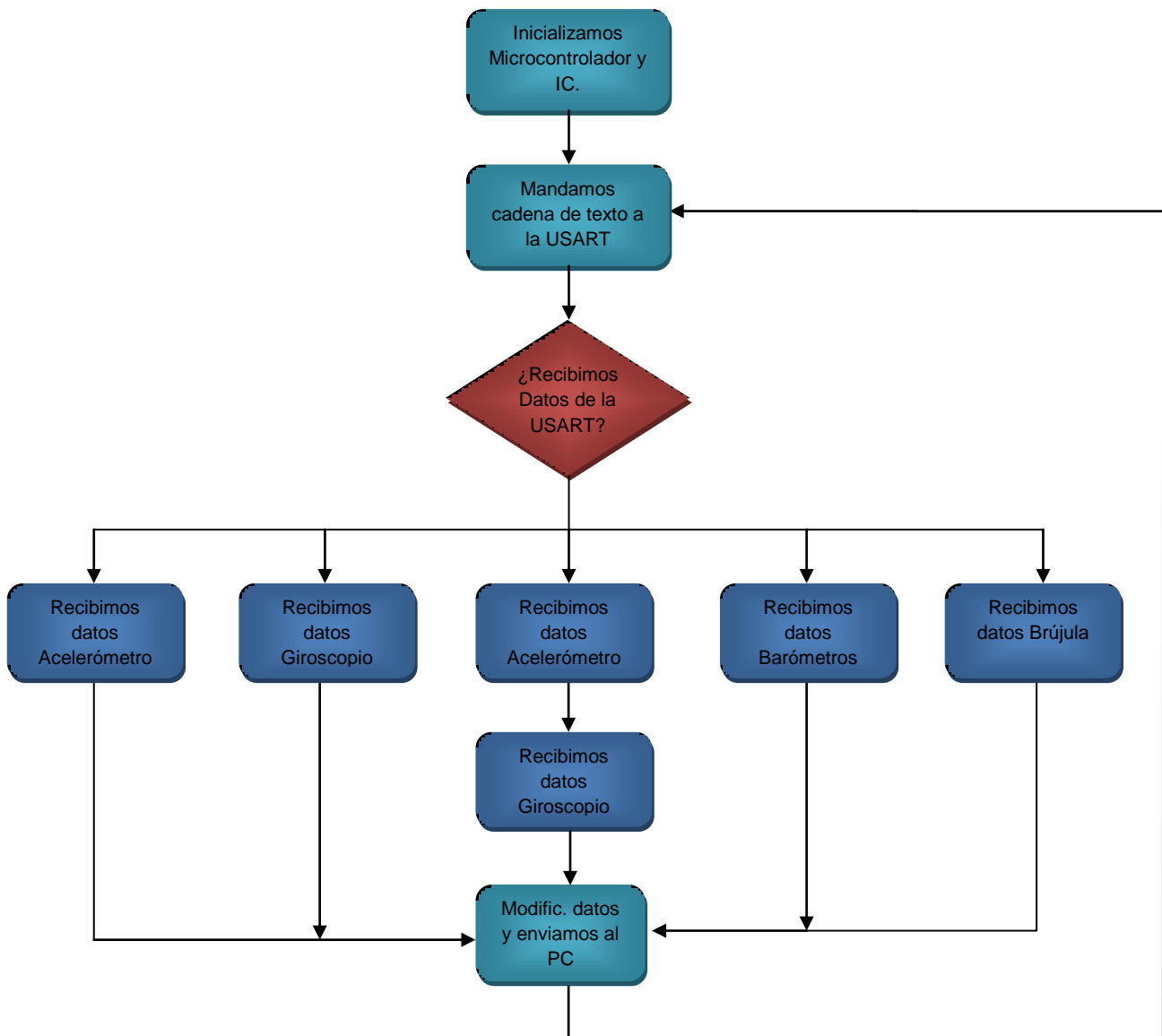


Figura 112 Flujograma del Código C.

Lo primero que hicimos fue configurar nuestro microcontrolador para conseguir que funcionara a la frecuencia deseada. Seguidamente, se programó el estado de los pines, comprobándose que funcionaban correctamente.

Luego se desarrolló por orden nuestro código C. Se fijó la frecuencia de comunicación de nuestro puerto USART, para que trabajara a una velocidad de

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

115.200 Kb/seg, que es una de las frecuencias que más se utilizan para la comunicación vía serie.

Se configuró también la velocidad del puerto SPI, la cual, debería ofrecer la misma velocidad para que se comunicaran todos los dispositivos conectados a este puerto. De esta manera, se dividió la frecuencia del microcontrolador entre cuatro, con lo cual quedaba nuestra velocidad de 500 Kb/seg, velocidad adecuada para que todos los dispositivos se comunicaran correctamente.

Para recibir la información de los dispositivos, se realizó un bucle infinito que siempre enviaba una cadena de texto con las opciones de medida que queríamos recibir. Cuando eligiéramos la opción, mostraríamos en pantalla un número determinado de veces, la medida que fue elegida.

9.0 Pruebas De Funcionamiento

Las pruebas de funcionamiento se han realizado para los distintos dispositivos que componen nuestro proyecto. Se empezó realizando pruebas por separado a cada uno de los dispositivos, y luego se completó probando todos los dispositivos funcionando al mismo tiempo.

TPS60500

La primera vez que conectamos nuestra placa PCB al puerto USB, lo principal fue comprobar la alimentación, pues teníamos que asegurarnos que todos los IC estaban recibiendo la tensión adecuada.

Se midió con un voltímetro la alimentación general de la placa y las diferentes zonas por separado de alimentación analógica. El resultado fue un voltaje de 3.29v, 0.01v menos de lo que habíamos pronosticado, algo que no afecta en absoluto al funcionamiento de nuestros integrados.

ATXMEGA128A3

A continuación, hicimos funcionar nuestro microcontrolador, que es el dispositivo clave de nuestro proyecto, ya que sin que el ATXMEGA128A3 funcione, es imposible que los demás dispositivos nos proporcionen los datos deseados.

Para probar si estaba bien alimentado y si podíamos acceder a sus registros y configurarlos, se conectó al puerto JTAG mediante el dispositivo JTAG MkII y vimos como éramos capaces de observar los estados de todos sus registros y fusibles.

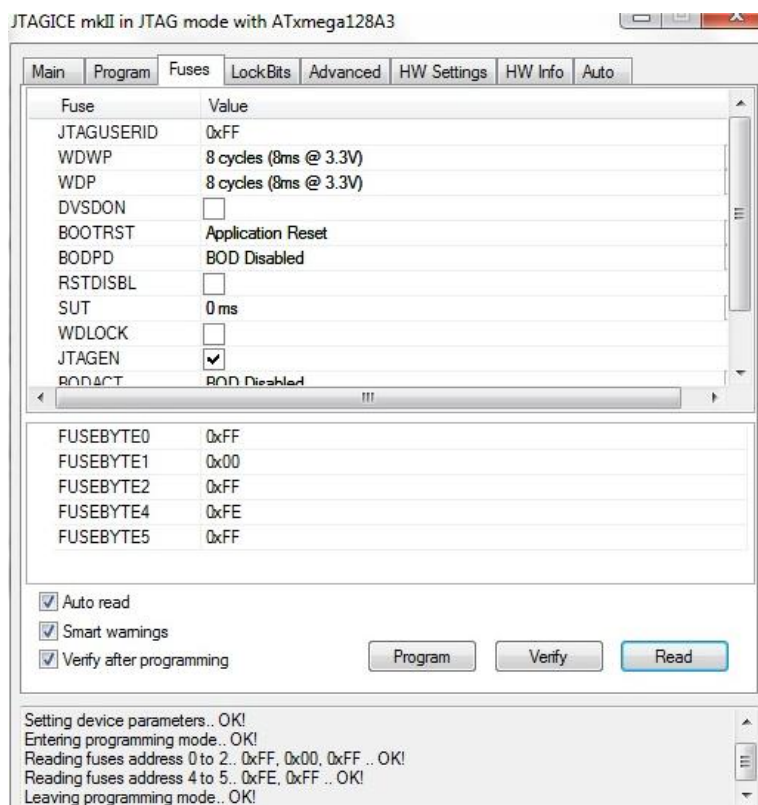


Figura 113 Lectura del estado del ATXMEGA128A3.

Además de poder leer nuestro controlador, fuimos un paso más allá e hicimos un pequeño programa de prueba, el cual activaba un pin a nivel alto y lo desactivaba. De esta manera sencilla pudimos comprobar que nuestro microcontrolador permitía ser programado.

La prueba fue satisfactoria y se pudo comprobar, a través de un osciloscopio, cómo nuestra señal cambiaba de estado. Debido a esto, pudimos seguir programando los demás dispositivos.

CP2102

Lo tercero que hicimos funcionar fue nuestro dispositivo USB, pues mediante él, podríamos observar los datos de los demás dispositivos en nuestro ordenador y así verificar si los datos eran correctos o no.

Mandamos una cadena de texto por nuestro puerto USART para ver si éramos capaces de verla en el PC y así comprobar si funcionaba el bus de transmisión.

Cuando se comprobó que recibíamos la cadena de texto, fuimos un paso más allá y probamos si transmitía un comando ASCII del teclado del PC. Para comprobar si lo recibíamos, nuestro CP2102 produciría un eco.

Y efectivamente, conseguimos enviar desde nuestra placa PCB una cadena de texto, de recibir un carácter del teclado del ordenador y de volverlo a enviar por el bus USART, por lo que nuestro dispositivo de comunicación CP2102 funcionó perfectamente.

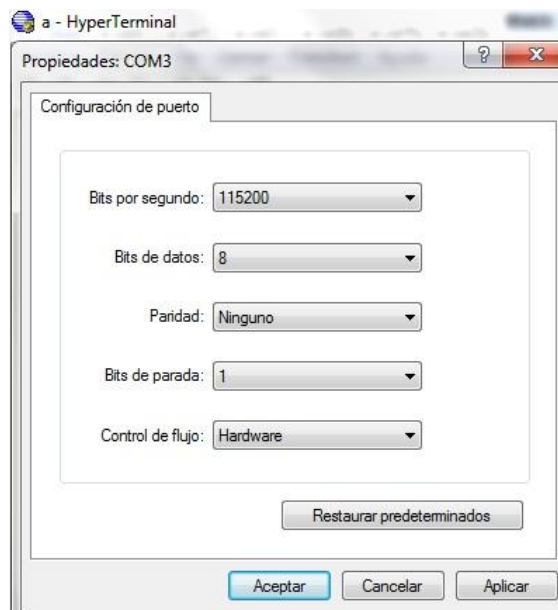
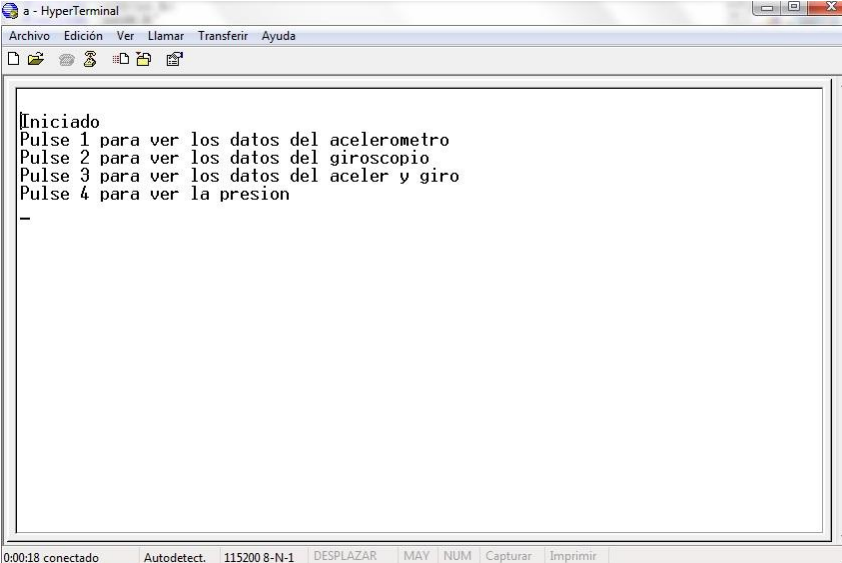


Figura 114 Configuración HyperTerminal.

En la figura anterior, observamos cómo con el Baud Rate indicado a 115.200 kb/seg, transmitimos al programa "HyperTerminal" el cual se comunica con el puerto COM VIRTUAL que crea nuestro dispositivo CP2102.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.



```
HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda
Iniciado
Pulse 1 para ver los datos del acelerometro
Pulse 2 para ver los datos del giroscopio
Pulse 3 para ver los datos del aceler y giro
Pulse 4 para ver la presion
-
0:00:18 conectado Autodetect. 115200 8-N-1 DESPLAZAR MAY NUM Capturar Imprimir
```

Figura 115 Test del puerto USART.

En la figura 115, podemos ver el texto que enviamos por el puerto USART a nuestro PC. Su correcto funcionamiento quedó probado.

Acelerómetro

Para comprobar el funcionamiento del ADXL345, se mandaron los datos que recogíamos de este por el puerto serie una vez modificados y adaptados en código ASCII para poder ser entendidos en nuestro PC.

Para comprobar que los datos que recibíamos eran correctos, se colocó la placa en una posición fija para que todos los ejes nos dieran los mismos datos de manera continuada. Una vez que observamos que los datos se mantenían de manera fija, movimos la placa para ver si los datos cambiaban. Al mover la placa, se observó que efectivamente, los datos variaban acorde al movimiento de la placa.

ADXLX=000C	Y=0107	Z=FFCD	i=E5	ADXLX=FFFB	Y=0005	Z=FEE8	i=E5
ADXLX=000A	Y=010A	Z=FFD5	i=E5	ADXLX=FFFC	Y=0005	Z=FEE5	i=E5
ADXLX=000F	Y=0108	Z=FFD4	i=E5	ADXLX=FFF8	Y=0005	Z=FEE5	i=E5
ADXLX=000D	Y=0109	Z=FFCB	i=E5	ADXLX=FFFA	Y=0006	Z=FEE9	i=E5
ADXLX=000C	Y=0108	Z=FFCD	i=E5	ADXLX=FFFF	Y=0001	Z=FEE5	i=E5
ADXLX=000C	Y=0109	Z=FFCB	i=E5	ADXLX=FFFB	Y=0004	Z=FEEC	i=E5
ADXLX=000C	Y=010B	Z=FFCF	i=E5	ADXLX=FFF8	Y=0004	Z=FEEA	i=E5
ADXLX=000D	Y=0105	Z=FFCB	i=E5	ADXLX=FFFC	Y=0004	Z=FEE4	i=E5
ADXLX=000E	Y=0108	Z=FFCF	i=E5	ADXLX=FFFD	Y=0002	Z=FEE6	i=E5
ADXLX=000B	Y=0108	Z=FFCE	i=E5	ADXLX=FFF8	Y=0006	Z=FEEA	i=E5
ADXLX=000A	Y=0109	Z=FFD6	i=E5	ADXLX=FFFA	Y=0008	Z=FEE6	i=E5
ADXLX=000E	Y=0107	Z=FFCD	i=E5	ADXLX=FFFC	Y=0003	Z=FEEC	i=E5
ADXLX=000B	Y=0108	Z=FFD4	i=E5	ADXLX=FFFB	Y=0003	Z=FEE8	i=E5
ADXLX=000A	Y=010C	Z=FFCF	i=E5	ADXLX=FFFC	Y=0004	Z=FEE3	i=E5
ADXLX=000D	Y=0108	Z=FFD1	i=E5	ADXLX=FFFB	Y=0005	Z=FEEE	i=E5
ADXLX=000C	Y=0109	Z=FFCE	i=E5	ADXLX=FFF9	Y=0003	Z=FEEB	i=E5
ADXLX=000A	Y=010B	Z=FFCF	i=E5	ADXLX=FFFA	Y=0009	Z=FEE8	i=E5
ADXLX=000C	Y=0109	Z=FFC8	i=E5	ADXLX=FFFE	Y=0006	Z=FEEA	i=E5

Figura 116 Datos del Acelerómetro.

En la figura de los Datos del Acelerómetro, se puede apreciar cómo en la columna de la izquierda los datos del eje “X” están en torno a 0 y los datos del eje “Y” están sobre los 100 en hexadecimal. Esto quiere decir que la placa estaba en posición vertical por el eje “Y”, por eso marcaba el máximo en positivo, y el eje “X” estaba en posición normal.

En la misma figura pero en la columna derecha, vemos que el eje “Y” ahora está con valores próximos a 0, y es el eje X, el que nos da valores negativos próximos a 0. Esto es debido al cambio de posición de la placa.

Para verificar aún más los datos que recibíamos del SPI del acelerómetro, se agregó el campo de identidad “i” en cada lectura de este.

Al ver que los datos cambiaban en consonancia a los movimientos de nuestra placa PCB y que el campo de identidad de nuestro ADXL345 era el correcto, quedó demostrado que nuestro acelerómetro funcionaba según lo esperado.

Giroscopio

La comprobación del funcionamiento de nuestro L3G4200 fue un poco más complicada, pues no es tan fácil apreciar el cambio de datos al mover nuestra placa.

Al ser un movimiento angular y no un movimiento lineal lo que mide el giroscopio, al mover la placa, los datos no son acordes a su movimiento, sino que hay que deducir su funcionamiento.

Para comprobar que los datos que recibíamos de nuestro giro eran correctos, se mantuvo la placa inmóvil para que todos los datos que recibíamos fueran 0, es decir, que no hubiera movimiento angular.

```
L3GX=FFEF Y=FFEF Z=0007 i=D3
L3GX=FFE3 Y=FFE7 Z=FFF0 i=D3
L3GX=FFD8 Y=FFEF Z=FFF3 i=D3
L3GX=FFE6 Y=FFE9 Z=FFDF i=D3
L3GX=FFDB Y=FFD7 Z=FFFB i=D3
L3GX=FFDC Y=FFF2 Z=FFF5 i=D3
L3GX=FFCE Y=FFE9 Z=FFEE i=D3
L3GX=FFDA Y=FFD5 Z=FFE6 i=D3
L3GX=FFE3 Y=FFF7 Z=FFE1 i=D3
L3GX=FFE0 Y=FFD8 Z=FFFF i=D3
L3GX=FFD1 Y=0003 Z=FFF3 i=D3
L3GX=FFCC Y=FFEB Z=FFDD i=D3
L3GX=FFDF Y=FFF4 Z=FFF6 i=D3
L3GX=FFDD Y=FFE6 Z=FFDC i=D3
L3GX=FFBE Y=FFFA Z=FFD1 i=D3
L3GX=FFED Y=000A Z=FFEA i=D3
L3GX=FFCA Y=FFF4 Z=FFF8 i=D3
L3GX=FFCF Y=FFF6 Z=FFF1 i=D3
```

Figura 117 Datos del Giroscopio.

Se pudo observar como los ejes tenían valores próximos a 0. Y también se leyó el campo identificativo del giroscopio, para poder afirmar que funcionaba correctamente.

```
ADXLX=000F Y=FFFF Z=00EC i=E5 L3GX=FFF7 Y=FF23 Z=0088 i=D3
ADXLX=000E Y=FFFE Z=00ED i=E5 L3GX=0036 Y=FF3C Z=008F i=D3
ADXLX=000F Y=0000 Z=00E9 i=E5 L3GX=FFA8 Y=FFE7 Z=FF81 i=D3
ADXLX=000D Y=FFF7 Z=00EB i=E5 L3GX=FFDA Y=008E Z=FFB4 i=D3
ADXLX=000E Y=FFFC Z=00EF i=E5 L3GX=FFBF Y=00EC Z=FF6F i=D3
ADXLX=000D Y=0003 Z=00E5 i=E5 L3GX=FFE1 Y=FF39 Z=0175 i=D3
ADXLX=000F Y=FFFD Z=00F0 i=E5 L3GX=FF75 Y=FEC6 Z=FEAE i=D3
ADXLX=000D Y=FFF9 Z=00E9 i=E5 L3GX=0049 Y=FFF8 Z=0117 i=D3
ADXLX=0011 Y=0000 Z=00E7 i=E5 L3GX=FF38 Y=00D3 Z=FE30 i=D3
ADXLX=000F Y=FFF9 Z=00E8 i=E5 L3GX=005E Y=FFA9 Z=0151 i=D3
ADXLX=0010 Y=FFFE Z=00EC i=E5 L3GX=FF58 Y=0066 Z=FF00 i=D3
ADXLX=0010 Y=FFF8 Z=00EE i=E5 L3GX=FFB2 Y=008F Z=FFC2 i=D3
ADXLX=000C Y=0005 Z=00E9 i=E5 L3GX=0093 Y=FD99 Z=00F1 i=D3
ADXLX=000F Y=0001 Z=00E5 i=E5 L3GX=004B Y=FEFE Z=00CA i=D3
ADXLX=000F Y=FFFF Z=00E7 i=E5 L3GX=FFFF Y=0000 Z=FFFC i=D3
ADXLX=0011 Y=FFFB Z=00E3 i=E5 L3GX=FF5F Y=01C5 Z=FEFA i=D3
ADXLX=000D Y=0001 Z=00F0 i=E5 L3GX=002E Y=FF7A Z=00A4 i=D3
ADXLX=000F Y=FFFB Z=00ED i=E5 L3GX=FF56 Y=0144 Z=FF30 i=D3
```

Figura 118 Datos Acelerómetro y Giroscopio.

En la figura anterior, se puede valorar el funcionamiento de acelerómetro y del giroscopio trabajando a la vez. En esta figura se intento mantener la placa con un ligero movimiento (nuestro pulso) para comprobar que el acelerómetro casi no apreciaba este movimiento, sin embargo el giroscopio sí nos daba valores apreciativos.

Barómetro

Para probar el funcionamiento de los barómetros, procedimos de distintas maneras. Lo primero fue leer varias veces los coeficientes, para ver si cada vez que los leíamos, los valores de estos eran los mismos.

Name	Value	Type	Location
a0_1	1919.75	float	0x22E5 [SI]
b1_1	-2.2606201	float	0x22B3 [SI]
b2_1	-0.92541504	float	0x2312 [SI]
c12_1	0.00077033043	float	0x22EA [SI]
pressure	386	int	0x22C5 [SI]
temperature	481	int	0x22CC [SI]

Figura 119 Lectura del Barómetro 1.

En la figura 119 podemos ver cómo los coeficientes a0, b1, b2 y c12, se mantienen constantes y no cambian en cada lectura; sin embargo, los datos de temperatura y presión, sí que cambiaban muchas veces, aunque de manera poco apreciable.

Cuando se comprobó esto, se cambió la programación, para que no leyera constantemente los coeficientes, puesto que no era necesario.

Luego, para comprobar si la presión que nos daba era la real, hicimos mediciones a nivel del mar, en donde la presión atmosférica es superior, y también realizamos mediciones a 320 metros de altura aproximadamente (universidad), y los valores estaban de acuerdo a lo que esperábamos. Es decir, obtuvimos una presión de 101 KPa a nivel del mar y menor (97KPa) en la universidad.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

P1=096.81 P2=096.98
P1=096.93 P2=097.03
P1=097.21 P2=096.98
P1=097.13 P2=097.07
P1=097.05 P2=097.17
P1=097.09 P2=096.98
P1=097.01 P2=096.98
P1=097.25 P2=096.98
P1=097.17 P2=096.98
P1=097.17 P2=096.98
P1=097.05 P2=096.98
P1=097.09 P2=096.98
P1=097.01 P2=096.93
P1=097.17 P2=097.03
P1=097.05 P2=097.25
P1=097.01 P2=096.98
P1=097.05 P2=096.98
P1=097.13 P2=097.22

Figura 120 Lectura de los barómetros.

Como comprobamos que un barómetro ya funcionaba, se conectó el otro también. En ambos obtuvimos medidas prácticamente iguales, confirmando aun más la teoría de que funcionaban correctamente.

En la figura 120 se observa lo dicho anteriormente.

10.0 Conclusiones y Líneas Futuras

De la realización de este proyecto se pueden destacar las siguientes conclusiones:

Primeramente cabe destacar que se ha logrado el objetivo de realizar una placa PCB de unas dimensiones considerablemente pequeñas. Se han aprovechando al máximo los espacios, evitando que quedaran huecos vacíos que incrementarían el tamaño de esta.

También hay que mencionar que hicimos funcionar uno de los microcontroladores más modernos que han salido al mercado y por lo tanto, este es uno de los primeros proyectos que utiliza este tipo de controlador XMEGA. Como consecuencia tuvimos que adaptarnos a la nueva forma de acceder a sus registros.

Además hay que señalar la gran cantidad y variedad de sensores que se han hecho funcionar correctamente. Muchos de ellos hubo que adaptarlos, debido a que no eran los que en un principio se diseñaron, sino que se implementó su funcionamiento después de tener nuestra PCB ya fabricada.

El acelerómetro, giroscopio y los dos barómetros, son los sensores fundamentales en el equipo de telemetría que diseñamos, y se consiguió que todos trabajaran individualmente y conjuntamente.

Tras haber tenido muchas dificultades por el cambio de algunos componentes debido a su ausencia en el mercado, se lograron los máximos resultados, para el tiempo que se nos ha planteado y puesto como límite para concluir este proyecto.

En el aspecto de líneas futuras se han tenido en cuenta posibles ampliaciones, por lo que dejamos un conector al puerto SPI, para que se pudieran implementar sensores adicionales que funcionen con este puerto.

Se dejó ese puerto SPI y no otro, porque este es el puerto con mayor proyección a seguir funcionando en un futuro, debido a la gran cantidad de circuitos integrados que funcionan con él.

De igual forma, también se ha dejado el circuito de nuestro ultrasonido preparado, para que en el supuesto de que se consiguiera el transmisor y receptor ultrasónico, se pudiera hacer funcionar debidamente, hecho que nosotros no pudimos finalizar debido a los motivos expuestos en párrafos anteriores.

11.0 Bibliografía

- [1] IEEE. (2006). 802.15.4-2006 - IEEE Standard for Local and Metropolitan Area Networks - Specific Requirements - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs). Obtenido de: <http://standards.ieee.org/findstds/standard/802.15.4-2006.html>.
- [2] IETF. (s.f.). RFC 4944 - Transmission of IPv6 Packets over IEEE 802.15.4 Networks. Obtenido de: <http://tools.ietf.org/pdf/rfc4944>.
- [3] IEEE 802.15 WPAN™ Task Group 4 (TG4). (s.f.). IEEE 802.15.4. Obtenido de: <http://ieee802.org/15/pub/TG4.html>.
- [4] Wireless Sensor Network, Docencia UCA (Universidad de Cádiz), (s.f.), Obtenido de: <http://www.mfbarcell.es/conferencias/wsn>
- [5] Tecnologías inalámbricas de corto alcance: zigbee y bluetooth, Barriga Vázquez, Walter Edwin Zúñiga Calle, Fabián Paúl, 29-nov-2006, Obtenido de: <http://dspace.ups.edu.ec/handle/123456789/206>
- [6] Giroscopios en el sistema GypsyGyro-18, publicación interna nº 4 (Universidad de Alicante), (s.f.), Obtenido de: <http://www.aurova.ua.es:8080/proyectos/dpi2005/docs/publicaciones/pub04-Giroscopios/Giroscopios%20en%20el%20Sistema%20GypsyGyro.pdf>
- [7] Diseño e Implementación de un Acelerómetro, Proyecto fin de carrera de Hrich Bouziane, (s.f.), Obtenido de: <http://upcommons.upc.edu/pfc/bitstream/2099.1/7998/1/memoria%20PFC.pdf>

- [8] Sensores de Posición, Facultad de Ingenierías de Uruguay, (s.f.), Obtenido de:
http://www.fing.edu.uy/iimpi/academica/grado/instindustrial/teorico/080310-Sensores-parte_V.posici%C3%B3n.pdf
- [9] Ultrasonidos, Infomed (Red de Salud de Cuba), (s.f.), Obtenido de:
<http://www.sld.cu/galerias/pdf/sitios/rehabilitacion-fis/ultrasonidos.pdf>
- [10] Tipos de Encapsulados, (s.f.), Obtenidos de:
<http://ayudaelectronica.com/tipos-de-encapsulados/>
- [11] Tipos de Barómetros,(s.f.), Obtenidos de:
<http://www.educar.org/inventos/barometro.asp>
- [12] Analog Devices, Acelerómetro,(s.f.), Obtenido de:
<http://www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl345/products/product.html>
- [13] Atmel,(s.f.), Radio, Obtenido de:
<http://www.atmel.com/devices/at86rf212.aspx>
- [14] Silicon Labs,(s.f.), USB, Obtenido de:
<http://www.silabs.com/Support%20Documents/TechnicalDocs/cp2102.pdf>
- [15] HoneyWell,(s.f.), Compass, Obtenido de:
http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5843.pdf
- [16] HoneyWell,(s.f.), Compass, Obtenido de:
<http://www.dfrobot.com/image/data/SEN0073/HMC5883.pdf>

- [17] StMicroelectronics,(s.f.), Giroscopio, Obtenido de:
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00265057.pdf
- [18] Amsys,(s.f.), Barómetro, Obtenido de:
<http://www.amsys.info/products/ms5540.htm>
- [19] Freescale Semiconductor,(s.f.), Barómetro, Obtenido de:
http://www.freescale.com/files/sensors/doc/data_sheet/MPL115A1.pdf
- [20] Texas Instruments,(s.f.), Regulador de Tensión, Obtenido de:
<http://www.ti.com/product/tps60502>
- [21] Atmel,(s.f.), Microcontrolador, Obtenido de:
<http://www.atmel.com/devices/atxmega128a3.aspx>

12.0 PRESUPUESTO

Frank Pérez Paz declara/certifica que el proyecto “Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos”, realizado en calidad de Proyecto Fin de Carrera, y durante un periodo de 15 meses, tiene un coste total de 69.639,00 €, correspondiente a la suma de las cantidades consignadas en los apartados considerados a continuación.

Tabla 12 Coste Total.

Concepto	Importe
Recursos Humanos	45.770,40 €
Recursos Hardware	1.800,00 €
Recursos Software	4.478,00 €
Fabricación Prototipos	1.920,00 €
Coste de la Redacción	3.453,98 €
Subtotal	57.422,38 €
Gastos Generales (5%)	2.871,12 €
Subtotal	60.293,50 €
Beneficio Industrial (10%)	6.029,35 €
Subtotal	66.322,85 €
IGIC (5%)	3.316,15 €
Presupuesto Total	69.639,00 €

Las Palmas de Gran Canaria

Julio 2012

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

La cuantía del trabajo elaborado se ha fijado según las indicaciones del Colegio Oficial de Ingenieros de Técnicos Telecomunicación (COITT). De este modo, el total del presupuesto se ha desglosado en las siguientes secciones, en las que se han separado los distintos costes asociados al desarrollo del Proyecto Fin de Carrera, según su naturaleza.

- Coste de los Recursos Humanos
- Coste de los Recursos Hardware
- Coste de los Recursos Software
- Coste de la Fabricación de Prototipos
- Coste de la Redacción del Proyecto
- Gastos Generales
- Beneficio Industrial
- Coste Total

12.1 Coste de los Recursos Humanos

El coste de recursos humanos está asociado al tiempo empleado por un ingeniero en la realización del proyecto, sin contar el tiempo de aprendizaje de las herramientas utilizadas. De acuerdo con las últimas tablas de honorarios orientativos publicadas en 2008 por el Colegio Oficial de Ingenieros Técnicos de Telecomunicación, el cálculo del coste de un trabajo en función del tiempo empleado se calcula a razón de 63,57 € / hora, al que se le aplicará un factor de reducción de 0,4:

Para la realización del presente Proyecto Fin de Carrera, sin contar el tiempo de aprendizaje, se estima que se ha trabajado durante, aproximadamente, 15 meses, es decir, 60 semanas hábiles, trabajando 5 días semanales y 6 horas efectivas diarias, por lo que el número de horas totales trabajadas resulta:

$$\text{Horas totales} = 6 \text{ horas/día} \times 5 \text{ días/semana} \times 60 \text{ semanas} = 1800 \text{ horas (P.1)}$$

El valor final al que ascienden estos honorarios es de:

$$\text{Honorarios (CosteRRHH)} = 0,4 \times 63,57 \text{ €/h} \times 1800 \text{ h} = 45.770,40 \text{ € (P.2)}$$

En la Tabla P-1 se recogen los costes asociados a los recursos humanos.

Tabla P-1 Costes de los Recursos Humanos.

Concepto	Tiempo Empleado	Coste Mensual Aproximado	Importe
Ingeniero Técnico	10 meses	4.577,04 €	45.770,40 €
Coste Total			45.770,40 €

Por tanto, el trabajo tarifado por tiempo empleado asciende a la cantidad de 45.770,40 €.

12.2 Coste de los Recursos Hardware

En la realización de este Proyecto de Fin de Carrera se han utilizado diferentes equipos informáticos como ordenadores portátiles o impresoras. También, se ha utilizado diverso equipamiento de laboratorio, como una estación de soldadura con aire caliente, diversos equipos de montaje, un microscopio digital de gran aumento y un polímetro.

En cuanto al material de laboratorio, no tendrá presencia en el presupuesto pues se asume que el montaje lo realizaría una empresa externa. Este coste vendrá reflejado en el apartado correspondiente. Los demás costes se reflejarán de manera íntegra en el presupuesto, de manera que estas herramientas son amortizadas durante la realización de este único proyecto.

Tabla P-2 Costes de los Recursos Hardware.

Concepto	Importe
Ordenador	1.500,00 €
Impresora	300,00 €
Coste Total	1.800,00 €

12.3 Costes de los Recursos Software

El coste de los recursos software se obtiene a partir del valor de las licencias y el mantenimiento de cada uno de los programas utilizados. En la Tabla P-3 se resumen estos costes.

Tabla P-3 Costes de los Recursos Software.

Conceptos	Importe
Altium Designer	3.600,00 €
Microsoft Window 7 Profesional	318,00 €
MicroSim Release 8	200,00 €
Microsoft Office Word	180,00 €
Microsoft Office Excel	180,00 €
Coste Total	4.478,00 €

12.4 Costes de Fabricación de Prototipos

El coste de fabricación de prototipos ha de ser contemplado en el presupuesto final, ya que éste forma parte del desarrollo del proyecto. El coste de fabricación contempla los diferentes procesos necesarios para obtener unos prototipos totalmente funcionales. En la Tabla P-4 se resumen estos costes para la fabricación correspondiente a 10 prototipos con el fabricante PCBCart.

Tabla P-4 Coste de la Fabricación del Prototipo.

Concepto	Importe
Fabricación de la PCB	120,00 €
Componentes para la Fabricación	300,00 €
Montaje de los Prototipos	1.500,00 €
Coste Total	1.920,00 €

12.5 Costes de la Redacción del Proyecto

Los costes de redacción del proyecto contemplan los gastos generados para la redacción de la memoria del Proyecto Fin de Carrera.

Siguiendo la recomendación del COITT, referente al valor monetario de la redacción de proyectos y trabajos en general, se aplica la siguiente fórmula para determinar el coste asociado a la redacción de la memoria del proyecto.

$$R=0,07 \times P_{Tel} \times Cr + 0.03 \times P_{Civil} \times Cr \quad (P.3)$$

Donde P_{Tel} es el presupuesto de ejecución del proyecto, que se obtiene sumando las cantidades obtenidas en los tres apartados anteriores y P_{Civil} es el presupuesto de la obra civil del proyecto, en nuestro caso es cero.

Por otro lado, Cr es el coeficiente reductor para honorarios en función del coste del presupuesto.

Por tanto, el valor de P_{Tel} es:

$$P_{Tel} = 45.770,40 \text{ €} + 1.800,00 \text{ €} + 4.478,00 \text{ €} + 1.920,00 \text{ €} = 53.968,40 \text{ €} \quad (P.4)$$

Para este valor de P_{Tel} , el coeficiente Cr vale 0.8. Por tanto, el valor de R es:

$$R = 0.07 \times 53.968,40 \text{ €} \times 0.8 = 3.453,98 \text{ €} \quad (P.5)$$

12.6 Gastos Generales y Beneficio Industrial

Además de los gastos presentados en las secciones anteriores, hay que incluir un 5% sobre el total de los gastos anteriores para incluir los costes derivados del uso de material necesario para la elaboración del proyecto.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

Todo proyecto ha de contemplar un margen de beneficio industrial que, para el caso del presente Proyecto de Fin de Carrera, se ha estimado en un 10%.

12.7 Coste Total

En la Tabla P-5 se recopila el conjunto de los gastos contemplados obteniéndose el importe final.

Tabla P-5 Coste Total

Concepto	Importe
Recursos Humanos	45.770,40 €
Recursos Hardware	1.800,00 €
Recursos Software	4.478,00 €
Fabricación Prototipos	1.920,00 €
Coste de la Redacción	3.453,98 €
Subtotal	57.422,38 €
Gastos Generales (5%)	2.871,12 €
Subtotal	60.293,50 €
Beneficio Industrial (10%)	6.029,35 €
Subtotal	66.322,85 €
IGIC (5%)	3.316,15 €
Presupuesto Total	69.639,00 €

El presupuesto total del Proyecto de Fin de Carrera asciende a un total de sesenta y nueva mil seiscientos treinta y nueve euros (69.639,00 €).

Anexos

Lista de Materiales

Cantidad	Descripción	Fabricante	Manufacturer Part Number	Precio\$/u
1	MCU AVR 128K FLASH 1.8V 64- QFN	Atmel	ATXMEGA128A3-MH	10.43
1	CR STAL 16.000000 MH 8PF SMD	NDK	NX3225SA- 16.000000MH	1.06
1	CR STAL 32.768 KH 9PF SMD	NDK	NX3215SA-32.768K- STD-MUA-9	1.34
2	CAP CER 16pF 50V C0G 0402	TDK Corporation	C1005C0G1H160J	0.024
2	CAP CER 18pF 50V 5% C0G 0402	Murata Electronics North America	GRM1555C1H180J 01D	0.019
2	CONN HEADER 8POS DL UNSHRD SMD	FCI	20021121-00008C4LF	1.02
6	CAP CER 0.1uF 16V X7R 0402	Murata Electronics North America	GRM155R71C104KA88D	0.015
1	RES 4.7K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ472X	0.05
1	IC ACCEL SPI/I2C 3AX 3G 14LGA	Analog Devices Inc	ADXL345BCC -RL	6.16
1	CAP TANTALUM 1uF 16V 20% SMD 0603	Rohm Semiconductor	TCM1C105M8R	0.62
1	CAP CER 0.1uF 16V	Murata Electronics	GRM155R71C104KA88D	0.015

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

	X7R 0402	North America		
1	FERRITE BEAD 100 OHM .20A 0402	Wurth Electronics Inc	742792711	0.30
1	CAP TANT 16V 10uF SMD 0805	Nichicon	F951C106MPAAQ2	1.16
2	MODULE BAROMETER MINITURE SENSOR	Measurement Specialties Inc.	MS5540-CM	24.49
2	CAP TANT 47uF 10V 20% SMD 0805	AVX Corporation	TLJR476M010R3200	3
2	RES 4.7K OHM 1/10W 5% 0402	Panasonic - ECG	ERJ-2GEJ472X	0.05
1	IC USB-TO-UART BRIDGE 28VQFN	Silicon Laboratories Inc	CP2102-GM	3.17
1	CAP CER 1uF 16V X5R 0402	TDK Corporation	C1005X5R1C105K	0.088
2	CAP CER 0.1uF 16V X7R 0402	Murata Electronics North America	GRM155R71C104KA88D	0.015
1	CAP CER 4.7uF 6.3V X5R 0402	Murata Electronics North America	GRM155R60J475ME87D	0.484
1	TVS AVAL DIODE ARRA 3 CH SOT143	Littelfuse Inc	SP0503BAHTG	0.94
3	RES 4.7K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ472X	0.05
1	CONN RECEPT USB 5POS RT ANG SMD	Molex Inc	56579-0576	2.70
1	DIODE SCHOTTK 200MA 30V 0402	Comchip Technology	CDBQR0230L	0.48

1	IC TXRX IGBE/802.15.4/ISM 32QFN	Atmel	AT86RF212- U	5.25
1	FILTER BALUN CER CHIP 863- 928MH	Johanson Technology Inc	0896FB15A0100E	0.73
4	CAP CERM 1UF 10% 16V X5R 0603	AVX Corporation	0603 D105KAT2A	0.077
1	CR STAL 16.000000 MH 8PF SMD	NDK	NX3225SA- 16.000000MH	1.06
2	CAP CER 16PF 50V C0G 0402	TDK Corporation	C1005C0G1H160J	0.024
2	CAP CERM 68PF 5% 50V NP0 0603	AVX Corporation	06035A680JAT2A	0.088
1	CAP CERM 2.2PF 50V NP0 0603	AVX Corporation	06035A2R2CAT2A	0.209
1	RES 680 OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ681X	0.05
1	CAP CERAMIC 100pF 25V NP0 0402	Kemet	C0402C101J3GACTU	0.055
1	RCPT IPEX MHF	Taoglas Limited	RECE-20279-001E-01	0.76
1	IC CHARGE PUMP STPDN 250MA 10MSOP	Texas Instruments	TPS60500DGSR	2.80
2	CAP CER 1.0uF 10V X7R 0805	Taiyo uden	LMK212B7105KG-T	0.36
1	CAP CERM 2.2uF 10V X7R 0805	Taiyo uden	LMK212B7225MG-T	0.31
1	CAP CER 10uF 6.3V 5V 0805	TDK Corporation	C2012 5V0J106	0.132
1	RES 750K OHM 1/10W 5% 0402	Panasonic - ECG	ERJ-2GEJ754X	0.05

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

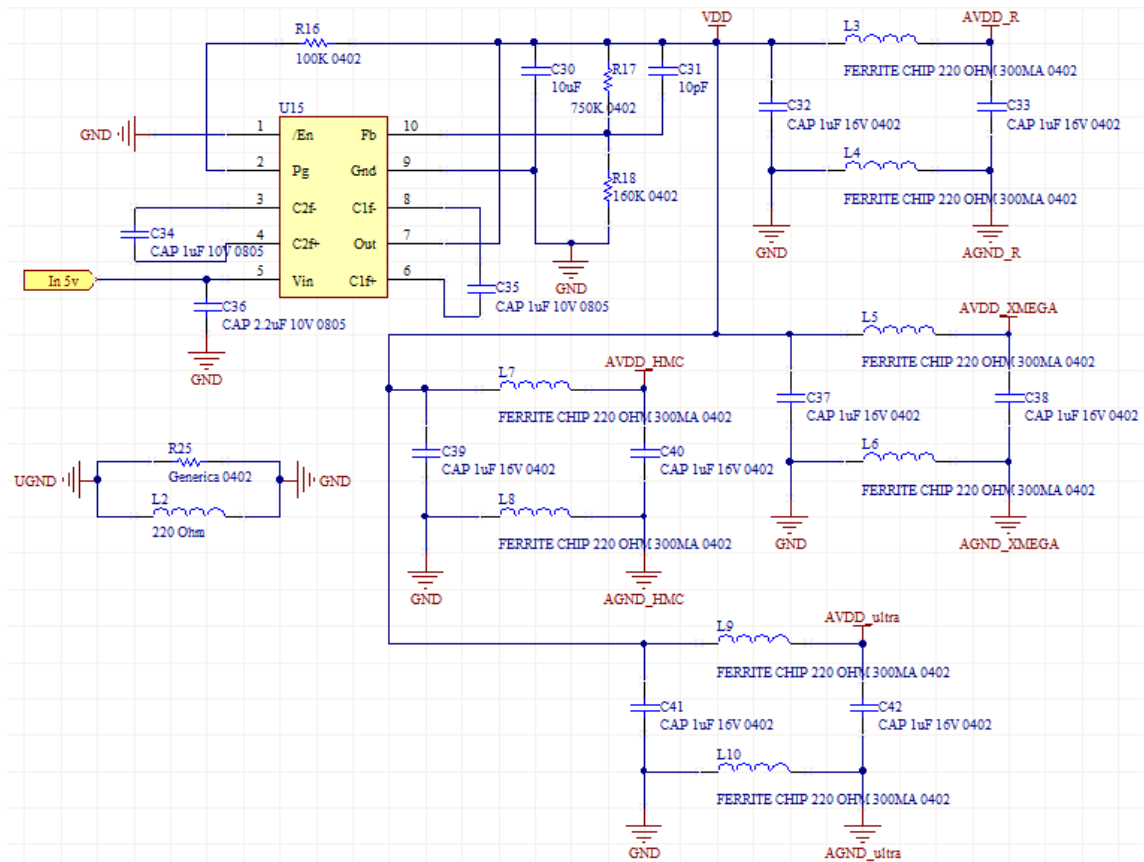
	SMD			
1	RES 160K OHM 1/10W 1% 0402 SMD	Panasonic - ECG	ERJ-2RKF1603X	0.06
1	RES 100K OHM .125W 5% 0402 SMD	Vishay/Dale	CRCW0402100KJNEDHP	0.154
1	CAP CERAMIC 10pF 50V NP0 0402	ageo	CC0402JRNP09BN100	0.012
8	FERRITE CHIP 220 OHM 300MA 0402	Murata Electronics North America	BLM15AG221SN1D	0.052
8	CAP CER 1uF 16V X5R 0402	TDK Corporation	C1005X5R1C105K	0.088
1	G ROSCOPE MEMS 3-AXIS 16LGA	STMicroelectronics	L3G4200D	12.98
1	CAP CERAMI 0.01uF 25V X7R 0402	ageo	CC0402KRX7R8BB103	0.011
1	CAP CER 0.47uF 10V X5R 0402	TDK Corporation	C1005X5R1A474K	0.077
1	RES 10.0K OHM 1/10W 1% 0402 SMD	Panasonic - ECG	ERJ-2RKF1002X	0.06
1	CAP CER 0.1uF 16V X7R 0402	Murata Electronics North America	GRM155R71C104KA88D	0.015
1	CAP CER 10uF 10V X5R 20% 0603	Taiyo uden	LMK107BJ106MALTD	0.64
1	SENSOR MAGNETIC 3 AXIS 20-LCC	Honeywell Microelectronics Precision Sensors	HMC5843-TR	16.52
1	CAP CER 0.22uF 16V 5V 0402	TDK Corporation	C1005 5V1C224	0.03
1	CAP CER 4.7uF	Murata Electronics	GRM155R60J475ME87D	0.484

	6.3V X5R 0402	North America		
2	RES 50K OHM .15W 0.1% 0603 SMD	Vishay/Thin Film	PNM0603E5002BST5	2.52
2	IC BUFFER TRI-ST QD LW V 14TSSOP	Fairchild Semiconductor	74LVX125MTCX	0.47
2	RES 4.7K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ472X	0.05
1	255-400ST16-ROX			
1	255-400SR16-ROX			
1	IC COMP 1.6V SNGL P-P SOT23-5	Microchip Technology	MCP6541RT-E/OT	0.34
1	IC OPAMP 2.5V DUAL R-R 8SOIC	Microchip Technology	MCP6022-I/SN	1.14
1	IC DGTL POT 10K 1CH SOT23-6	Microchip Technology	MCP4013T-103E/CH	0.49
1	IC DGTL POT 50K 1CH RHEO SOT23- 6	Microchip Technology	MCP4012T-503E/CH	0.49
1	CONN HEADER .050 4POS DL SMD AU	Sullins Connector Solutions	GRPB022VWQS-RC	0.27
1	IC RS232 3V5.5V 15KVESD 16-SOIC	Texas Instruments	MAX3232ECDR	1.68
4	CAP CER 0.1uF 16V X7R 0402	Murata Electronics North America	GRM155R71C104KA88D	0.015
1	CAP CERAMIC 100pF 25V NP0 0402	Kemet	C0402C101J3GACTU	0.055
1	RES 78.7K OHM 1/10W 1% 0402 SMD	Panasonic - ECG	ERJ-2RKF7872X	0.05
2	RES 1.0K OHM 1/10W 5% 0402	Panasonic - ECG	ERJ-2GEJ102X	0.04

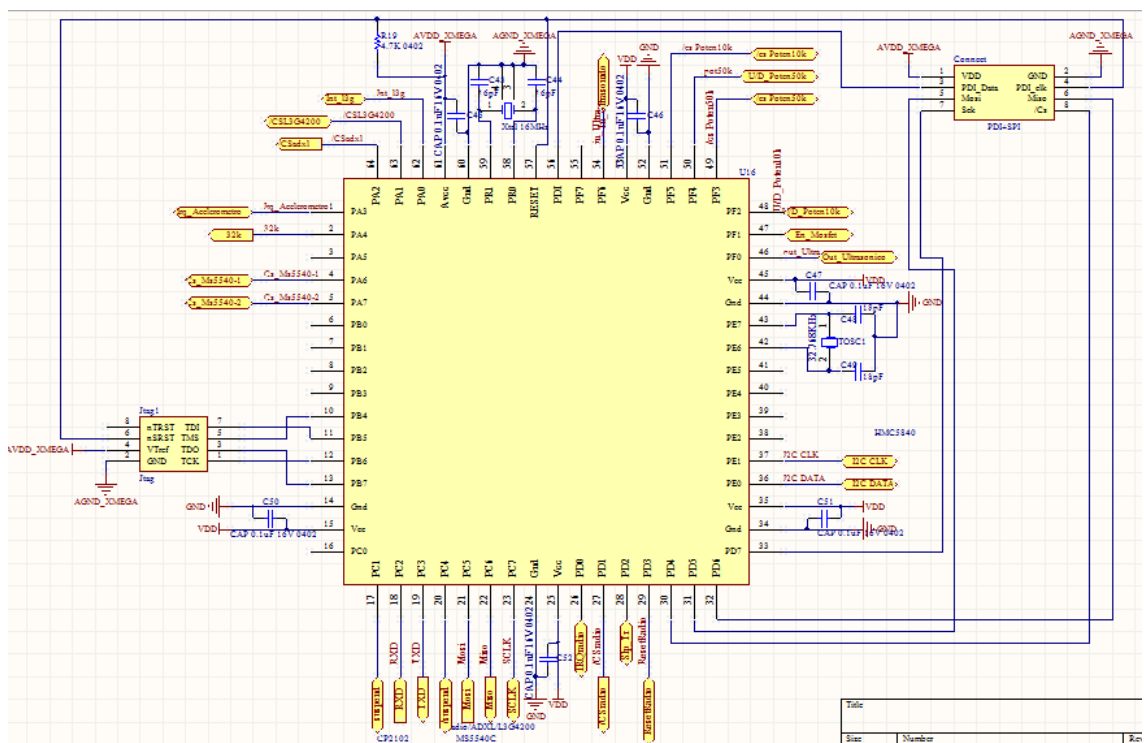
Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

	SMD			
1	RES 240K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ244X	0.04
1	CAP CER 1uF 16V X5R 0402	TDK Corporation	C1005X5R1C105K	0.088
2	RES 3.3K OHM 1/10W 5% 0402 SMD	Panasonic - ECG	ERJ-2GEJ332X	0.04
1	IC LOAD SWITCH HI SIDE 2A 4-TMLF	Micrel Inc	MIC94080 FT TR	0.62
2	CAP CER 10uF 10V X5R 20% 0603	Taiyo uden	LMK107BJ106MALTD	0.64

Esquemáticos

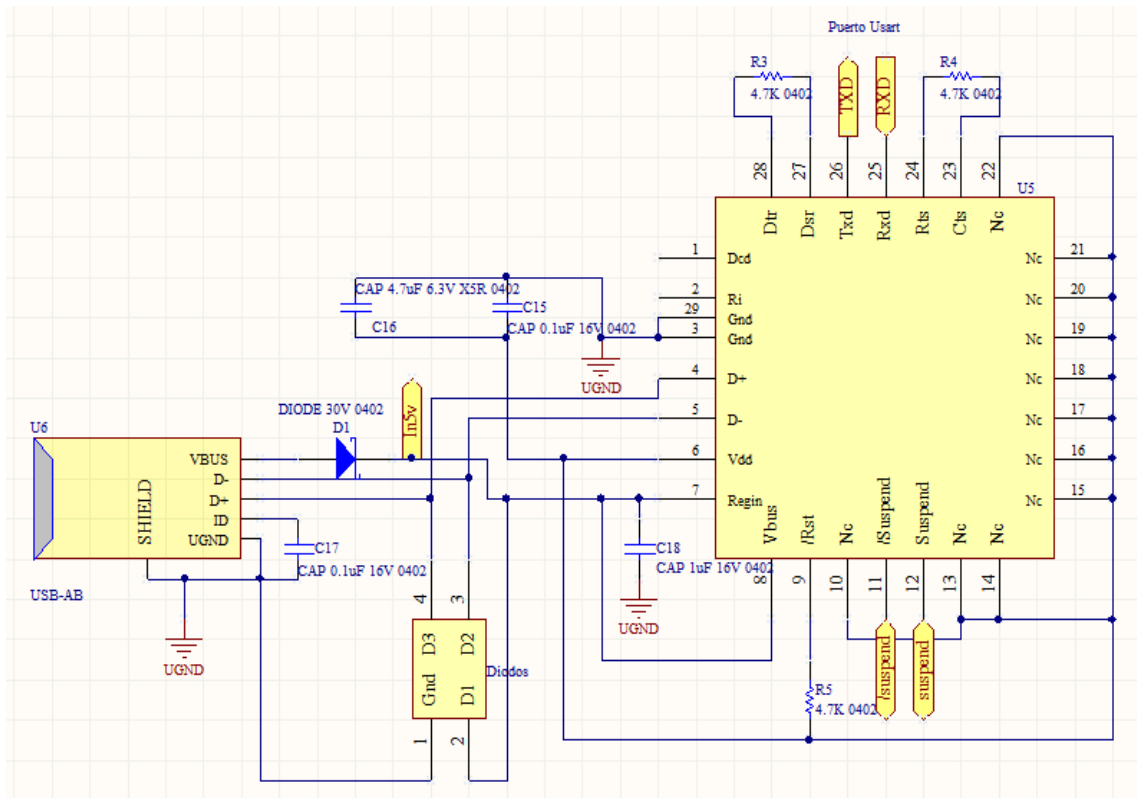


Esquemático TPS60500.

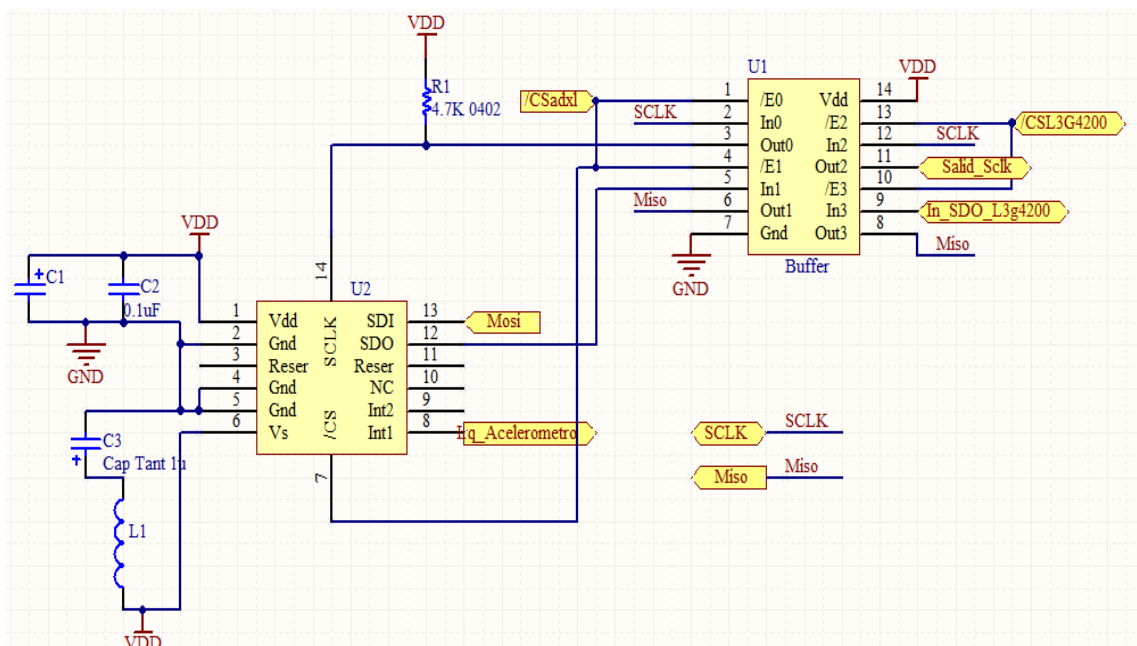


Esquemático Microcontrolador ATXMEGA128A3.

Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.

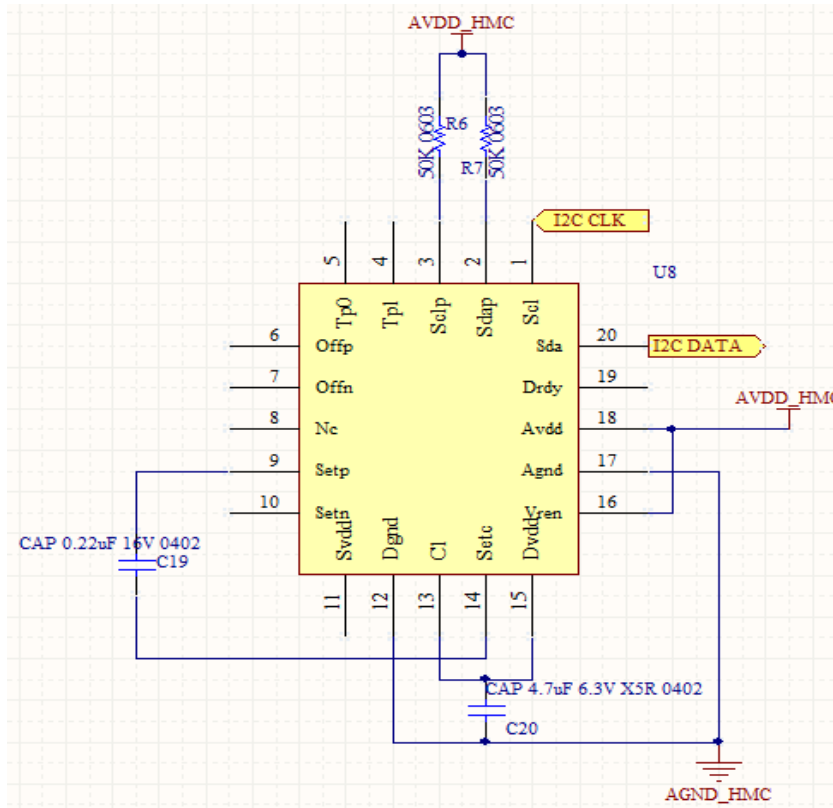


Esquemático USB CP2102.

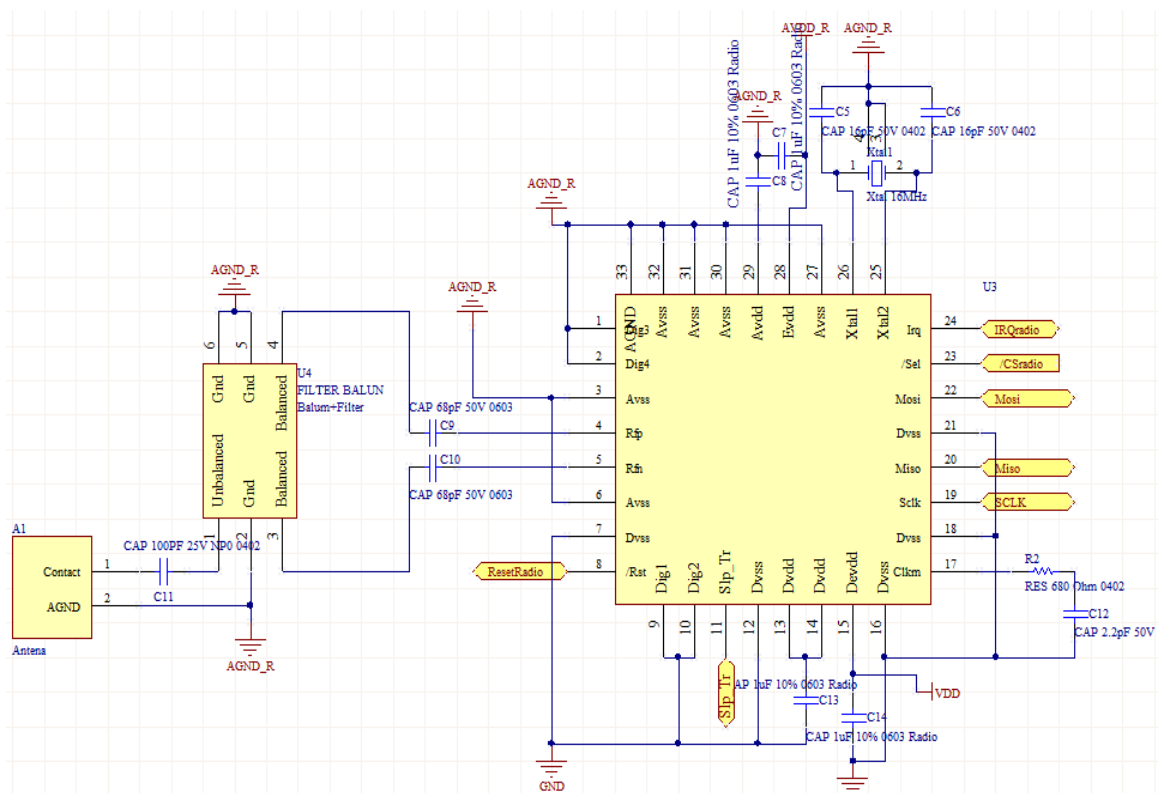


Esquemático Acelerómetro ADXL345.

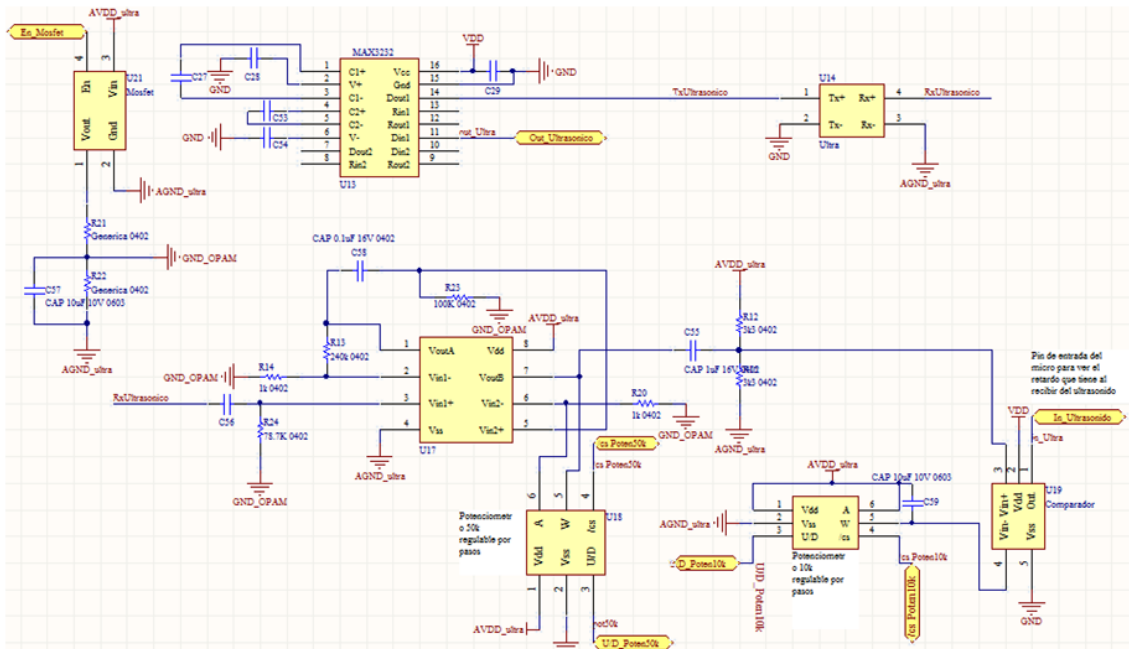
Diseño e Implementación de una Unidad de Telemetría de bajo costo para Sistemas de Navegación Autónomos.



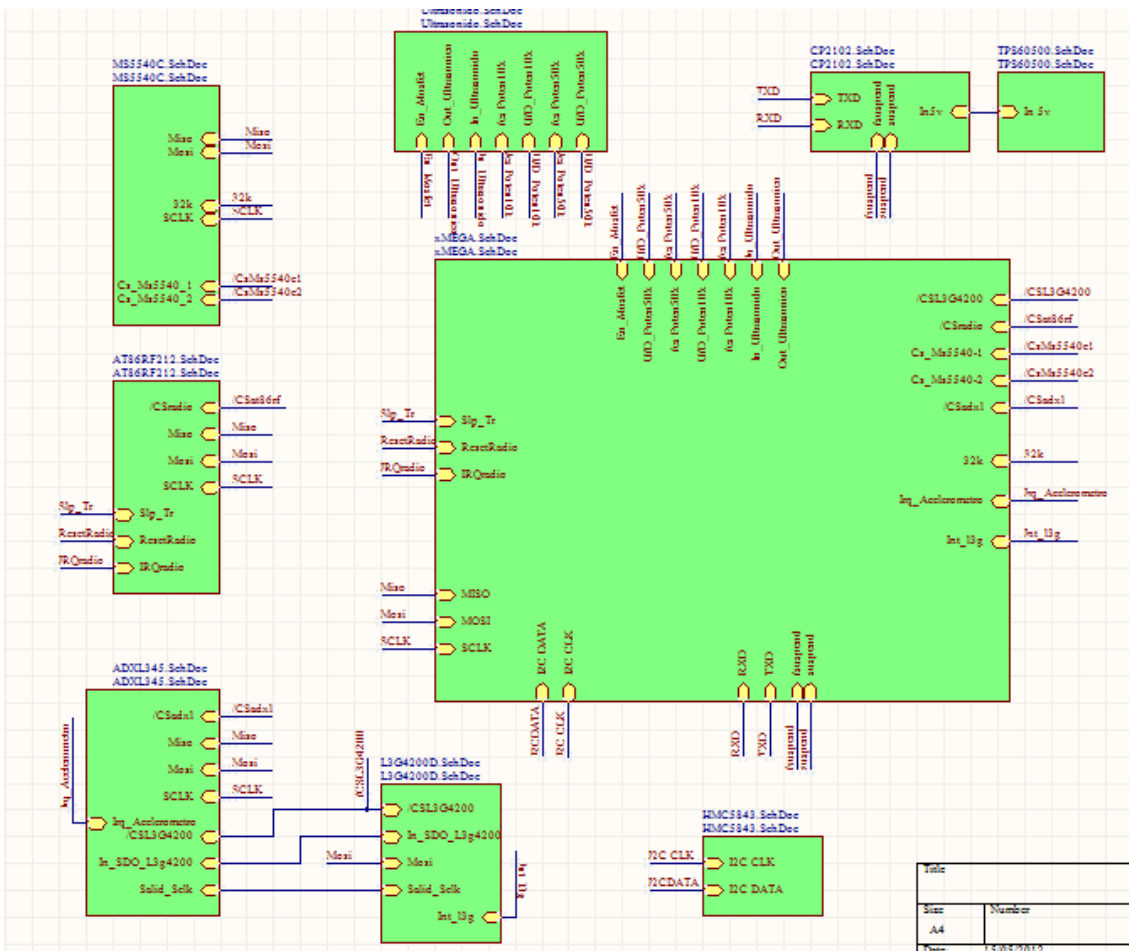
Esquemático Brújula HMC5843.



Esquemático Radio AT86RF212.

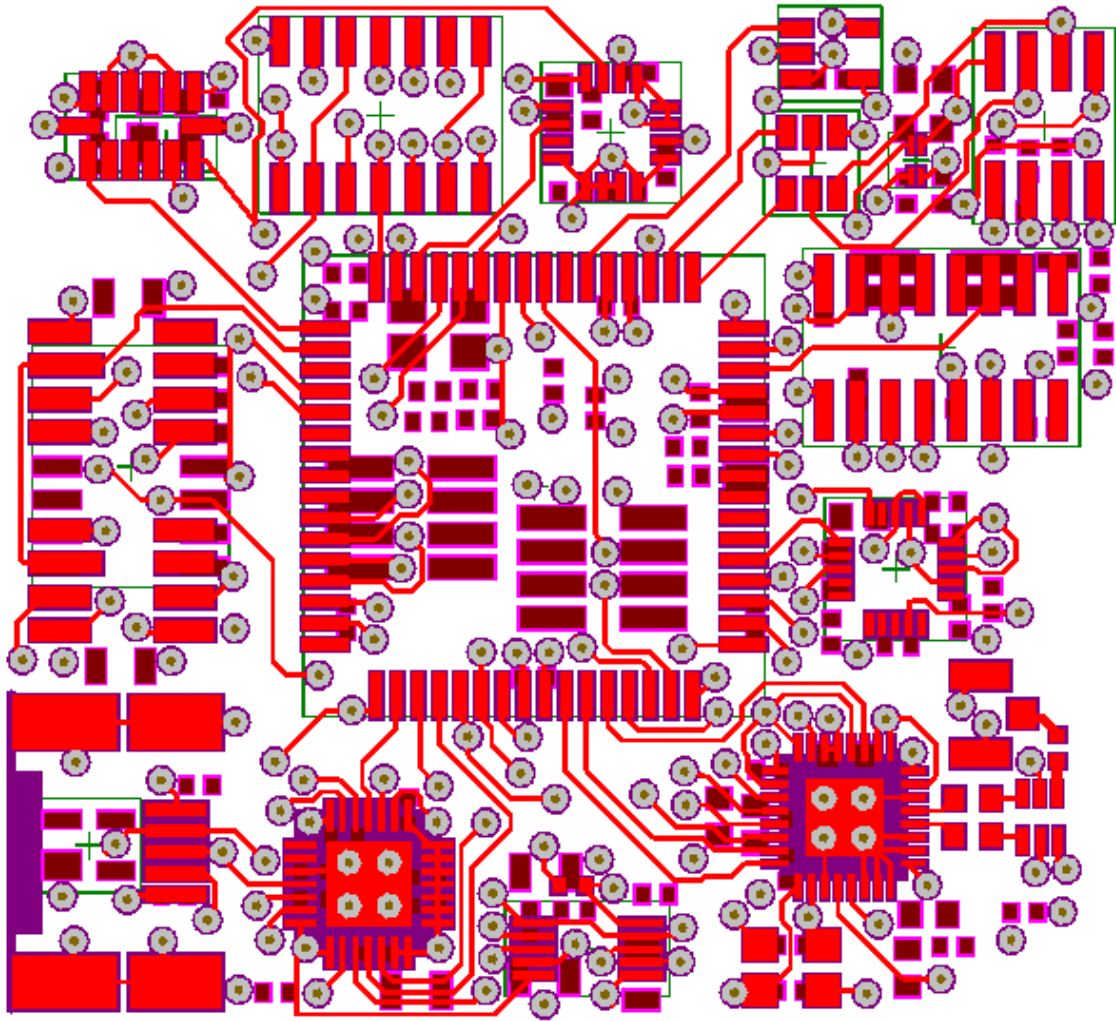


Esquemático Ultrasonido.

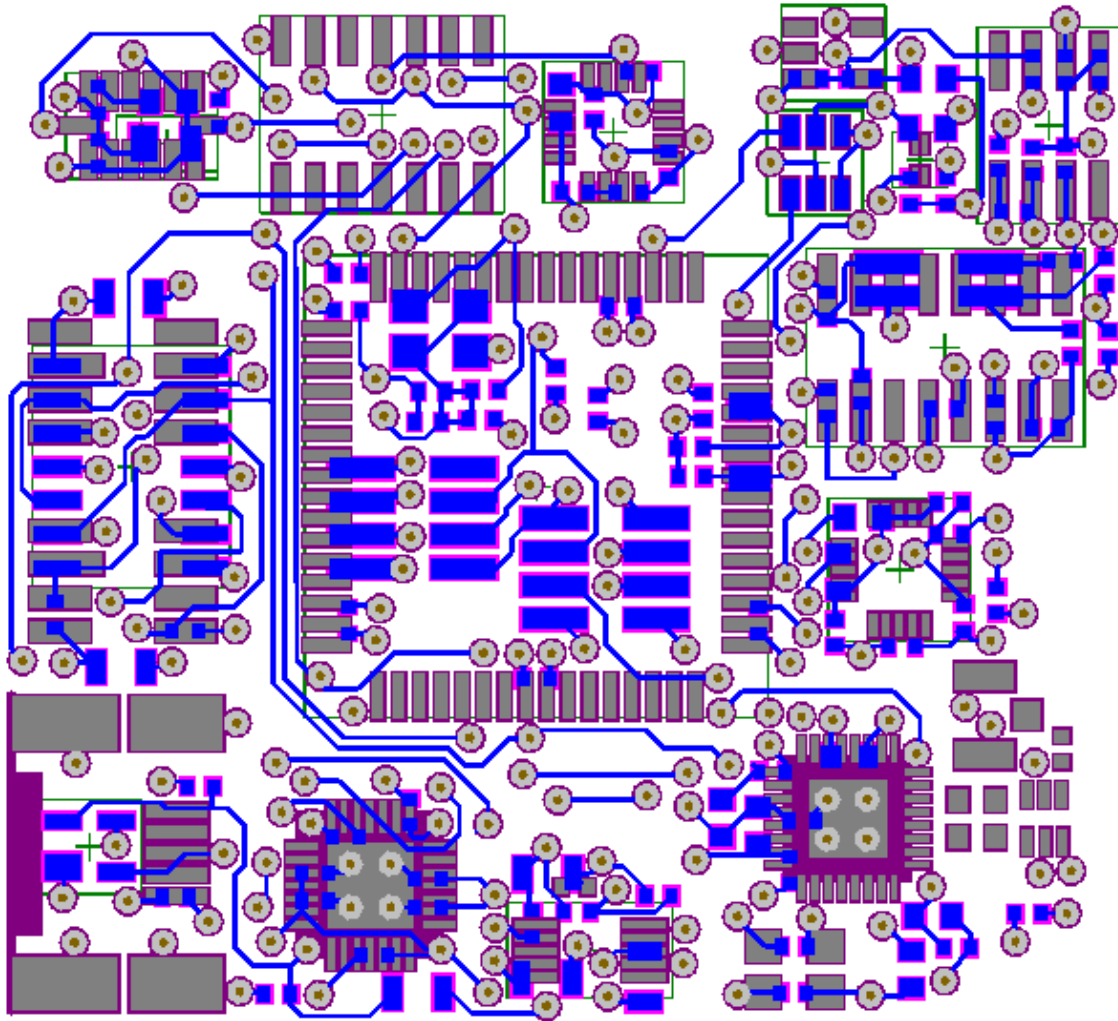


Esquemático General

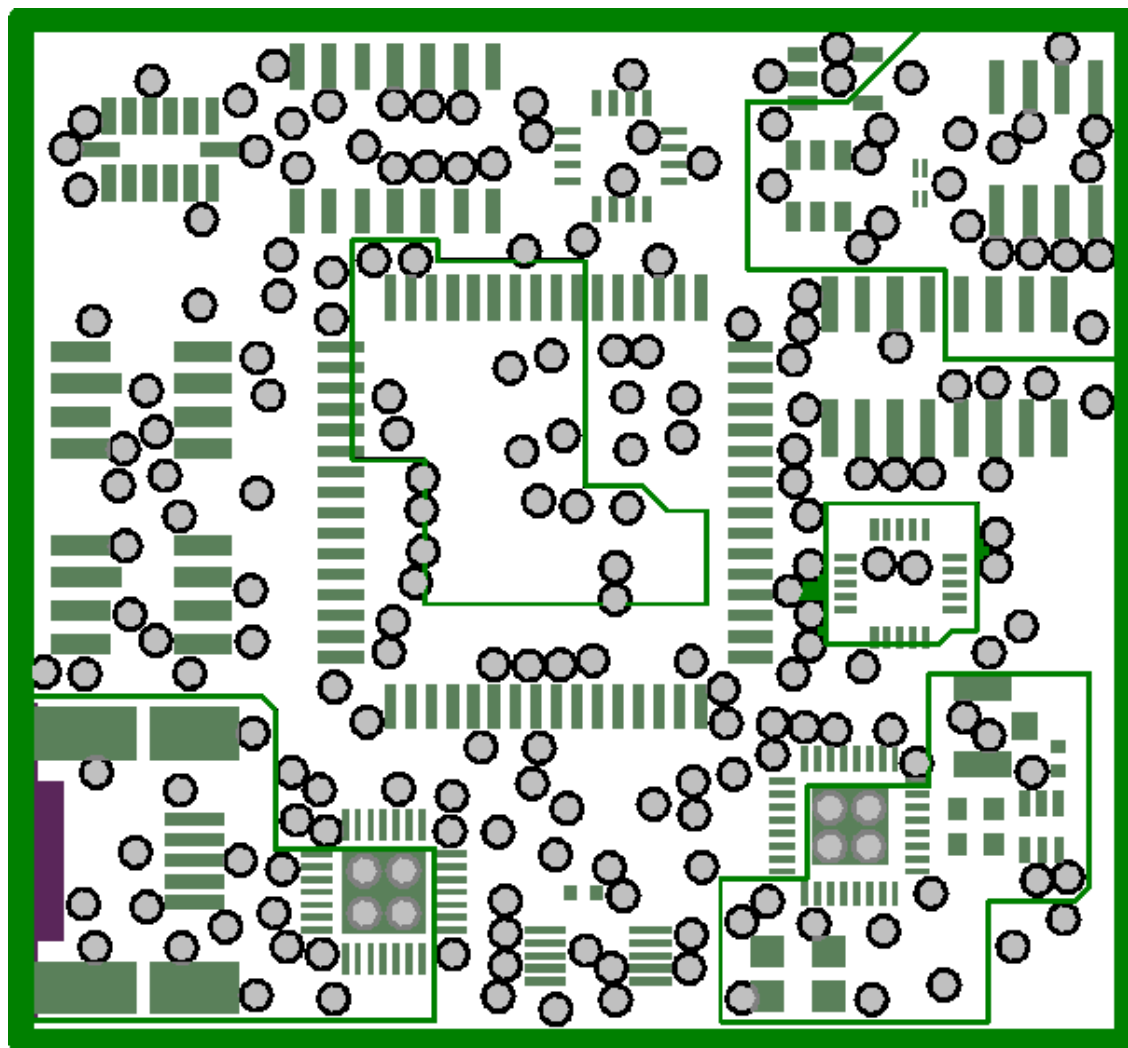
Máscaras PCB



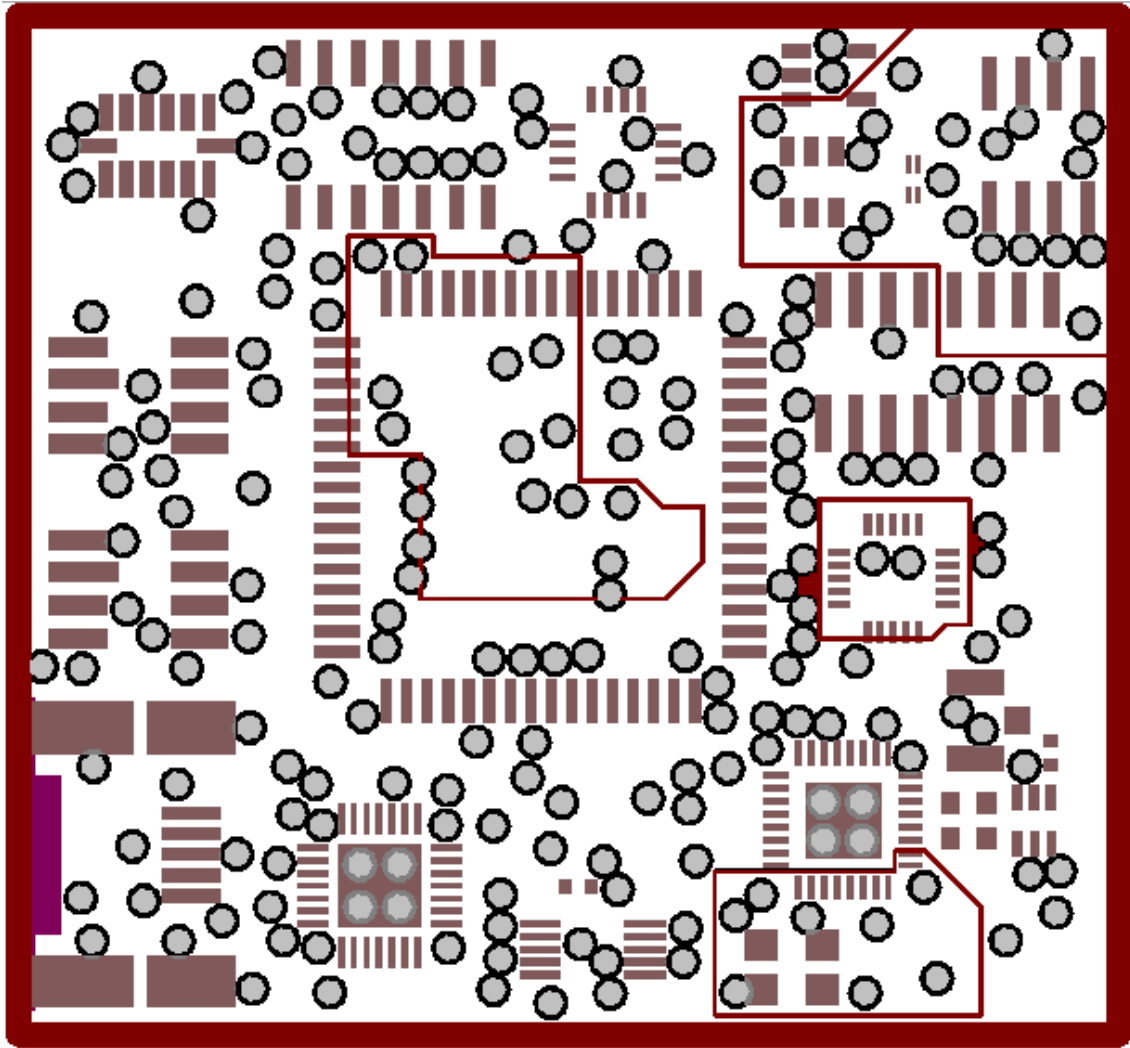
Capa Superior



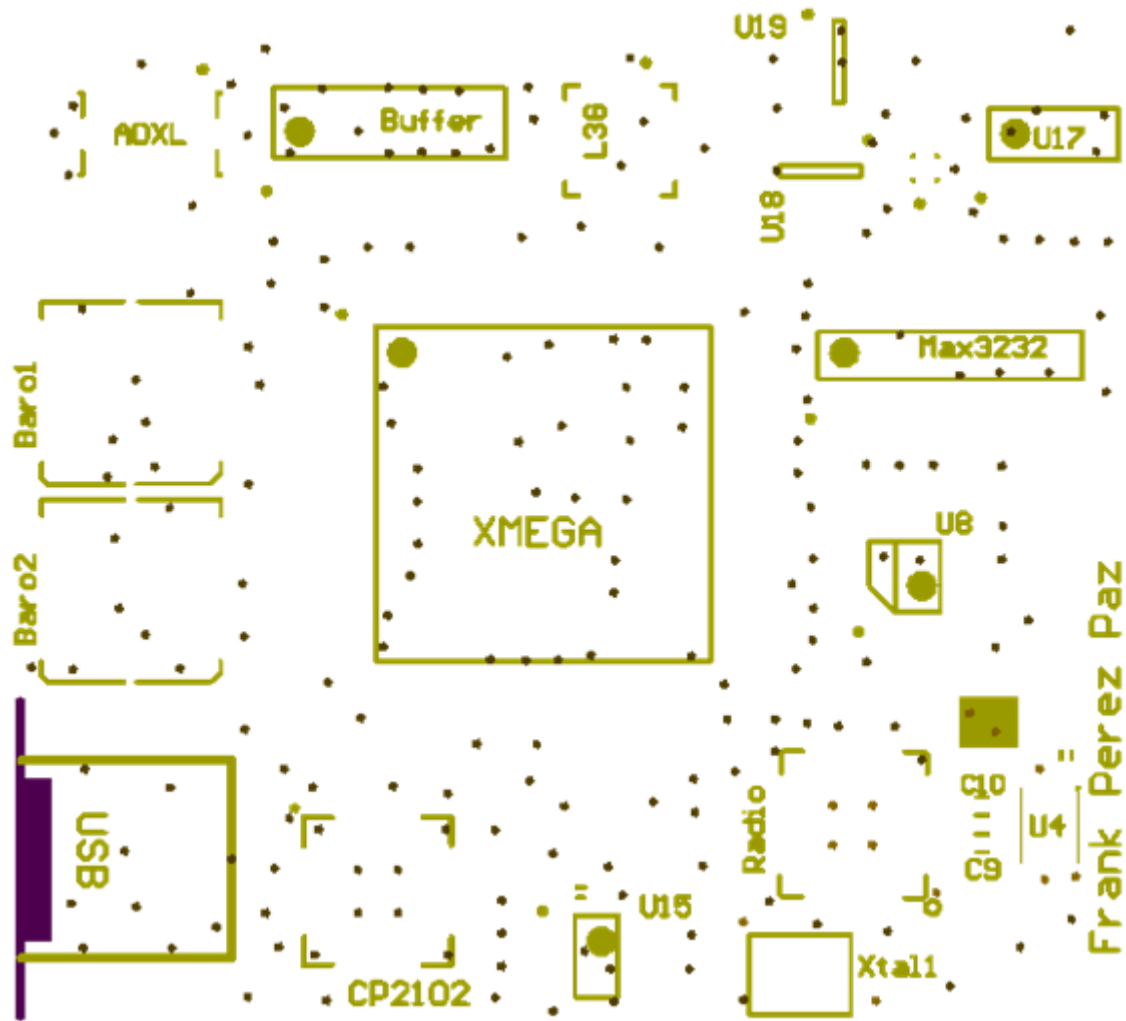
Capa Inferior



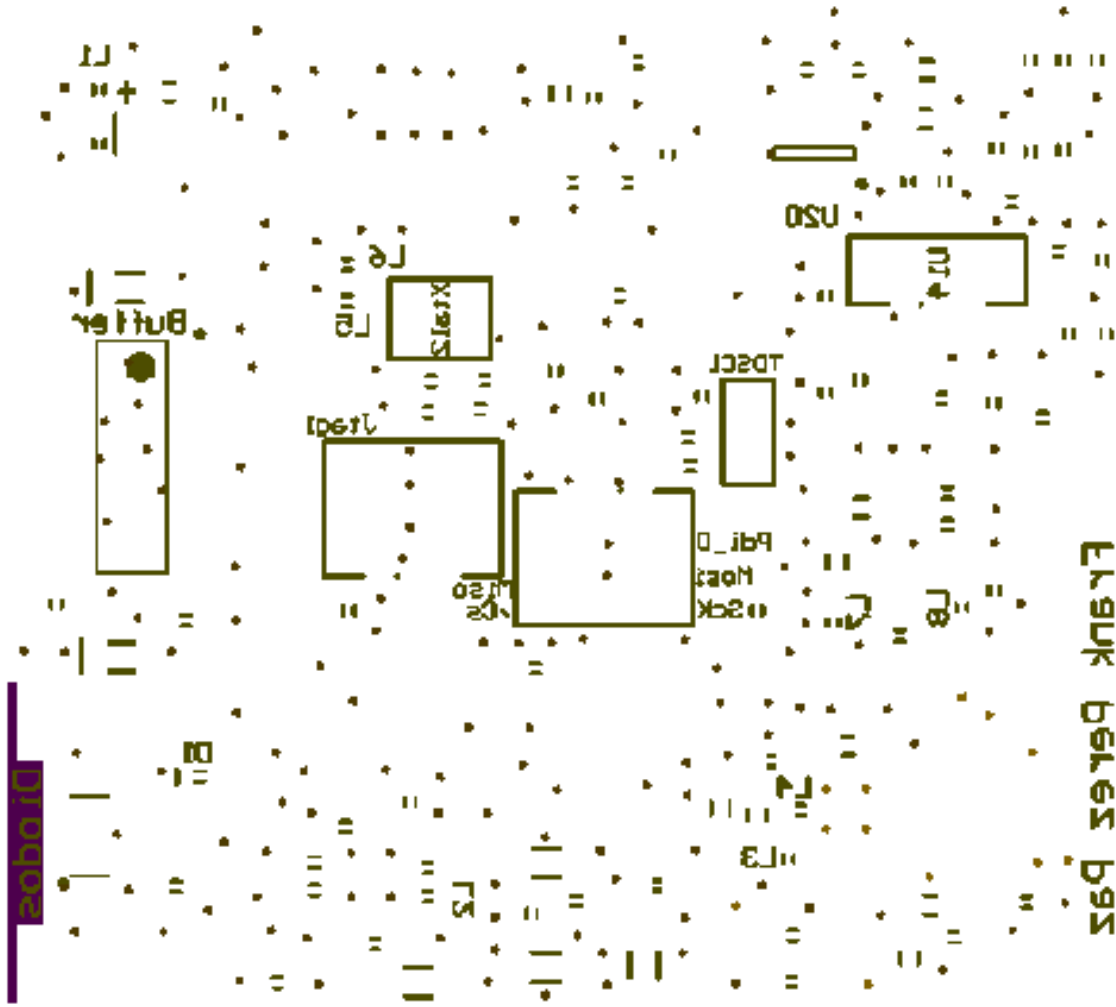
Plano de Masa



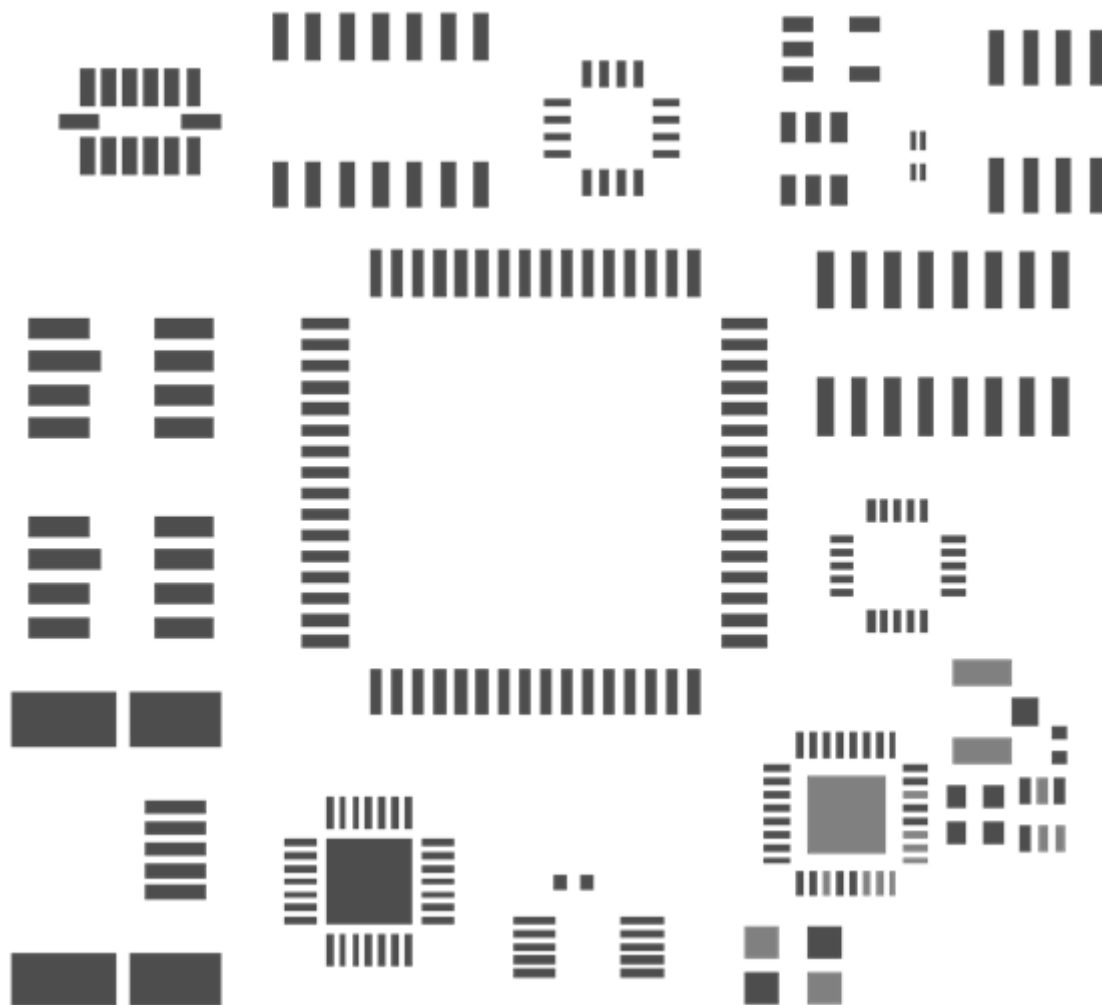
Plano Alimentación



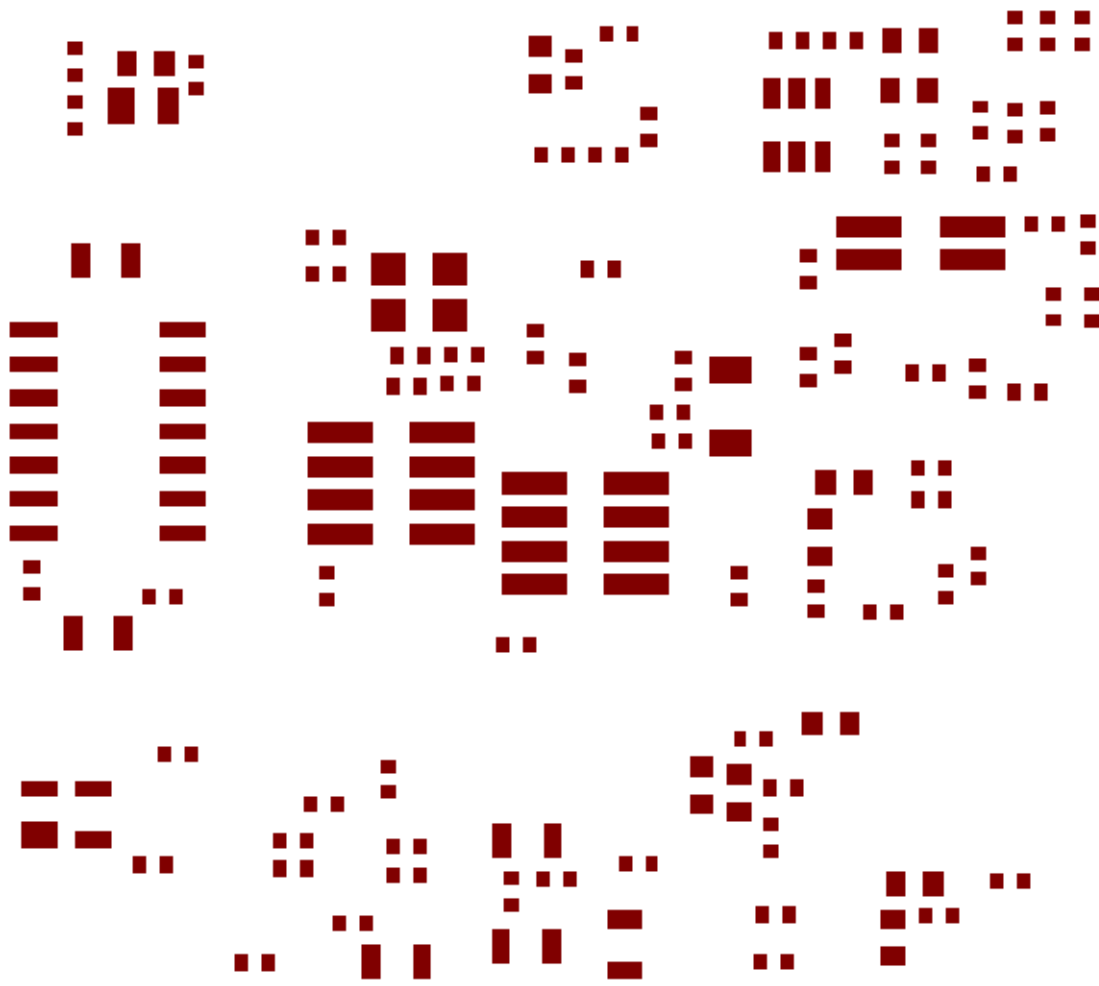
Serigrafía Superior



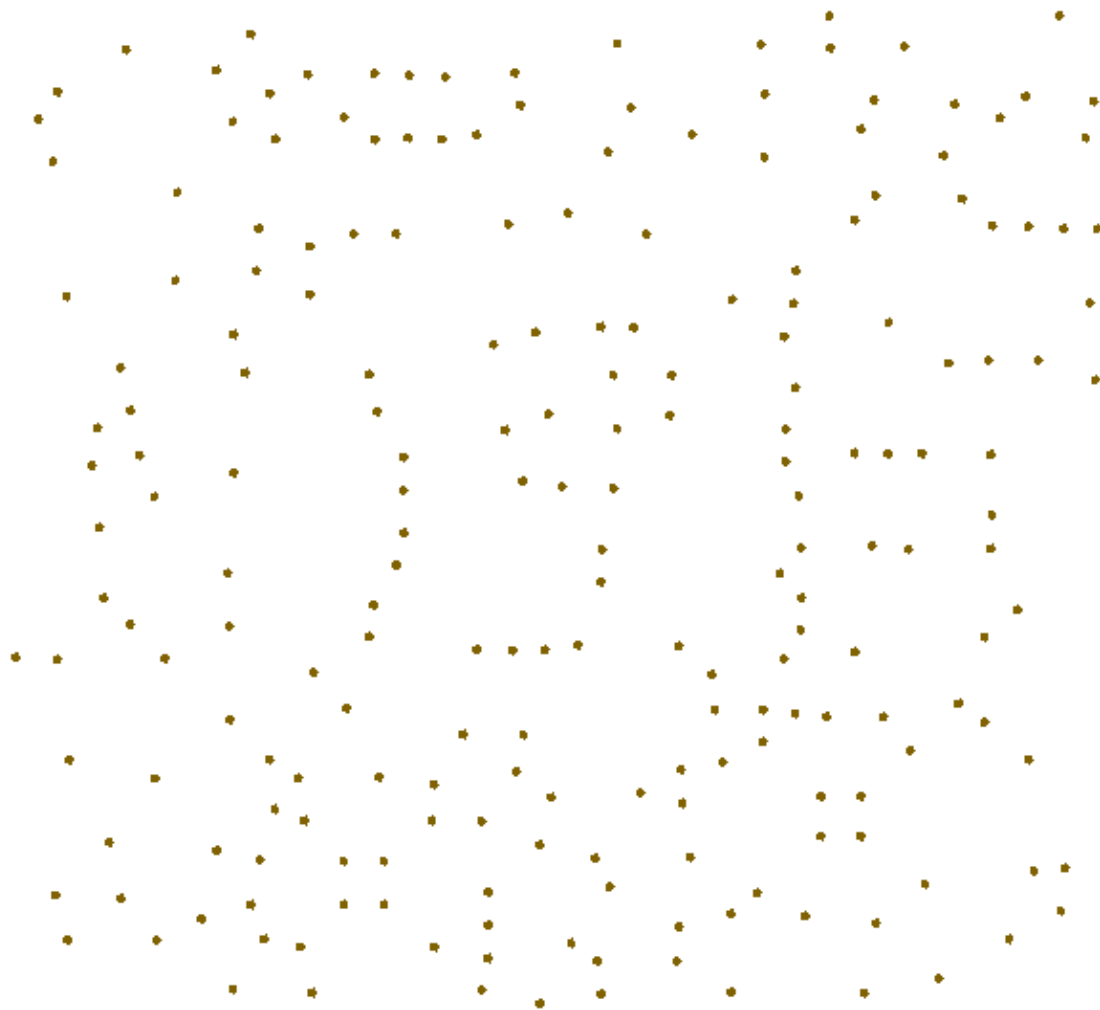
Serigrafía Inferior



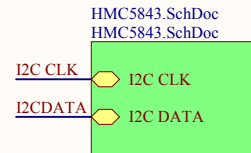
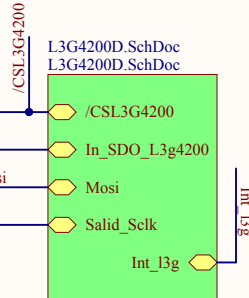
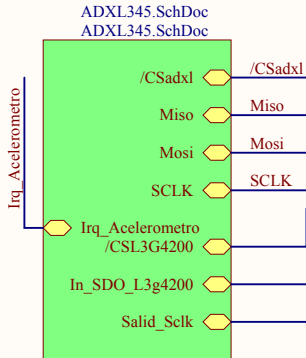
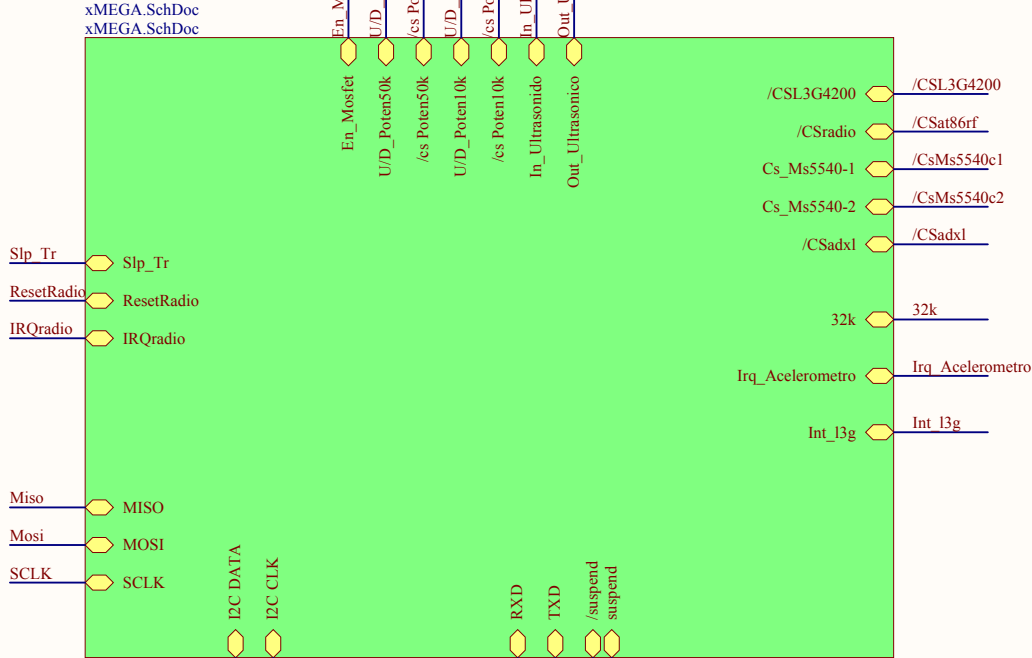
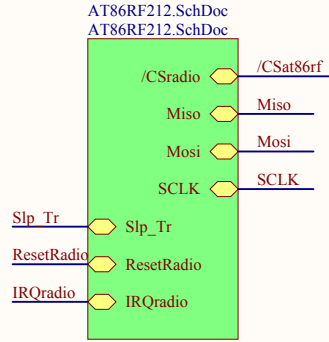
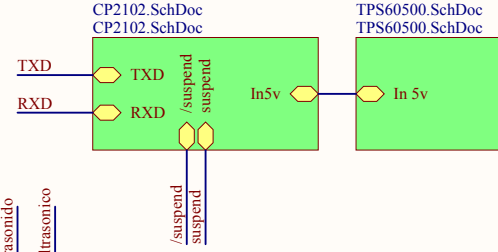
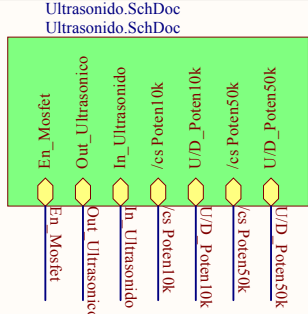
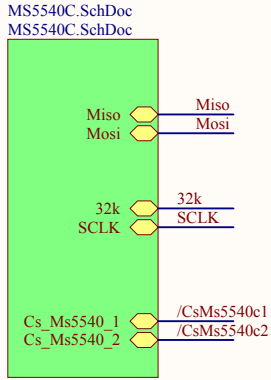
Máscara de Soldadura Superior



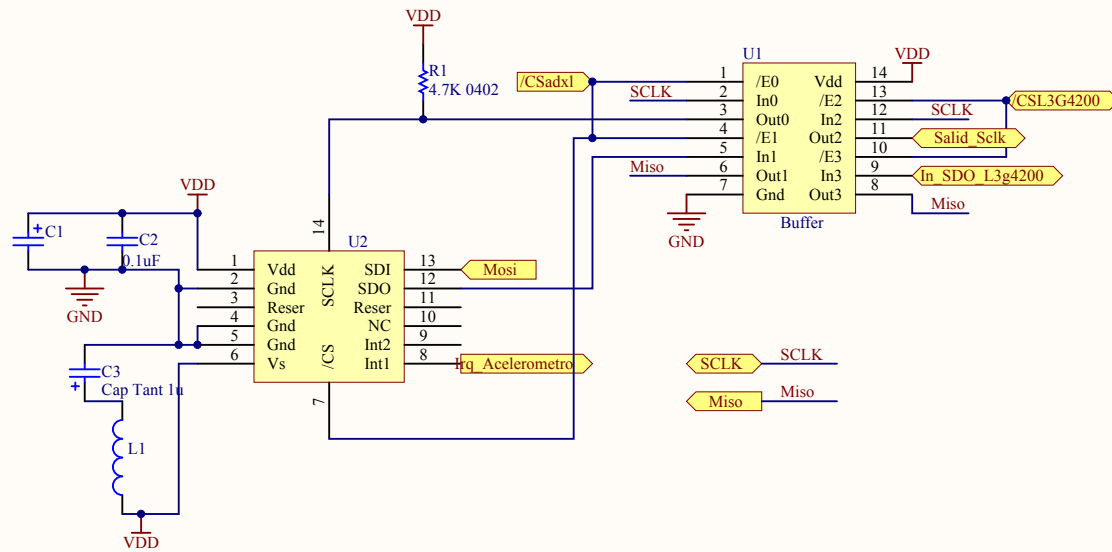
Máscara de Soldadura Inferior



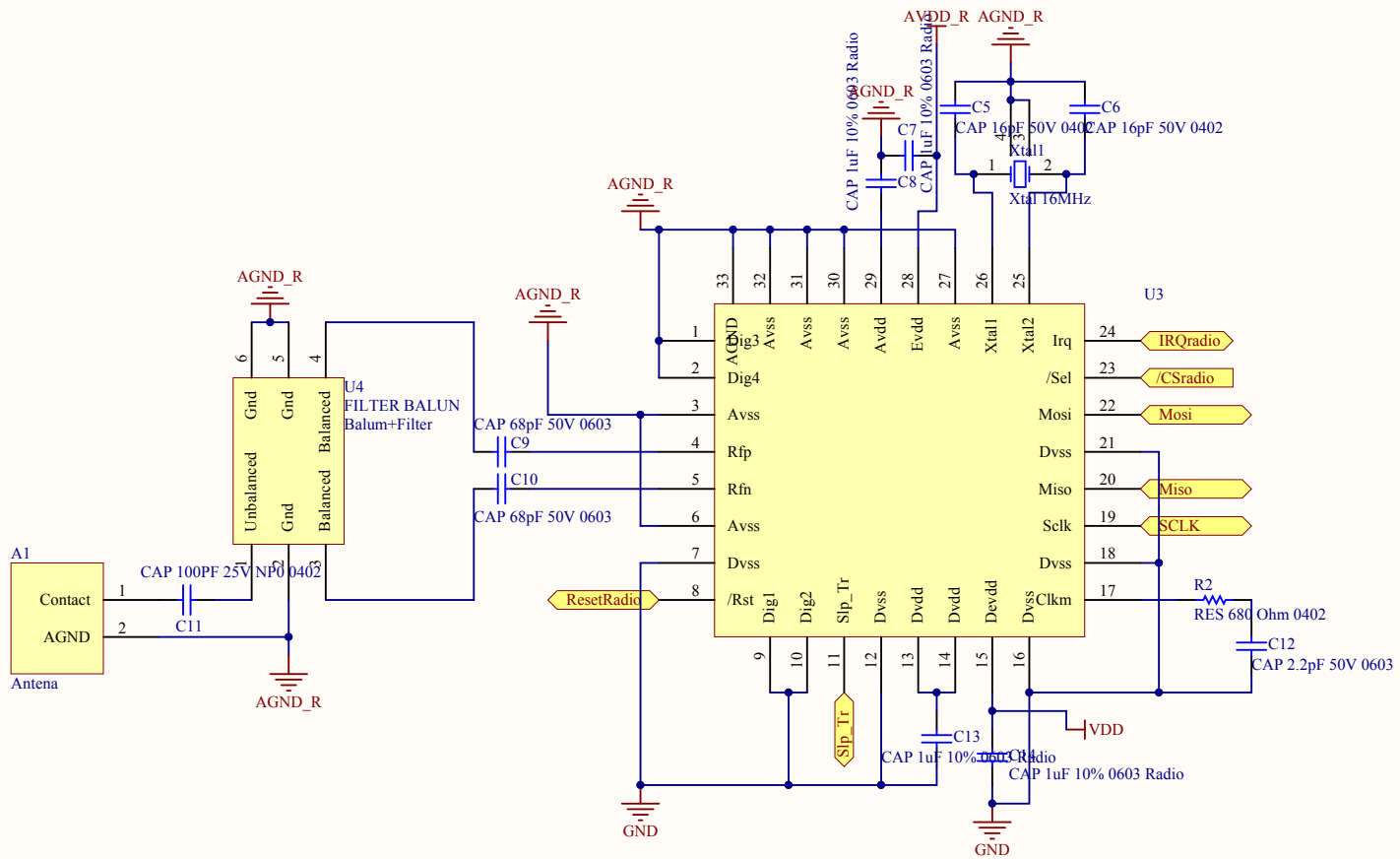
Puntos de Taladrado



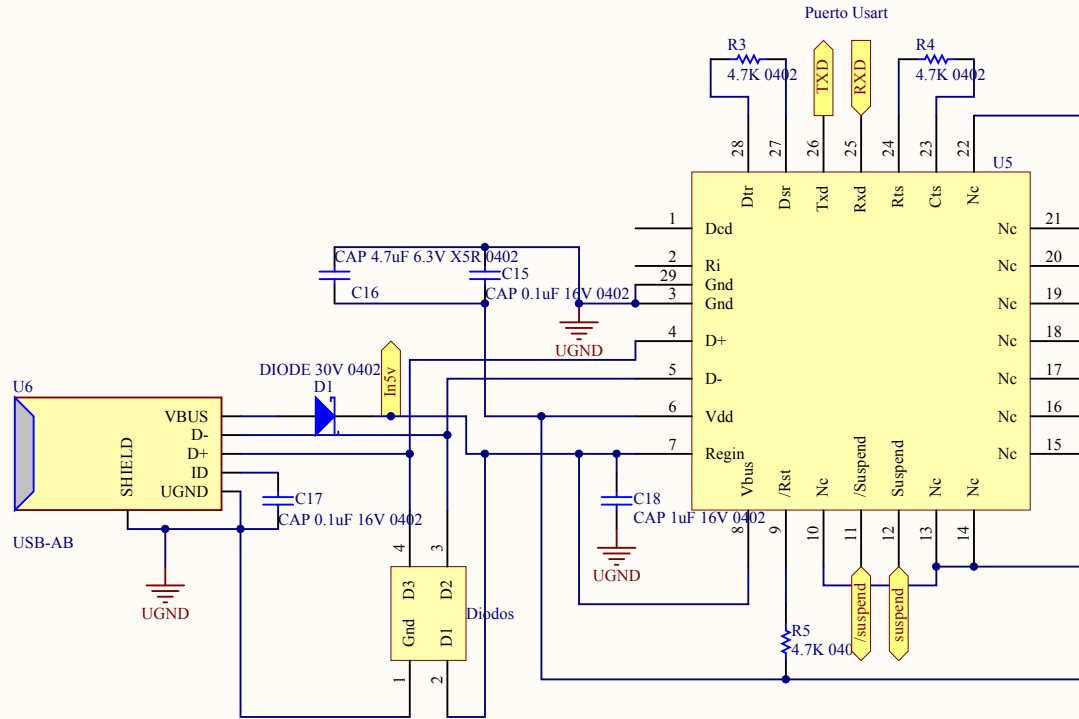
Title			Main.SchDoc		
Size	Number		Revision		
A4					
Date:	28/06/2012		Sheet of		
File:	D:\Proyecto\...\Main.SchDoc		Drawn By:		



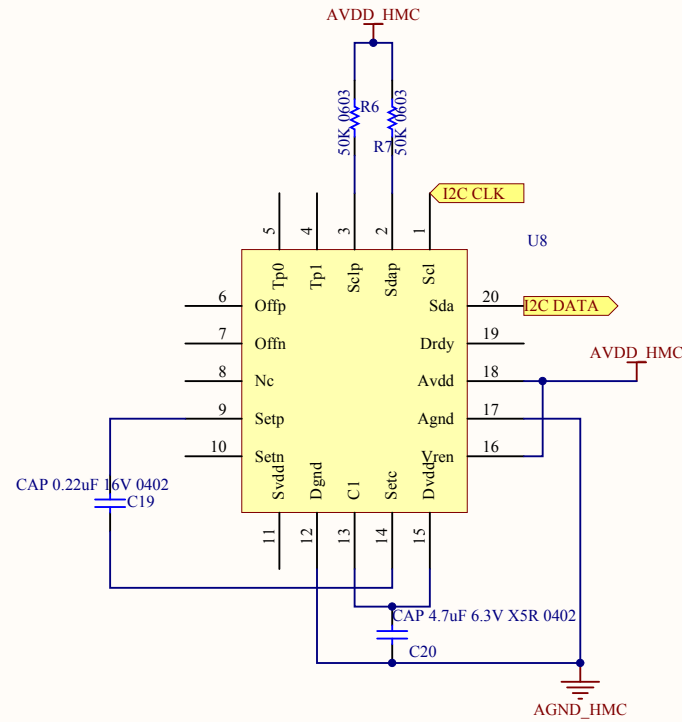
Title		
ADXL345.SchDoc		
Size	Number	Revision
A4		
Date:	28/06/2012	Sheet of
File:	D:\Proyecto\...\ADXL345.SchDoc	Drawn By:



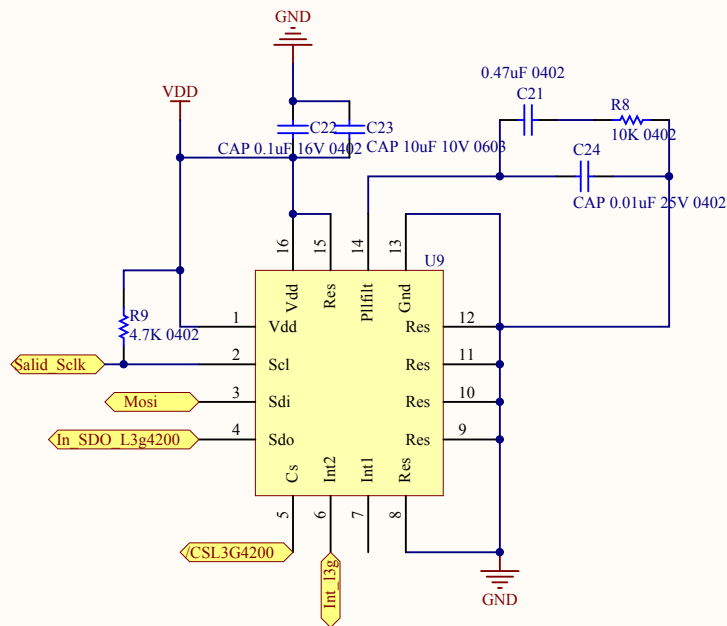
Title		
AT86RF212.SchDoc		
Size	Number	Revision
A4		
Date:	28/06/2012	Sheet of
File:	D:\Proyecto\...\AT86RF212.SchDoc	Drawn By:



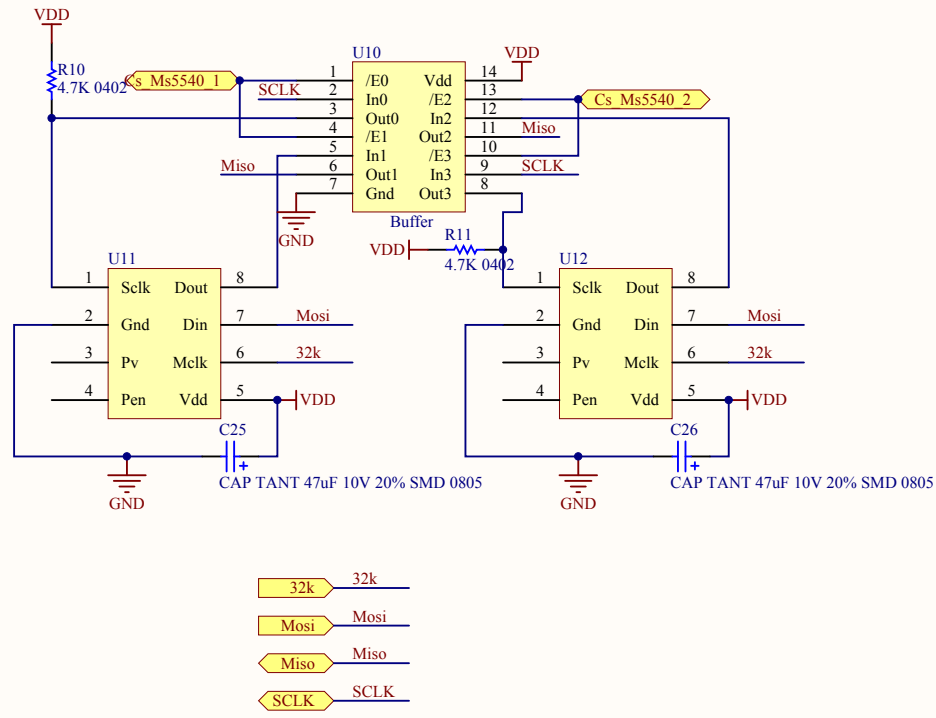
Title		
CP2102.SchDoc		
Size	Number	Revision
A4		
Date:	28/06/2012	Sheet of
File:	D:\Proyecto\...\CP2102_SchDoc	Drawn By:



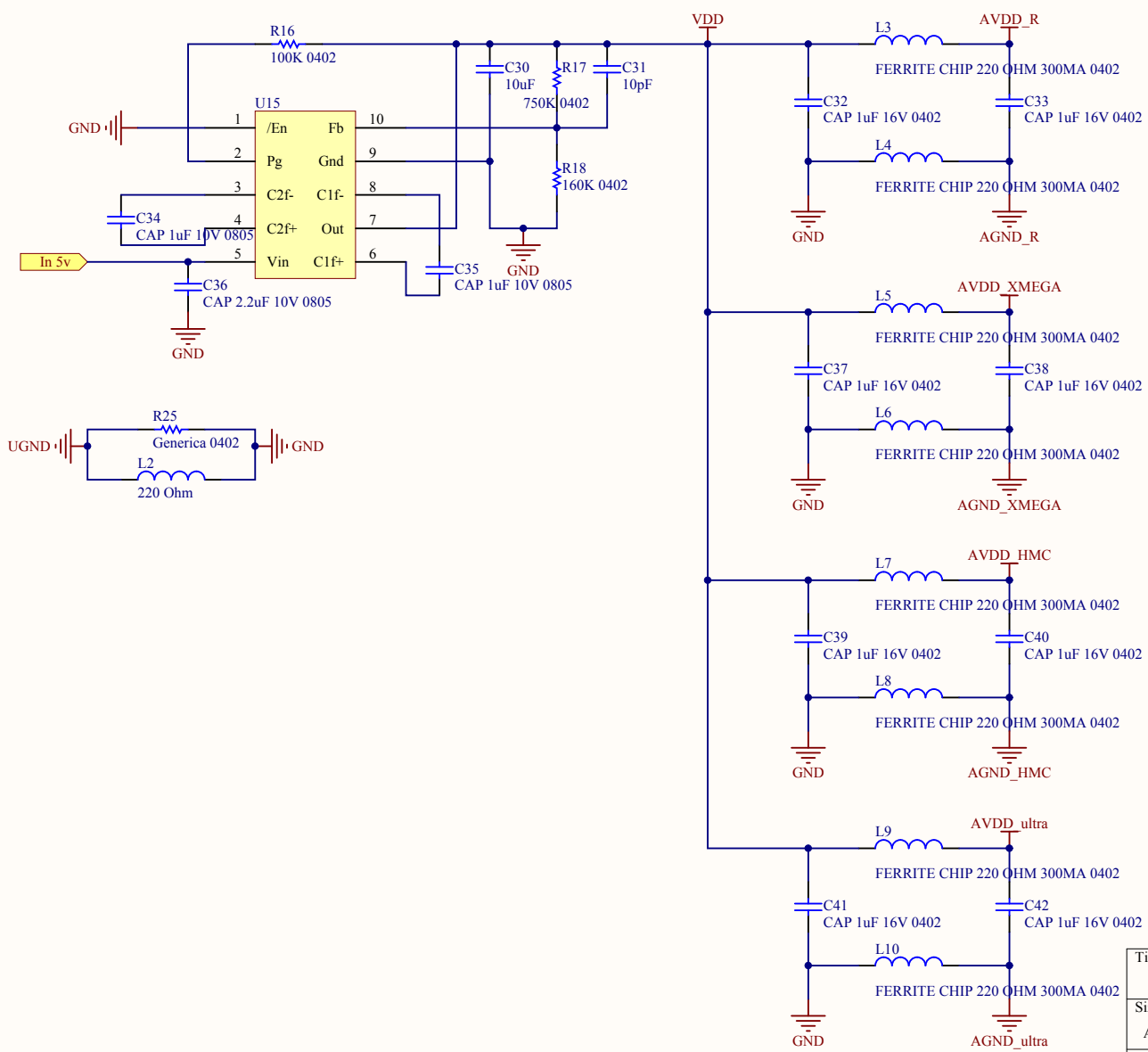
Title		
HMC5843.SchDoc		
Size	Number	Revision
A4		
Date:	28/06/2012	Sheet of
File:	D:\Proyecto\...\HMC5843.SchDoc	Drawn By:



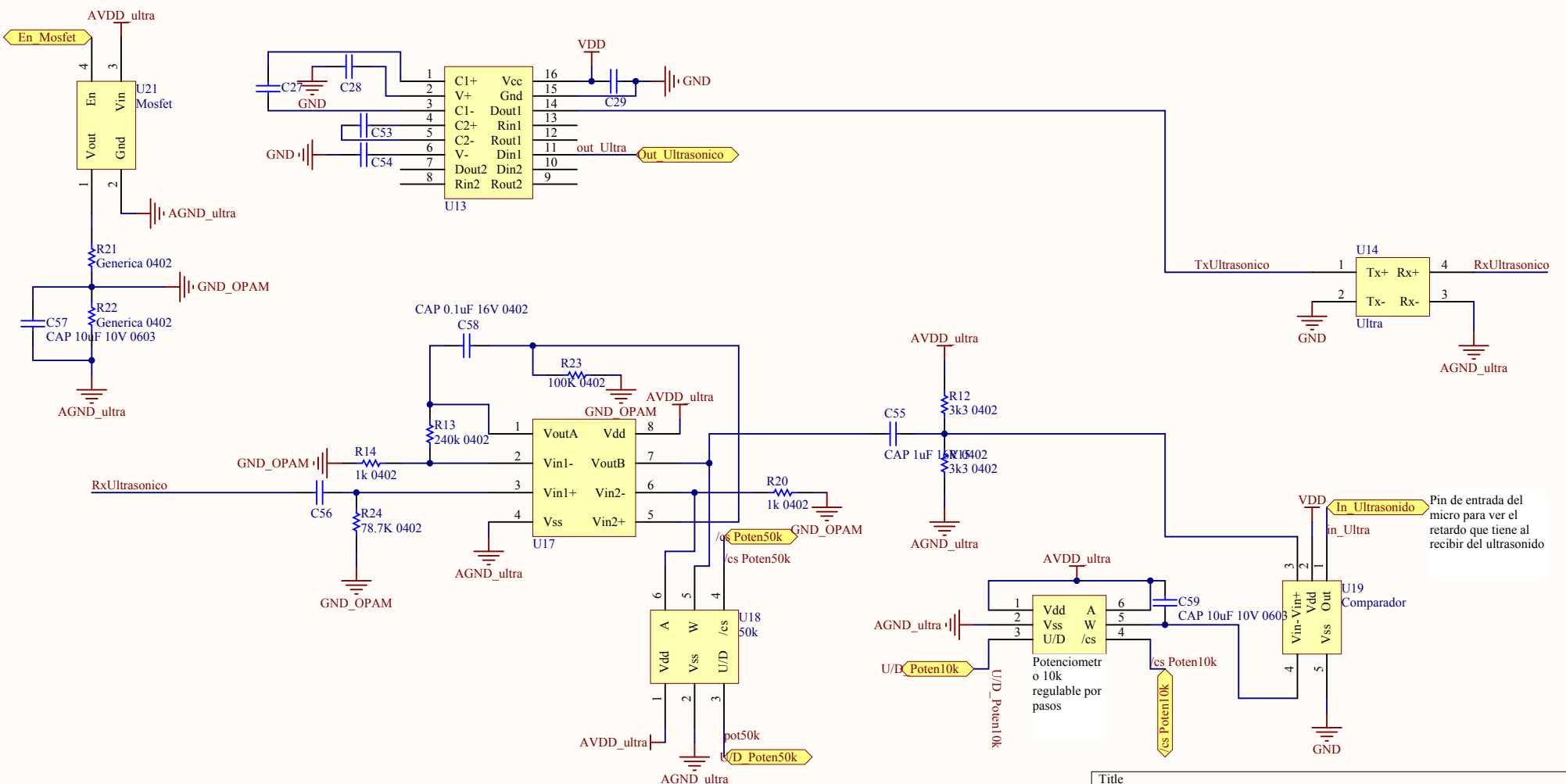
Title		
L3G4200D.SchDoc		
Size	Number	Revision
A4		
Date:	28/06/2012	Sheet of
File:	D:\Proyecto\...L3G4200D.SchDoc	Drawn By:



Title		
MS5540C.SchDoc		
Size	Number	Revision
A4		
Date:	28/06/2012	Sheet of
File:	D:\Proyecto\...\MS5540C.SchDoc	Drawn By:

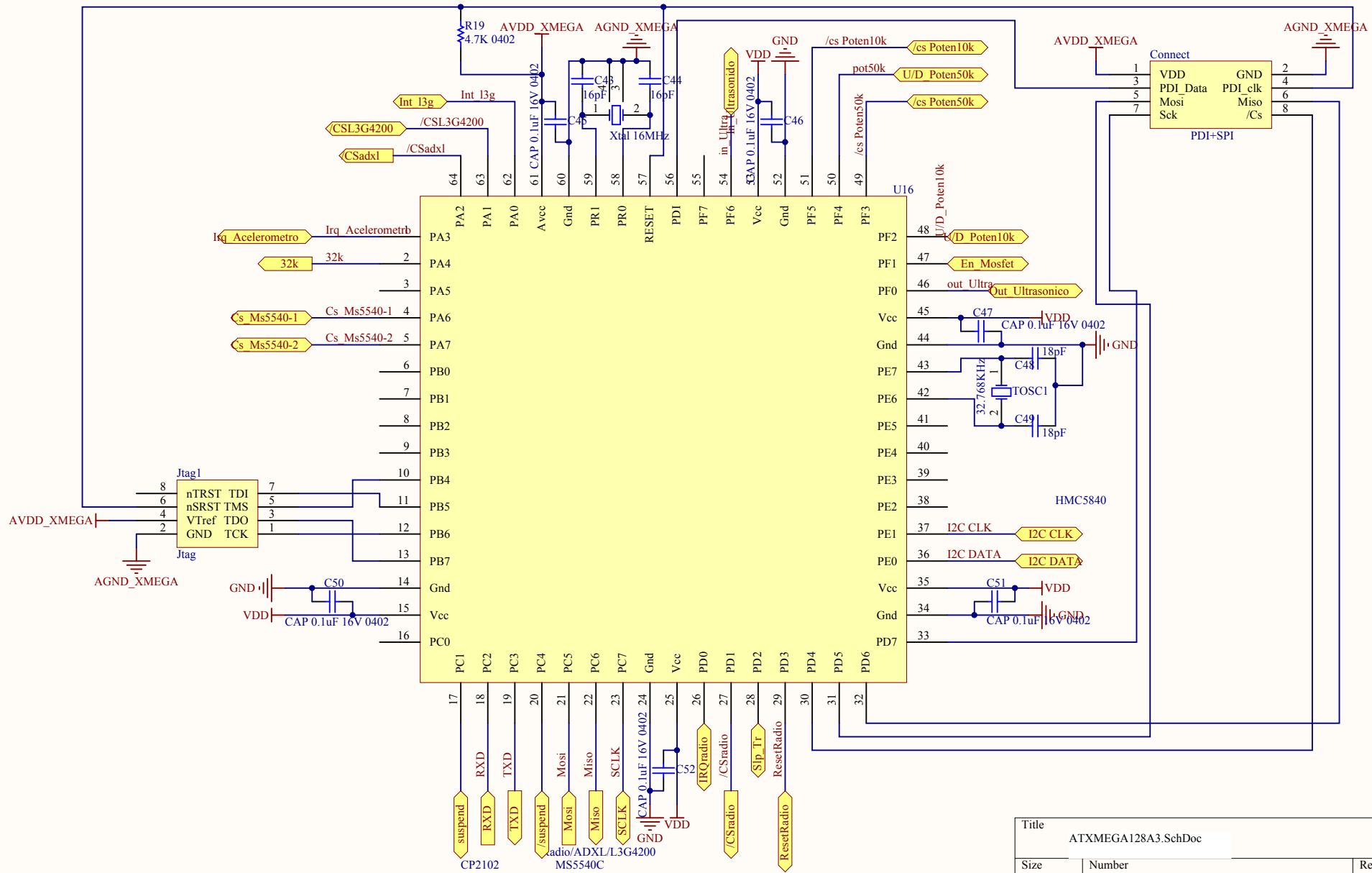


Title			TPS60500.SchDoc		
Size	Number		Revision		
A4					
Date:	28/06/2012		Sheet		of
File:	D:\Proyecto\...\TPS60500.SchDoc		Drawn By:		

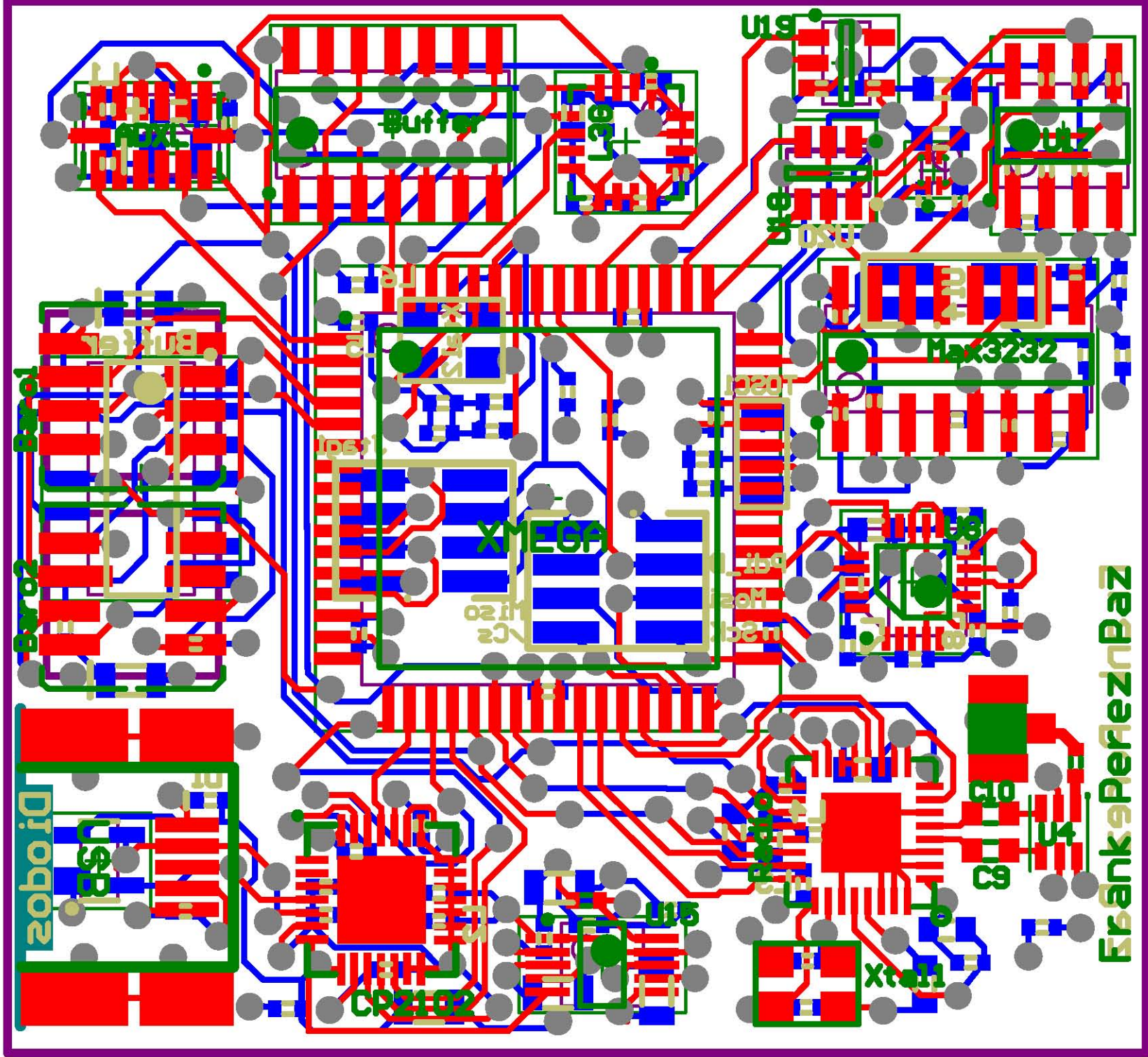


Pin de entrada del micro para ver el retardo que tiene al recibir del ultrasonido

Title			Ultrasonido.SchDoc		
Size	Number		Revision		
A4					
Date:	28/06/2012		Sheet of		
File:	D:\Proyecto\...\Ultrasonido.SchDoc		Drawn By:		



Title			ATXMEGA128A3.SchDoc		
Size	Number		Revision		
A4					
Date:	28/06/2012		Sheet of		
File:	D:\Proyecto\...XMEGA.SchDoc		Drawn By:		



EranksPerfeznPaz

Comment	Description	Designator	Footprint	LibRef	Quantity
Antena	RCPT IPLEX MHF	A1	Antena	Antena	1
CAP TANT 16V 100F SMD 0805	CAP TANT 16V 100F SMD 0805	C1	TANT 10uF 0805	CAP TANT 16V 100F SMD 0805	1
CAP 0.1uF 16V 0402	CAP 0.1uF 16V 0402	C2, C15, C17, C22, C27, C28, C29, C45, C46, C47, C50, C51, C52, C53, C54, C58	0402 NOPOL	CAP 0.1uF 16V 0402	16
Cap Tant 1u	CAP TANT 1uF 10V 0603	C3	0603 POL	Cap Tant 1u	1
CAP 16pF 50V 0402	CAP 16pF 50V 0402	C5, C6, C43, C44	0402 NOPOL	CAP 16pF 50V 0402	4
CAP 10uF 10V 0603	CAP 10uF 10V 0603 Radio	C7, C8, C13, C14	0603 NOPOL	CAP 10uF 10V 0603 Radio	4
CAP 68pF 50V 0603	CAP 68pF 50V 0603	C9, C10	0603 NOPOL	CAP 68pF 50V 0603	2
CAP 100pF 25V NPO 0402	CAP CERAMIC 100PF 25V NPO 0402	C11, C56	0402 NOPOL	CAP 100PF 25V NPO 0402	2
CAP 2.2pF 50V 0603	CAP 2.2pF 50V 0603	C12	0603 NOPOL	CAP 2.2pF 50V 0603	1
CAP 4.7uF 6.3V XSR 0402	CAP CER 4.7UF 6.3V XSR 0402	C16, C20	0402 NOPOL	CAP 4.7UF 6.3V XSR 0402	2
CAP 1uF 16V 0402	CAP 1uF 16V 0402	C18, C32, C33, C37, C38, C39, C40, C41, C42, C55	0402 NOPOL	CAP 1uF 16V 0402	10
CAP 0.22uF 16V 0402	CAP 0.22uF 16V 0402	C19	0402 NOPOL	CAP 0.22uF 16V 0402	1
0.47uF 0402	CAP 0.47UF 10V XSR 0402	C21	0402 NOPOL	CAP 0.47UF 10V XSR 0402	1
CAP 10uF 10V 0603	CAP 10uF 10V 0603	C23, C57, C59	0603 NOPOL	CAP 10uF 10V 0603	3
CAP 0.01uF 25V 0402	CAP 0.01uF 25V 0402	C24	0402 NOPOL	CAP 0.01uF 25V 0402	1
CAP TANT 47uF 10V 20% SMD 0805	CAP TANT 47UF 10V 20% SMD 0805	C25, C26	CAP TANT 0805	CAP TANT 47UF 10V 20% SMD 0805	2
10uF	CAP 10uF 6.3V 0805	C30	0805 NOPOL	CAP 10uF 6.3V 0805	1
10pF	CAP 10pF 25V 0402	C31	0402 NOPOL	CAP 10pF 50V 0402	1
CAP 1uF 10V 0805	CAP 1uF 10V 0805	C34, C35	0805 NOPOL	CAP 1uF 10V 0805	2
CAP 2.2uF 10V 0805	CAP 2.2uF 10V 0805	C36	0805 NOPOL	CAP 2.2uF 10V 0805	1
CAP 18pF 50V 0402	CAP 18pF 50V 0402	C48, C49	0402 NOPOL	CAP 18pF 50V 0402	2
PDI+SPI	CONN HEADER 8POS DL UNSHRD SMD	Connect	JTAG	PDI+SPI	1
DIODE 30V 0402	DIODE SCHOTTKY 200MA 30V 0402	D1	0402 Pol	DIODE 200mA 30V 0402	1
Jtag	CONN HEADER 8POS DL UNSHRD SMD	Jtag1	JTAG	Jtag	1
Ferr 100 OHM 0402	FERRITE BEAD 100 OHM 20A 0402	L1	0402 NOPOL	FERRITE 100 OHM 20A 0402	1
FERRITE CHIP 220 OHM 300MA 0402	FERRITE CHIP 220 OHM 300MA 0402	L2, L3, L4, L5, L6, L7, L8, L9, L10	0402 NOPOL	FERRITE CHIP 220 OHM 300MA 0402	9
4.7K 0402	RES 4.7K OHM 1/10W 5% 0402 SMD	R1, R3, R4, R5, R9, R10, R11, R19	0402 NOPOL	RES 4.7K OHM 1/10W 5% 0402 SMD	8
RES 680 Ohm 0402	RES 680 Ohm 0402	R2	0402 NOPOL	RES 680 Ohm 0402	1
50K 0603	RES 50K OHM 1/16W 0.1% 0603 SMD	R6, R7	0603 NOPOL	RES 50K 0603	2
10K 0402	RES 10.0K OHM 1/10W 1% 0402 SMD	R8	0402 NOPOL	RES 10.0K OHM 0402	1
3k3 0402	RES 3.3K 0402	R12, R15	0402 NOPOL	RES 3.3K OHM 1/10W 5% 0402 SMD	2
240k 0402	RES 240K 0402	R13	0402 NOPOL	RES 240K OHM 1/10W 5% 0402 SMD	1
1k 0402	RES 1K 0402	R14, R20	0402 NOPOL	RES 1.0K OHM 1/10W 5% 0402 SMD	2
100K 0402	RES 100K OHM 1/16W 5% 0402 SMD	R16, R23	0402 NOPOL	RES 100K 0402	2
750K 0402	RES 750K OHM 1/10W 5% 0402 SMD	R17	0402 NOPOL	RES 750K OHM 0402	1
160K 0402	RES 160K OHM 1/10W 1% 0402 SMD	R18	0402 NOPOL	RES 160K OHM 0402	1
Generica 0402	RES Ohm 0402	R21, R22, R25	0402 NOPOL	Generica 0402	3
78.7K 0402	RES 78.7K 0402	R24	0402 NOPOL	RES 78.7K OHM 1/10W 1% 0402 SMD	1
Tosc 32.768KHz	CRYSTAL 32.768KHZ 9PF SMD	TOSC1	Xtal 32.768KHz	Tosc 32.768KHz	1
Buffer	IC BUFFER 1R1ST QUAD LW V 14TSSOP	U1, U10	74LVX125	Buffer	2
ADXL345BCCZ-RL	IC ACCEL SPI 12C 3RA 3G 14LGA	U2	LG414	ADXL345BCCZ-RL	1
AT86RF212-ZU	IC TXRX ZIGBEE 802.15.4/ISM 32QFN	U3	32QFN2	AT86RF212-ZU	1
FILTER BALUN	FILTER BALUN CER CHIP 863-928MHZ	U4	Balum	FILTER BALUN	1
CP2102-GM	IC USB TO UART BRIDGE 28VQFN	U5	QFN 28pin	CP2102-GM	1
USB-AB	CONN RECEPT USB SPOS RT ANG SMD	U6	USB-MINI-AB-SMT	WM1722	1
SP0503BAHTG	TVS ARRAY 3 CH SOT143	U7	SOT143 - 4	SP0503BAHTG	1
HMCS843	SENSOR MAGNETIC 3 AXIS 20-LCC	U8	20 pin L-PCC16F	HMCS843	1
L3g4200D	GYROSCOPE MEMS 3-AXIS 16LGA	U9	L3G4200D	L3g4200D	1
MSS540-CM	MIDDLE BAROMETRIC MINITURE SENSOR	U11, U12	MSS540-CM	MSS540-CM	2
Max3232	IC RS232 3V3.5V 15KVESD 16-SOIC	U13	SOIC16pins	MAX3232	1
Ultra	HEADER .050" 4POS	U14	Ultras	Ultrasonido	1
TPS60500	IC CHARGE PUMP STPDN 250MA 10MSOP	U15	TPS605000	TPS60500	1
Xmega	MCU AVR 128K FLASH 1.8V 64-QFN	U16	ATXMEGA128A3-MH	Xmega	1
MCP6022	IC OPAMP 25V DUAL CIR RSOC	U17	SOIC127P600-8N	MCP6022	1
50k	POT 50K SOT23-6	U18	SOT23-6	MCP4012	1
Comparador	IC COMP 1.8V SINGL PPF SOT23-5	U19	SOT-23 lead-5	MCP6541R	1
Potenciometro 10K	IC DIGI POT 10K 1CH SOT23-6	U20	SOT23127P600-8N	MCP4013	1
Mosfet	IC LOAD SWITCH HI SIDE 2A 4-TMLF	U21	Mosfet	Mosfet	1
Xtal 16MHz	CRYSTAL 16.000000 MHz	Xtal1, Xtal2	XTAL3225	Xtal 16MHz	2

Código C

MAIN.C

```
#include <avr/io.h>

#include "main.h"

#include "baro.h"

#include "Spi.h"

#include "Usart.h"

//#include "rutinas.h"

#include "i2c.h"

const char texto[7][50] = {
/* 0 */    {0x0A,0x0D}, /* 0 */
/* 1 */    {"Iniciado"},
/* 2 */    {"Pulse 1 para ver los datos del acelerometro"},
/* 3 */    {"Pulse 2 para ver los datos del giroscopio"},
/* 4 */    {"Pulse 3 para ver los datos del aceler y giro"},
/* 5 */    {"Pulse 4 para ver la presion"},
};

char ejeXl,ejeYl,ejeZl,ejeXh,ejeYh,ejeZh;

long int j;

float Pressur_T_1,Pressur_T_2;

int main(){

    int i;

    char test_usart, sel_opt;

    init_puertos();                //Inicializamos puertos
```



```

confi_clock();           //configuramos reloj
confi_oscillator();     //configuramos oscilador
USART_Init();          //configuramos USART
confi_spi();           //Configuramos valores del SPI
comand_acel();         //Configuramos valores del ADXL
comand_giro();         //Configuramos valores del L3G
baro_inici(0x46);      //Leemos coeficientes barometro 1, n0x46
= /cs baro1

baro_inici(0x86);     //Leemos coeficientes barometro 2, n0x86 = /cs
baro2

//  iniciar_i2c();      //Habilitamos I2C
//  comandos_i2c();     //Mandamos Comandos

while(1){

    for(i=0;i<=5;i++) USART_Transmit(texto[0][i]);

    for(i=0;i<=12;i++) USART_Transmit(texto[1][i]); //Mandamos
un texto por la USART

    for(i=0;i<=5;i++) USART_Transmit(texto[0][i]);

    for(i=0;i<=42;i++) USART_Transmit(texto[2][i]);

    for(i=0;i<=5;i++) USART_Transmit(texto[0][i]);

    for(i=0;i<=40;i++) USART_Transmit(texto[3][i]);

    for(i=0;i<=5;i++) USART_Transmit(texto[0][i]);

    for(i=0;i<=43;i++) USART_Transmit(texto[4][i]);

    for(i=0;i<=5;i++) USART_Transmit(texto[0][i]);

    for(i=0;i<=50;i++) USART_Transmit(texto[5][i]);

    for(i=0;i<=5;i++) USART_Transmit(texto[0][i]);

    test_usart=USART_Receive();           //Esperamos a recibir
                                           un dato de la USART

    sel_opt=test_usart;

    USART_Transmit(test_usart); //lo volvemos a enviar para
comprobar que a sido leido correctamente

```

```

        i=20;

        while(i>=0){
            //Hacemos que muestre los
valores 1000 veces

            if((sel_opt==0x31)|(sel_opt==0x33)){

                datos_acelerom(); //Obtenemos los datos del
acelerometro

                visualizar(sel_opt); //Configuramos los
datos para mostrarlos correctamente 0-9

                for(j=0;j<=0xffff;j++); //Retardos para darnos
tiempo a ver los valores

            }

            if((sel_opt==0x32)|(sel_opt==0x33)){

                datos_giro(); //Obtenemos los datos
del giroscopio

                visualizar(sel_opt);

                for(j=0;j<=0xffff;j++);

            }

            if(sel_opt==0x34){

                baro_temp_press(0x46);

                baro_temp_press(0x86);

                calcular_presion();

            }

            i--;

        }

    }

    return 0;

}

void init_puertos(void){ //Configuramos los puertos de entrada
y salida

    PORTA.DIR=0xC6;

```

```
PORTA.OUT=0xC6;

PORTB.DIR=0x80;

PORTB.OUT=0x80;

PORTC.DIR=0xA8;

PORTC.OUT=0xA8;

PORTD.DIR=0xBE;

PORTD.OUT=0xB2;

PORTE.DIR=0x02;

PORTE.OUT=0x02;

PORTF.DIR=0x3F;

PORTF.OUT=0x3F;

}
```

```
void retar_r(void) {

    int i;

    i=0xffff;

    while(i==0) {

        i--;

    }

}
```

SPI.C

```
#include <avr/io.h>

#include "Usart.h"

#include "Spi.h"

unsigned char
ace_ejeX1,ace_ejeY1,ace_ejeZ1,ace_ejeXh,ace_ejeYh,ace_ejeZh,temp,optio
ns;

unsigned char
gir_ejeX1,gir_ejeY1,gir_ejeZ1,gir_ejeXh,gir_ejeYh,gir_ejeZh;

unsigned char ejeX0,ejeX1,ejeX2,ejeY0,ejeY1,ejeY2,ejeZ0,ejeZ1,ejeZ2;

char aux,aux1,identi1,identi2;

    char angX0,angX1,angX2,angX3;

    char angY0,angY1,angY2,angY3;

    char angZ0,angZ1,angZ2,angZ3;

//*****

//                SPI

//*****

void confi_spi(void)

{

    SPIC.CTRL = 0X50; //Habilitamos SPI,MSB primero,Master, clk/16

}

void transmit_spic(char reg)

{

    while((SPIC.STATUS & 0x40)); //esperamos flag

    SPIC.DATA=reg;                //metemos dato en el
registro del spi

    while(!(SPIC.STATUS & 0x80)); //esperamos que lo envíe

}
```

```

char resivir_spic(void)
{
    while(!(SPIC.STATUS & 0X80)); //esperamos que entre un dato
    return SPIC.DATA;           //retornamos dato
}

//*****
//                               ADXL345
//*****

void comand_acel(void){

    PORTA_OUT= 0xC2;

    transmit_spic(0x27); //Axis enable control for activity and
inactivity detection

    transmit_spic(0x70);

    PORTA_OUT=0XC6;

    PORTA_OUT=0XC2;

    transmit_spic(0x28); //Free-fall threshold

    transmit_spic(0x00);

    PORTA_OUT=0XC6;

    PORTA_OUT=0XC2;

    transmit_spic(0x29); //Free-fall time

    transmit_spic(0x00);

    PORTA_OUT=0XC6;

    PORTA_OUT=0XC2;

    transmit_spic(0x2C); //Data rate and power mode control

```

```

transmit_spic(0x0A);

PORTA_OUT=0XC6;

PORTA_OUT=0XC2;

transmit_spic(0x2D); //Power ctrl
transmit_spic(0x08);
PORTA_OUT=0XC6;

PORTA_OUT=0XC2;
transmit_spic(0x31); //Data format control
transmit_spic(0x00);
PORTA_OUT=0XC6;
}

void datos_acelerom(void)
{

PORTA_OUT=0XC2; //activamos /Cs del
acelerometro

transmit_spic(0x80); //mandamos registro identificativo
transmit_spic(0xee); //mandamos dato aleatorio
temp=SPIC.DATA; //guardamos X0
PORTA_OUT=0XC6; //desactivamos /Cs

PORTA_OUT=0XC2; //activamos /Cs del
acelerometro

transmit_spic(0xB2); //mandamos registro dato X1, modo
Rd=1

transmit_spic(0xee); //mandamos dato aleatorio
ace_ejeX1=SPIC.DATA; //guardamos X0
PORTA_OUT=0XC6; //desactivamos /Cs

```

```

        PORTA_OUT=0XC2;           //activamos /Cs del
acelerometro

        transmit_spic(0xB3);     //mandamos registro dato Xh

        transmit_spic(0xee);     //mandamos dato aleatorio

        ace_ejeXh=SPIC.DATA;     //guardamos Xh

        PORTA_OUT=0XC6;         //desactivamos /Cs

//
        acelX=temp+(temp1<<4); //guardamos X1X0;

        PORTA_OUT=0XC2;

        transmit_spic(0xB4);     //mandamos registro dato Yl

        transmit_spic(0xee);

        ace_ejeYl=SPIC.DATA;

        PORTA_OUT=0XC6;

        PORTA_OUT=0XC2;

        transmit_spic(0xB5);     //mandamos registro dato Yh

        transmit_spic(0xee);

        ace_ejeYh=SPIC.DATA;

        PORTA_OUT=0XC6;

        PORTA_OUT=0XC2;

        transmit_spic(0xB6);     //mandamos registro dato Zl

        transmit_spic(0xee);

        ace_ejeZl=SPIC.DATA;

        PORTA_OUT=0XC6;

        PORTA_OUT=0XC2;

        transmit_spic(0xB7);     //mandamos registro dato Zh

        transmit_spic(0xee);

        ace_ejeZh=SPIC.DATA;

        PORTA_OUT=0XC6;

```

```
        confhexad(ace_ejeXl,ace_ejeYl,ace_ejeZl,ace_ejeXh,ace_ejeYh,ace_
ejeZh);
```

```
        options=1;
```

```
    }
```

```
//*****
```

```
//                L3G
```

```
//*****
```

```
void comand_giro(void){
```

```
    PORTA_OUT=0XC4;
```

```
    transmit_spic(0x20);    //CTRL_REG1 (20h)
```

```
    transmit_spic(0x0F);    //Output data rate=95;XYZ axis enable
```

```
    PORTA_OUT=0XC6;
```

```
}
```

```
void datos_giro(void){
```

```
    PORTA_OUT=0XC4;
```

```
    transmit_spic(0x8F);    //mandamos registro identificativo
```

```
    transmit_spic(0xee);
```

```
    temp=SPIC.DATA;        //guardamos dato identificativo
```

```
    PORTA_OUT=0XC6;
```

```
    PORTA_OUT=0XC4;
```

```
    transmit_spic(0xE8);    //mandamos registro dato Xl
```

```
    transmit_spic(0xee);
```

```
    gir_ejeXl=SPIC.DATA;    //guardamos dato Xl
```

```
    transmit_spic(0xee);    //escritura aleatoria para que se
incremente automaticamente la direccion del registro
```



```

        gir_ejeXh=SPIC.DATA;          //guardamos dato Xh
//
        PORTA_OUT=0XC6;

//
        PORTA_OUT=0XC4;
//
        transmit_spic(0xEa);
        transmit_spic(0xee); //repetimos todo mandando dato
aleatorio para que
        gir_ejeYl=SPIC.DATA;          //siga incrementando la
direccion de los registros
        transmit_spic(0xee); //y seguimos guardando los
respectivos datos
        gir_ejeYh=SPIC.DATA;
//
        PORTA_OUT=0XC6;

//
        PORTA_OUT=0XC4;
//
        transmit_spic(0xEc);
        transmit_spic(0xee);
        gir_ejeZl=SPIC.DATA;
        transmit_spic(0xee);
        gir_ejeZh=SPIC.DATA;
        PORTA_OUT=0XC6;

        confhexad(gir_ejeXl,gir_ejeYl,gir_ejeZl,gir_ejeXh,gir_ejeYh,gir_
ejeZh);

        options=2;
}

//*****
//
//          RUTINA PARA TRANSFORMAR A ASCII
//*****

void confhexad(char ejeXl,char ejeYl,char ejeZl

```

```

    ,char ejeXh,char ejeYh,char ejeZh){
//    int Xvalor,Yvalor,Zvalor;

//*****EJE X*****

    aux=(ejeXl&0x0f);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

    angX0=aux1;

    aux=((ejeXl&0xf0)>>4);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

    angX1=aux1;

    aux=(ejeXh&0x0f);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

    angX2=aux1;

    aux=((ejeXh&0xf0)>>4);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

    angX3=aux1;

//*****EJE Y*****

    aux=(ejeYl&0x0f);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

```

```

angY0=aux1;

    aux=((ejeYl&0xf0)>>4);
    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/
    else aux1=aux+0x30; /*Caracter ascii*/
    angY1=aux1;

    aux=(ejeYh&0x0f);
    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/
    else aux1=aux+0x30; /*Caracter ascii*/
    angY2=aux1;

    aux=((ejeYh&0xf0)>>4);
    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/
    else aux1=aux+0x30; /*Caracter ascii*/
    angY3=aux1;

//*****EJE Z*****

    aux=(ejeZl&0x0f);
    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/
    else aux1=aux+0x30; /*Caracter ascii*/
    angZ0=aux1;

    aux=((ejeZl&0xf0)>>4);
    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/
    else aux1=aux+0x30; /*Caracter ascii*/
    angZ1=aux1;

```

```

    aux=(ejeZh&0x0f);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

    angZ2=aux1;

    aux=((ejeZh&0xf0)>>4);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

    angZ3=aux1;

//*****identificativo*****

    aux=(temp&0x0f);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

    identil=aux1;

    aux=((temp&0xf0)>>4);

    if(aux>=0x0a) aux1=aux+0x37; /*Desplazados, y lo transformamos
a su*/

    else aux1=aux+0x30;          /*Caracter ascii*/

    identi2=aux1;
}

```

```

void visualizar(char option){

    if((option == 0x31 || option == 0x33) && options == 1){

        USART_Transmit(0x0A); //salto Linea

        USART_Transmit(0x0D); //retroceso carro

        USART_Transmit('A');
    }
}

```

```

        USART_Transmit('D');

        USART_Transmit('X');

        USART_Transmit('L');
    }

    if((option ==0x32 || option == 0x33) && options ==2){
        if(option == 0x32){
            USART_Transmit(0x0A);    //salto Linea
            USART_Transmit(0x0D);    //retroceso carro
        }

        USART_Transmit(' ');
        USART_Transmit('L');
        USART_Transmit('3');
        USART_Transmit('G');
    }else;

    USART_Transmit('X');
    USART_Transmit('=');
    USART_Transmit(angX3);
    USART_Transmit(angX2);
    USART_Transmit(angX1);
    USART_Transmit(angX0);
    USART_Transmit('_');
    USART_Transmit('_');

    // USART_Transmit(0x08);    //retroceso
    // USART_Transmit(0x0A);    //salto Linea
    // USART_Transmit(0x0D);    //retroceso del carro
    // USART_Transmit(0x01);    //inicio encabezado
    // USART_Transmit(0x02);    //inicio texto

```

```

    USART_Transmit('Y');
    USART_Transmit('=');
    USART_Transmit(angY3);
    USART_Transmit(angY2);
    USART_Transmit(angY1);
    USART_Transmit(angY0);
    USART_Transmit('_');
    USART_Transmit('_');
//    USART_Transmit(0x0A);    //salto Linea
//    USART_Transmit(0x0D);    //retroceso carro

    USART_Transmit('Z');
    USART_Transmit('=');
    USART_Transmit(angZ3);
    USART_Transmit(angZ2);
    USART_Transmit(angZ1);
    USART_Transmit(angZ0);
    USART_Transmit('_');
    USART_Transmit('_');

    USART_Transmit('i');
    USART_Transmit('=');
    USART_Transmit(identi2);
    USART_Transmit(identi1);

}

```

USART . C

```
#include <avr/io.h>

/*****
/*          USART          */
*****/

void USART_Init( void )
{
    USARTC0.CTRLB |= USART_RXEN_bm;    //Habilitamos transmision y
repcion
    USARTC0.CTRLB |= USART_TXEN_bm;
    USARTC0.CTRLC=0X03;    //Habilitamos 8bit R&T
    USARTC0.BAUDCTRLA=0X0b; //BSEL=11 para baud rate de 115200
    USARTC0.BAUDCTRLB=0X90; //BSCALE= -7 (0b1001);
}

void USART_Transmit( unsigned char dato )
{
    while(!(USARTC0_STATUS & 0X20));    //Examinamos DREIF (buffer
transmicion vacio)
    USARTC0.DATA = dato;                //Mandamos dato al
buffer
}

unsigned char USART_Receive( void )
{
    while(!(USARTC0_STATUS & 0X80));    //Examinamos RXCIF (recepcion
completada)
    return USARTC0.DATA;                //Retornamos dato de
entrada de la usart
}
```

TIMER.C

```
#include <avr/io.h>

void confi_clock( void ){

    CLK.CTRL=0x03;          //HABILITAMOS RELOJ EXTERNO(tosc)
    CLK.PSCTRL=0x00;      //NO UTILIZAMOS DIVISORES PRESCALERS
    // CLK.RTCCTRL=0x0B; //HABILITAMOS TOSC PARA 32.768K
}

void confi_oscillator( void ){
    OSC.CTRL=0x09;        //HABILITAMOS OSCILADOR EXTERNO(xtal)
    OSC.XOSCCTRL=0xC0;    //FRECUENCIA DE 12-16mhz
    OSC.PLLCTRL=0xC1;    //Fuente reloj "external oscillator"
}

```

RUTINAS.C

```
#include <avr/io.h>

void waiting(int retardo){

    while(retardo >= 0)
        retardo--;

}

```


I2C.C

```
#include <avr/io.h>

#include "i2c.h"

char Bruj_X_M, Bruj_X_L, Bruj_Y_M, Bruj_Y_L, Bruj_Z_M, Bruj_Z_L;

void transmit_addr_i2c(char addr){

    TWIE_MASTER_ADDR = addr;

    while(TWIE_MASTER_STATUS && 0x08); //Esperar ACK del esclavo
}

void transmit_dato_i2c(char dato){

    TWIE_MASTER_DATA = dato;

    while(TWIE_MASTER_STATUS && 0x08); //Esperar ACK del esclavo
}

char leer_i2c(void){

    char dato;

    while(TWIE_MASTER_STATUS && 0x80); //Esperamos activar flag de
lectura

    dato = TWIE_MASTER_DATA;

    return dato;
}

void datos_i2c(void){

    transmit_addr_i2c(0x3D); //Comando de escritura

    Bruj_X_M = leer_i2c;

    Bruj_X_L = leer_i2c;

    Bruj_Y_M = leer_i2c;
```

```

    Bruj_Y_L = leer_i2c;

    Bruj_Z_M = leer_i2c;

    Bruj_Z_L = leer_i2c;

}

void iniciar_i2c(void){

    TWIE_MASTER_CTRLA = 0x38; //Enable Twi

    TWIE_MASTER_CTRLC = 0x02; //Comand modo 10, en (/W) no hace
nada, en R manda ack

}

void comandos_i2c(void){

    transmit_addr_i2c(0x3C); //Comando de escritura

    transmit_dato_i2c(0x02); //Registro del HMC

    transmit_dato_i2c(0x00); //Codigo para tomar medidas siempre

}

```

BARO.C

```

#include <avr/io.h>

#include "Baro.h"

#include "Spi.h"

#include "Usart.h"

char coef_a0_M, coef_a0_L, coef_b1_M, coef_b1_L, coef_b2_M,
coef_b2_L,aux;

char coef_c12_M, coef_c12_L, presure_M, presure_L, temper_M,temper_L;

char sing_a0,sing_b1,sing_b2,sing_c12;

int temperature, presure,coef_a0_1,coef_b1_1,coef_b2_1,coef_c12_1,x;

int coef_a0_2,coef_b1_2,coef_b2_2,coef_c12_2;

```

```

float a0,b1,b2,c12,a0_1,b1_1,b2_1,c12_1,a0_2,b1_2,b2_2,c12_2;
float c12x2,a1,a1x1,y1,a2x2,PComp,Pressure_total_1,Pressure_total_2;

void baro_temp_press(char n_baro){

    PORTA_OUT = n_baro;// activamos /CS del barometro 2
    transmit_spic(0x24);
    transmit_spic(0x00);
    PORTA_OUT = 0xC6;
    for(x=0;x<=0x5fff;x++); //esperamos 2milisegundos aprox

    PORTA_OUT = n_baro;    //Volvemos Activar /CS para leer datos de
presion
    transmit_spic(0x80);    //y temperatura
    transmit_spic(0x00);
    presure_M = SPIC.DATA;    //guardamos los datos leidos
    transmit_spic(0x82);
    transmit_spic(0x00);
    presure_L = SPIC.DATA;

    transmit_spic(0x84);
    transmit_spic(0x00);
    temper_M = SPIC.DATA;
    transmit_spic(0x86);
    transmit_spic(0x00);
    temper_L = SPIC.DATA;
    PORTA_OUT = 0xC6; //descativamos /CS

    presure = ((presure_M<<8)+presure_L);
    temperature = ((temper_M<<8) + temper_L);

```

```

    pressure = ((pressure>>6) & 0x03ff);           //Los datos
pressure y temperature los dejamos

    temperature = ((temperature>>6) & 0x03ff);     //como tienen que
estar porque son n° enteros.

    if(n_baro==0x46){

        c12x2 = c12_1 * temperature;

        a1 = b1_1 + c12x2;

        a1x1 = a1 * pressure;

        y1 = a0_1 + a1x1;

        a2x2 = b2_1 * temperature;

        PComp = y1 + a2x2;

        Pressure_total_1 = (PComp*0.06353)+50;

    }

    else{

        c12x2 = c12_2 * temperature;

        a1 = b1_2 + c12x2;

        a1x1 = a1 * pressure;

        y1 = a0_2 + a1x1;

        a2x2 = b2_2 * temperature;

        PComp = y1 + a2x2;

        Pressure_total_2 = (PComp*0.06353)+50;

    }

}

void baro_inici(char nbaro){

    PORTA_OUT = nbaro;// activamos /CS del barometro 2

    transmit_spic(0x88);           //Empezamos accediendo a la lectura
de sus registros

    transmit_spic(0x00);

```

```

    coef_a0_M = SPIC.DATA;           //Vamos guardando los diferentes
coeficientes en una variable

    transmit_spic(0x8A);

    transmit_spic(0x00);

    coef_a0_L = SPIC.DATA;

    transmit_spic(0x8C);

    transmit_spic(0x00);

    coef_b1_M = SPIC.DATA;

    transmit_spic(0x8E);

    transmit_spic(0x00);

    coef_b1_L = SPIC.DATA;

    transmit_spic(0x90);

    transmit_spic(0x00);

    coef_b2_M = SPIC.DATA;

    transmit_spic(0x92);

    transmit_spic(0x00);

    coef_b2_L = SPIC.DATA;

    transmit_spic(0x94);

    transmit_spic(0x00);

    coef_c12_M = SPIC.DATA;

    transmit_spic(0x96);

    transmit_spic(0x00);

    coef_c12_L = SPIC.DATA;

    transmit_spic(0x00);

    PORTA_OUT = 0xC6; //Desactivamos /CS

    if(nbaro == 0x46){

```

```

        a0_1 = ((coef_a0_M<<8)+coef_a0_L); //Aqui tendremos los
coeficientes en un registro de 16bits

        b1_1 = ((coef_b1_M<<8)+coef_b1_L);

        b2_1 = ((coef_b2_M<<8)+coef_b2_L);

        c12_1 = ((coef_c12_M<<8)+coef_c12_L);

    }

    else {

        a0_2 = ((coef_a0_M<<8)+coef_a0_L); //Aqui tendremos los
coeficientes en un registro de 16bits

        b1_2 = ((coef_b1_M<<8)+coef_b1_L);

        b2_2 = ((coef_b2_M<<8)+coef_b2_L);

        c12_2 = ((coef_c12_M<<8)+coef_c12_L);

    }

    transf_a_float(nbaro);

}

```

```

void transf_a_float(char nbaro){

    if(nbaro==0x46){

        a0_1 = a0_1/8;

        b1_1 = b1_1/8192;

        b2_1 =b2_1/16384;

        c12_1=c12_1/16777216;

    }

    else{

        a0_2 = a0_2/8;

        b1_2 = b1_2/8192;

        b2_2 =b2_2/16384;

        c12_2=c12_2/16777216;

    }

}

```

```
}
```

```
void calcular_presion(void) {  
    char centena, decena, unidad, decima, centecima;  
    int auxiliar1, auxiliar2;  
  
    auxiliar1 = Pressure_total_1*100;  
    auxiliar2 = Pressure_total_2*100;  
  
    centena = (auxiliar1/10000)+0x30;  
    decena = ((auxiliar1%10000)/1000)+0x30;  
    unidad = ((auxiliar1%1000)/100)+0x30;  
    decima = ((auxiliar1%100)/10)+0x30;  
    centecima = (auxiliar1%10)+0x30;  
  
    USART_Transmit(0x0A); //salto Linea  
    USART_Transmit(0x0D); //retroceso carro  
  
    USART_Transmit('P');  
    USART_Transmit('1');  
    USART_Transmit('=');  
    USART_Transmit(centena);  
    USART_Transmit(decena);  
    USART_Transmit(unidad);  
    USART_Transmit('.');  
    USART_Transmit(decima);  
    USART_Transmit(centecima);  
  
    centena = (auxiliar2/10000)+0x30;  
    decena = ((auxiliar2%10000)/1000)+0x30;
```

```
unidad = ((auxiliar2%1000)/100)+0x30;
decima = ((auxiliar2%100)/10)+0x30;
centecima = (auxiliar2%10)+0x30;

USART_Transmit(' ');
USART_Transmit('P');
USART_Transmit('2');
USART_Transmit('=');
USART_Transmit(centena);
USART_Transmit(decena);
USART_Transmit(unidad);
USART_Transmit('.');
USART_Transmit(decima);
USART_Transmit(centecima);

//USART_Transmit(0x0A); //salto Linea
USART_Transmit(0x0D); //retroceso carro
}
```