



**FACULTAD DE
INGENIERIA**



**UNIVERSIDAD NACIONAL
de MAR DEL PLATA**

“Sistema de alerta de obstáculos aéreos y rastreador GPS para personas no videntes”

Autor: Gonzalo Nicolás Lizarralde

Carrera: Ingeniería Electrónica

Matrícula: 11.035

Director: Dr. Ing. Jorge Castiñeira Moreira

Codirector: Ing. Miguel A. Revuelta

Año: 2014



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Índice:

I. RESUMEN:	3
II. INTRODUCCIÓN:	4
III. ANTEPROYECTO:	6
III.1 ELECCIÓN DEL SENSOR DE PROXIMIDAD:	7
<i>III.1.1 Sensor Óptico:</i>	7
<i>III.1.2 Sensor Ultrasónico:</i>	8
III.2 MÓDULO GPS Y GSM/GPRS:	9
III.3 MICROCONTROLADOR:	10
III.4 MÓDULO BLUETOOTH:	12
III.5 LENGUAJE DE PROGRAMACIÓN:	13
<i>III.5.1 Lenguaje de Bajo Nivel:</i>	14
III.5.1.a Lenguaje de Máquina:	14
III.5.1.b Lenguaje Ensamblador (Assembler):	14
<i>III.5.2 Lenguaje de Alto Nivel:</i>	15
III.6 ASPECTOS TEÓRICOS:	16
<i>III.6.1 Sensores Ultrasónicos:</i>	16
III.6.1.a Sensores Ultrasónicos para medición de Distancia.....	17
III.6.1.b Principales parámetros de los ultrasonidos para la medición de distancias.....	19
III.6.1.c Transmisión y Recepción de ultrasonidos para la medición de Distancia.....	20
III.6.1.d Conclusiones	20
<i>III.6.2 El Sistema de Posicionamiento Global</i>	22
III.6.2.a Descripción del sistema.....	22
III.6.2.b Las señales	23
III.6.2.c Cálculo de la posición geográfica	27

IV. PROYECTO:	29
IV.1 SENSOR DE PROXIMIDAD.....	29
IV.2 MÓDULO GPS Y GSM/GPRS	32
<i>IV.2.1 Comandos Utilizados</i>	33
IV.3 PROGRAMACIÓN DEL MICROCONTROLADOR:.....	36
IV.4 DISEÑO:.....	38
IV.5 PROTOTIPO:.....	44
IV.6 MEJORAS FUTURAS	47
V. CONCLUSIÓN:	48
VI. BIBLIOGRAFÍA:	50
VII. ANEXO	51

I. Resumen:

En el siguiente informe se presenta un proyecto final de la carrera de Ingeniería Electrónica, cuyos objetivos principales fueron dos. El primero consistió en diseñar y construir un sistema de asistencia para personas no videntes, capaz de detectar obstáculos que se encuentren a una altura media (con respecto de una persona adulta), para luego alertarla, ya sea por sonido o vibración. El segundo objetivo fijado fue detectar la posición geográfica de la persona utilizando los satélites de GPS, para luego dar aviso de dicha ubicación a un contacto de la persona invidente mediante un mensaje de texto. Este dispositivo se diseñó teniendo en cuenta que se utilizaría a la altura de la cintura o del pecho.

El dispositivo posee un sensor ultrasónico HC-SR04 capaz de detectar objetos hasta una distancia de 4 metros; pero en este proyecto se optó por limitar el rango a 1.5 metros, ya que detectar obstáculos más allá de esa distancia sería confuso para el usuario.

Se utilizó un módulo GSM/GPRS y GPS SIM908-C, capaz de recibir las señales transmitidas por los satélites y calcular la posición geográfica de la persona. El mismo módulo permite enviar mensajes de texto si se utiliza un Chip con línea celular habilitada, esta función se empleó para poder informar las coordenadas geográficas obtenidas con un link de Google Maps en un mensaje de texto a un contacto conocido de la persona no vidente. Dicho link permite ver la localización en un mapa.

Para realizar la comunicación entre el sensor y el módulo se utilizó el microcontrolador PIC18F26K22 de la firma Microchip, que posee las siguientes características: dos puertos COM por Hardware (necesarios para el sensor y módulo anteriormente mencionados), 7 timers (necesario para poder calcular la distancia a los obstáculos), memoria RAM de 3.896 bytes y un oscilador de 64 MHz. Siendo esta la unidad de control del dispositivo. Se programó en lenguaje C utilizando el entorno CCS de PIC-C.

Al final se le agregó un módulo Bluetooth HC-06, que permitió enlazar el dispositivo con una PC o Smart Phone. Este módulo otorgó la posibilidad de utilizar el GPS de un celular en lugar del SIM908-C para calcular la ubicación de la persona no vidente, haciendo más versátil al dispositivo.

Los objetivos detallados anteriormente se pudieron alcanzar satisfactoriamente, ya que se construyó una placa prototipo y se pudieron comprobar todas las funciones íntegramente.

II. Introducción:

La idea original de este proyecto consistía en realizar un bastón electrónico capaz de detectar obstáculos más allá de la punta del bastón, pero mi Co-director me recomendó conversar con la Directora de UMASDECA (Unión Marplatense de Acción Social por los Derechos del Ciego y Amblópe). Ella me comentó que no les era útil sentir más allá de la punta del bastón, ya que la distancia de sensado de 1.5 metros, aproximadamente, que le permite el bastón es suficiente para tomar una acción correctiva según el obstáculo, y que si le colocaba un motor vibrador en el bastón (como sistema de alerta), le quitaría sensibilidad para detectar las distintas superficies. Pero me aclaró que si tenían problemas detectando obstáculos aéreos, es decir, por encima de la cintura. Por lo tanto, decidí diseñar un dispositivo que detecte ese tipo de obstáculos, con la adición de un módulo GPS capaz de detectar la posición geográfica y que permitiera enviar un mensaje a un contacto de la persona no vidente, indicando donde se encuentra en caso de perderse. Este dispositivo fue pensado como un complemento al bastón blanco.

También se optó por incorporar un módulo bluetooth, para así tener la posibilidad de utilizar el GPS propio del celular (como el que utiliza la Directora de UMASDECA, que posee una aplicación que le recita el texto y los botones que aparecen en la pantalla) y poder obviar el módulo GPRS/GPS, haciendo más versátil al dispositivo.

Por lo tanto, se fijaron los siguientes objetivos:

- ✓ Analizar las características de los sensores de proximidad disponibles en el mercado y seleccionar uno que posea un rango adecuado junto con bajo consumo y bajo costo.
- ✓ Analizar las características de los módulos GSM/GPRS y GPS disponibles en el mercado y seleccionar uno de tamaño reducido, además de poseer bajo consumo y bajo costo.
- ✓ Aprender a programar un microcontrolador, para poder hacer interactuar los distintos módulos, y que el dispositivo trabaje de forma automática.
- ✓ Realizar pruebas independientes con cada módulo para aprender su funcionamiento y comprobar sus características.
- ✓ Diseñar un circuito para integrar todos los módulos de hardware, teniendo como núcleo el microcontrolador que realizará las tareas de cálculo y coordinación entre los diferentes módulos.
- ✓ Incorporar un módulo bluetooth para establecer una conexión con un teléfono celular.

Con la finalidad de llevar a cabo los objetivos establecidos, se realizaron diversas consultas a docentes de la Facultad de Ingeniería; se buscaron antecedentes de proyectos similares, tanto en referencias bibliográficas como a través de Internet. Solo se encontraron proyectos detallando distintos tipos de bastones electrónicos para no-videntes o descripciones y análisis sobre distintos tipos de sensores de proximidad.

Posteriormente, se buscó información específica sobre cada uno de los módulos mencionados anteriormente, con la finalidad de elegir el más adecuado para el problema a resolver. Por ejemplo, los sensores de proximidad utilizan diferentes tecnologías (ultrasonido, laser, efecto fotoeléctrico, etc.) por lo que se debió analizar cuál era el más adecuado para este caso en particular. A continuación se evaluaron los módulos GSM/GPRS y GPS, de los que se encontraron diferentes marcas como Globalsat y SIMCom que producen estos módulos con diferente nivel de integración. Y por último, el microcontrolador, de los cuales existe una amplia variedad en el mercado, con diversas funciones y capacidades, siendo Microchip, Freescale y Atmel dos de las marcas más reconocidas.

Durante esta investigación, siempre se tuvo presente que estos dispositivos fuesen de tamaño compacto, fáciles de conseguir en el mercado, posean bajo consumo y bajo costo.

III. Anteproyecto:

Una vez definido el sistema a diseñar y fijados los objetivos principales, se procedió a decidir la manera en la cual se implementaría la solución deseada, es decir, tomar las decisiones sobre el tipo de tecnología a utilizar, el lenguaje de programación con el cual se desarrollaría el proyecto, y los componentes que se utilizarían; considerando la confiabilidad, tamaño, costo, etc.

A continuación se presenta un diagrama en bloques del diseño propuesto, en el que se aprecian los diferentes elementos que conformaran el sistema.

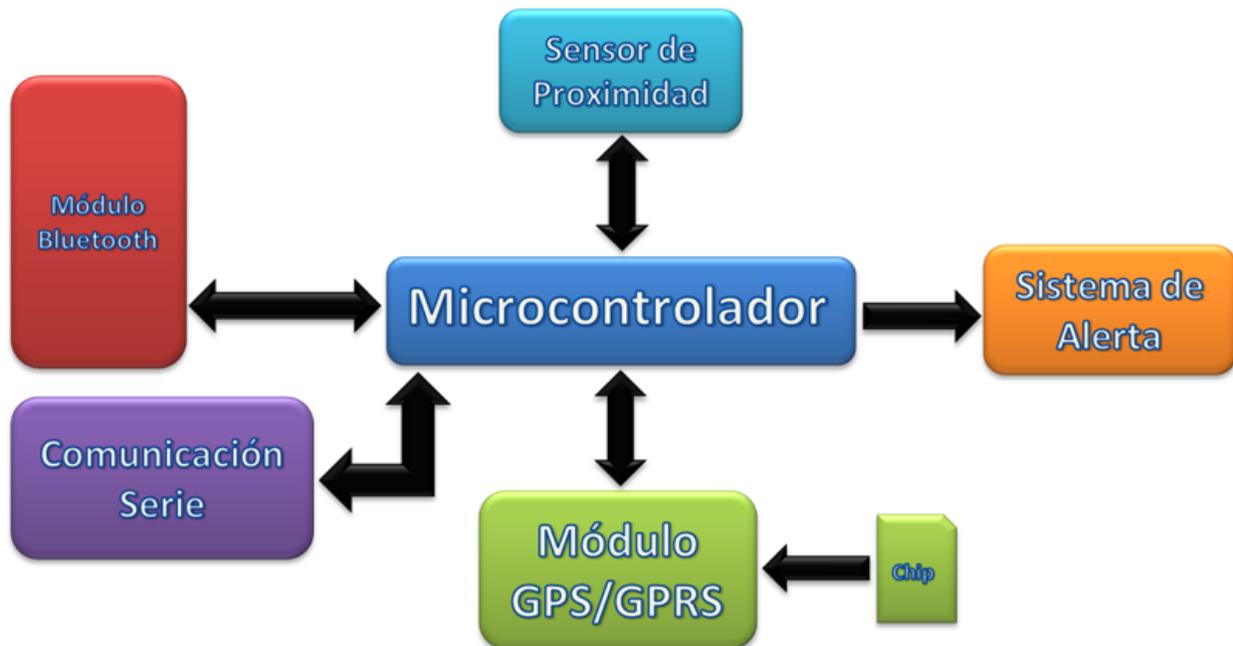


Figura III.1 – Diagrama de Bloques del Sistema completo a diseñar.

III.1 Elección del sensor de proximidad:

Las características que se eligieron para el sensor que midiera la proximidad y/o distancia a objetos ubicados a media altura, fueron las siguientes: rápido, tener buena sensibilidad, amplio rango, operar en distintos ambientes, tener bajo consumo; y considerando que el costo sea el menor posible.

Se estudiaron dos posibilidades que estaban disponibles, ampliamente, en el mercado: sensores ópticos o sensores ultrasónicos.

III.1.1 Sensor Óptico:

Los sensores de proximidad ópticos detectan la cercanía de un objeto por su influencia sobre una onda propagadora que se desplaza desde un transmisor hasta un receptor. Este tipo de sensor está constituido por un diodo emisor de luz de estado sólido (led), que actúa como un transmisor de luz infrarroja y un fotodiodo de estado sólido que actúa como el receptor.

Entre los disponibles en el mercado se evaluó el modelo: **GP2Y0A02YK0F**. Es un sensor medidor de distancia, compuesto por una combinación integrada de PSD (Position Sensitive Detector), IRED (Infrared Emitting Diode) y un circuito procesador de señal. La variación en la reflectividad del objeto, la temperatura del ambiente y la duración de la operación no influyen demasiado en la detección de distancia, debido al método de triangulación que utiliza este dispositivo. La tensión de salida de este dispositivo corresponde a la detección de la distancia, por lo tanto, este sensor puede ser usado como un sensor de proximidad. Para utilizar este sensor se debe realizar diferentes pruebas para determinar a qué distancia corresponde la tensión de salida, y si esta tiene un comportamiento lineal; de esa forma se podría calcular la distancia mediante un simple cálculo de regla de tres realizado por el microcontrolador.



Tensión	4.5 a 5.5 V
Consumo	33 mA
Distancia Mínima:	20 cm.
Distancia Máxima:	150 cm.
Tipo de Salida:	Analógica
Tamaño:	29.5 x 13 x 21.6 mm
Costo:	\$187

Figura III.2 – Sensor Óptico.

III.1.2 Sensor Ultrasónico:

Los sensores de ultrasonidos trabajan libres de roces mecánicos y pueden detectar objetos a distancias de hasta 8 metros. El sensor emite impulsos ultrasónicos, los cuales se reflejan en un objeto, luego el sensor recibe el eco producido y lo convierte en señales eléctricas. Estos sensores trabajan solamente en el aire, y pueden detectar objetos con diferentes formas, superficies y de diferentes materiales. Los materiales pueden ser sólidos, líquidos o polvorientos, sin embargo deben ser deflectores de sonido. Los sensores trabajan según el tiempo de transcurrido del eco, es decir, se evalúa la distancia temporal entre el pulso de emisión y el pulso del eco.

Estos dispositivos poseen el problema de zonas ciegas, comprendidas entre el lado sensible del detector y el alcance mínimo en el que ningún objeto puede ser detectado de forma fiable.

Entre los disponibles en el mercado se evaluó el modelo: **HC-SR04**. Es un dispositivo que utiliza el sonar (técnica que emplea la propagación del sonido para navegar y detectar objetos) para determinar la distancia a un objeto, como lo hacen los murciélagos o delfines. Ofrece una distancia de detección sin contacto excelente, con alta precisión y lecturas estables en un circuito fácil de usar, y con amplio rango. Su operación no es afectada por la luz solar (aunque los materiales acústicamente blandos como la tela pueden ser difíciles de detectar). Consta de emisor de ultrasonidos y módulo receptor. Este tipo de sensores se utilizan por lo general en robots, controlados por una placa como puede ser la Arduino, la cual posee como núcleo un microcontrolador.



Tensión	5V
Consumo	15 mA
Frecuencia:	40 Khz.
Distancia Mínima:	2 cm.
Distancia Máxima:	400 cm.
Sensibilidad:	Detecta un palo de escoba a 3 m.
Pulso de Disparo	10 uS min. TTL
Pulso de Eco:	100 uS – 18 mS
Retardo entre pulsos:	10 mS Mínimo
Tamaño:	45 x 20 x 15 mm
Peso:	10 gr.
Costo:	\$35

Figura III.3 – Sensor Ultrasónico.

III.2 Módulo GPS y GSM/GPRS:

Los módulos GSM y GPS tienen la función principal de detectar y comunicar la posición geográfica utilizando, a su elección, el sistema de referencia de antenas celulares o los satélites de GPS. Normalmente son de pequeño tamaño, lo cual permite diseñar dispositivos portátiles.

Entre los disponibles en el mercado se evaluó el modelo: **SIM908-C**, el cual está integrado con un motor de GSM / GPRS de alto rendimiento y un motor GPS. El motor de GSM / GPRS consiste en un módulo de cuatro bandas que funciona en las frecuencias de GSM 850MHz, 900MHz EGSM, DCS 1800MHz y 1900MHz PCS. La localización GPS ofrece la adquisición de las mejores en su clase y sensibilidad de rastreo, Time-To-First-Fix (TTFF) y precisión.

SIM908-C puede satisfacer casi todas las necesidades de espacio de las aplicaciones del usuario, tales como M2M, teléfonos inteligentes, PDA, y otros dispositivos móviles; ya que posee las siguientes dimensiones: 50mm x 33mm x 8.8mm, y un conector DIP de 60 pines, que proporciona todas las interfaces de hardware entre el módulo y las placas de los usuarios.

Características:

- Puerto Serie y puerto Debug que permiten a los diseñadores desarrollar fácilmente aplicaciones de usuario.
- GPS puerto serie.
- Dos canales de audio, que incluyen dos entradas de audio y dos salidas de audio. Éstos pueden ser configurados fácilmente mediante comandos AT.
- Interfaz de carga de batería.
- Entrada y Salida programable de uso general.
- El teclado como las interfaces de pantalla SPI que otorga a los usuarios la flexibilidad para desarrollar aplicaciones personalizadas.
- Interfaz del conector RF.

SIM908-C está diseñado con la técnica de ahorro de energía para que el consumo de corriente sea tan bajo como 1,0 mA en modo de reposo (motor de GPS está apagado) e integra el protocolo TCP / IP y ampliado TCP / IP, con comandos AT que son fáciles de usar y muy útiles para aplicaciones de transferencia de datos.



Figura III.4 – Módulo SIM908-C

III.3 Microcontrolador:

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), periféricos, puertos de entrada y salida. Estas partes se encuentran interconectadas dentro del microcontrolador, y en conjunto forman lo que se conoce como microcomputadora. Existen diferentes empresas dedicadas a la construcción de semiconductores, las cuales producen sus propios microcontroladores, lo que significa que el mercado es muy grande y variado. Existe la ventaja para los estudiantes de la universidad de solicitar muestras de forma gratuita a los distintos fabricantes, y así poder probar diferentes modelos.

Con respecto a las interfaces y periféricos se requiere un esfuerzo adicional para desarrollar los circuitos correspondientes, pero las opciones son ilimitadas, lo que genera mayor flexibilidad.

En este proyecto se optó por trabajar con microcontroladores llamados PIC (*Peripheral Interface Controller*), de la firma Microchip, debido a que se dispone de mucha información con respecto a estos en libros e internet, y su programación se puede realizar con diferentes lenguajes (software) de fácil acceso.

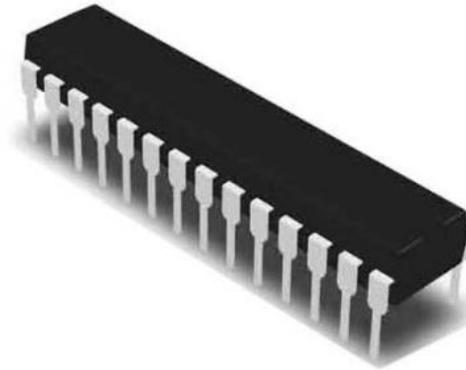


Figura III.5 – Microcontrolador de 28 pines

Microchip ofrece una gran cantidad de microcontroladores con diversas funciones. Debido a la cantidad y características de los periféricos a controlar se decidió trabajar con un PIC que posea la cantidad de entradas/salidas necesarias para este proyecto. Por eso se eligió un microcontrolador de 28 pines en formato DIP (*Dual in-line package*), el cual permite trabajar en una placa experimental y realizar su soldadura con herramientas convencionales, al pasar el diseño a plaqueta. Otra característica que se buscaba consistía en que posea pines de comunicación de EUSART (*Enhanced Universal Synchronous Asynchronous Receiver-Transmitter*) para poder programarlo directamente desde la PC.

Con respecto al microcontrolador se comenzó haciendo pruebas con el PIC18F2550, debido a que es con el cual se enseña a programar en la materia “Instrumentación Virtual”. Luego, de verificar que las rutinas funcionaban bien, se procedió a buscar un PIC que se adecuara mejor a las necesidades que se requerían.

Se terminó eligiendo el PIC18F26K22, el cual posee 2 puertos serie por hardware (uno para el sensor ultrasónico y otro para el módulo GPS/GPRS), mayor memoria, oscilador de mayor frecuencia y mayor cantidad de timers.

PIC18F2550

Parámetro	Valor
Program Memory Type	Flash
Program Memory (KB)	32
CPU Speed (MIPS)	12
RAM Bytes	2,048
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-UART, 1-A/E/USART, 1-SPI, 1-I2C1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	2 CCP
Timers	1 x 8-bit, 3 x 16-bit
ADC	10 ch, 10-bit
Comparators	2
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	2 to 5.5
Pin Count	28
Oscillator	48 MHz
USB (ch, speed, compliance)	1, Full Speed, USB 2.0

PIC18F26K22

Parámetro	Valor
Program Memory Type	Flash
Program Memory (KB)	64
CPU Speed (MIPS)	16
RAM Bytes	3,896
Data EEPROM (bytes)	1024
Digital Communication Peripherals	2-UART, 2-A/E/USART, 2-SPI, 2-I2C2-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	2 CCP, 3 ECCP
Timers	3 x 8-bit, 4 x 16-bit
ADC	17 ch, 10-bit
Comparators	2
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	28
Oscillator	64 MHz (using PLL)
XLP	Yes
Cap Touch Channels	17

Figura III.6 – Tabla comparativa de Microcontroladores PIC.

III.4 Módulo Bluetooth:

Bluetooth es una denominación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Existen distintos tipos de módulos como Maestro-Esclavo, lo cual significa que además de recibir conexiones desde una PC o Tablet (Modo Esclavo), y también es capaz de generar conexiones hacia otros dispositivos bluetooth. Esto nos permite por ejemplo, conectar dos módulos bluetooth y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos.

Entre los disponibles en el mercado se evaluó el siguiente: **HC-06**. Posee pines de transmisión (TX) y recepción (RX), por los cuales envía y recibe datos mediante un protocolo serie; con un nivel lógico TTL (igual que un microcontrolador). La computadora en su puerto serie mediante la norma RS232 maneja niveles de tensión de +/-10V, por lo que si conectamos directamente el módulo al puerto serie quedaría inservible. Se debe colocar en el medio un transeceptor, como el MAX232 el cual convierte

los niveles de tensión TTL en RS232 y de RS232 en TTL, de esta forma podemos comunicar el módulo con la PC.

Una de las ventajas principales del módulo HC-06, además de su pequeño tamaño y sus buenas características de transmisión y recepción que le brindan un alcance muy amplio (por tratarse de un sistema local Bluetooth), es el bajo consumo de corriente que posee tanto en funcionamiento, como en modo de espera.

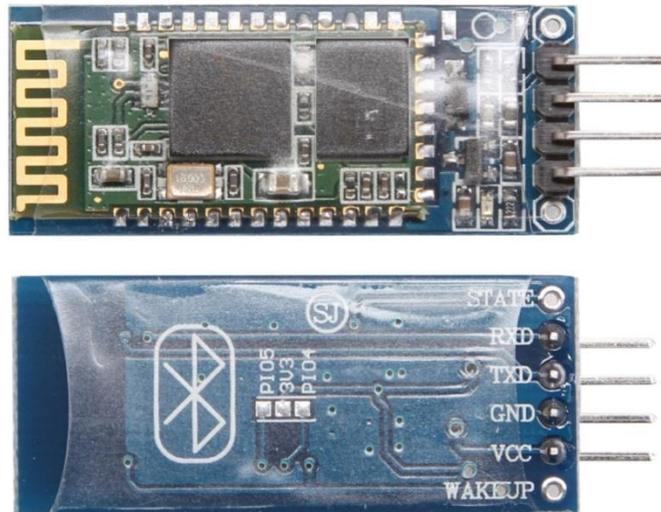


Figura III.7 – Módulo Bluetooth

III.5 Lenguaje de Programación:

En Informática, un “Lenguaje de Programación” es un lenguaje artificial y como tal posee un conjunto de símbolos, reglas sintácticas y semánticas. La sintaxis y la semántica definen su estructura y el significado de sus elementos, respectivamente. Con estos elementos de un lenguaje de programación se crea una secuencia de instrucciones (programa) para su procesamiento por una computadora. Un programa refleja un algoritmo a fin de resolver un problema específico.

Las rutinas para los microcontroladores PIC, se pueden programar en los siguientes lenguajes:

III.5.1 Lenguaje de Bajo Nivel:

Los lenguajes de bajo nivel, son aquellos que dependen intrínsecamente del ordenador. Es decir, estos programas informáticos son exclusivamente dependientes del hardware, no pueden migrarse, ya que están justamente diseñados, para un hardware específico.

Existen 2 tipos de lenguajes de bajo nivel: Lenguaje de Máquina y Lenguaje Ensamblador.

III.5.1.a Lenguaje de Máquina:

Es el Primer Lenguaje de Programación o Lenguaje de Primera Generación, y el único que comprende directamente la computadora. Utiliza el alfabeto binario formado por los símbolos 0 y 1 (apagado / encendido). Es realmente difícil de aprender y de manejar, ya que no puede ser escrito o leído usando un editor de texto, por lo tanto, no resulta comprensible para el ser humano. Pero es muy rápido, ya que se ahorra mucho tiempo en la respuesta, debido a que se evitan las traducciones intermedias.

Este lenguaje está compuesto por un conjunto de instrucciones que determinan acciones que serán ejecutadas por la máquina. Un programa consiste en una cadena de estas instrucciones de lenguaje de máquina (más los datos); las cuales son normalmente ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos. El lenguaje de máquina es específico de cada máquina o arquitectura de la máquina, aunque el conjunto de instrucciones disponibles pueda ser similar entre ellas.

Estas instrucciones son fáciles de interpretar por la computadora, pero difíciles para un programador, y viceversa. Para evitar la tediosa tarea de escribir programas en lenguaje máquina se han diseñado otros lenguajes de programación que facilitan la escritura y posterior ejecución de los programas.

III.5.1.b Lenguaje Ensamblador (Assembler):

Es un lenguaje de programación de bajo nivel para los ordenadores, microprocesadores, microcontroladores, y otros circuitos integrados programables. Implementa una representación simbólica

de los códigos de máquina binarios y otras constantes necesarias para programar una arquitectura dada de CPU y constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador. Esta representación es usualmente definida por el fabricante de hardware, y está basada en mnemónicos. Los mnemónicos son siglas que sustituyen a los códigos de operación (lenguaje de máquina) motivo por el cual resulta más fácil la programación. Además, el lenguaje ensamblador emplea instrucciones (que simbolizan los pasos de procesamiento), registros del procesador, las posiciones de memoria, y otras características del lenguaje. Un lenguaje ensamblador es por lo tanto, específico a cierta arquitectura de computadora física (o virtual). También es conocido como lenguaje de segunda generación.

III.5.2 Lenguaje de Alto Nivel:

Los lenguajes de alto nivel, son aquellos cuya característica principal, consiste en una estructura sintáctica y semántica legible, acorde a las capacidades cognitivas humanas. Se aproxima al lenguaje natural. A diferencia de los lenguajes de bajo nivel, son independientes de la arquitectura del hardware, motivo por el cual, asumen mayor portabilidad; pero se debe disponer de un programa traductor o compilador, que es suministrado por el fabricante, para obtener el programa ejecutable en lenguaje binario de la máquina que se trate. Además, no se necesita conocer el hardware específico de dicha máquina. Se aproxima al lenguaje natural, para que el programa se pueda escribir y leer de una forma más sencilla, eliminando muchas de las posibilidades de cometer errores que ocurrían en el lenguaje ensamblador, ya que se utilizan palabras en inglés (Por Ej: READ, WRITE, PRINT, OPEN, etc.) en lugar de cadenas de símbolos sin ningún significado aparente.

Se incluyen rutinas de uso frecuente, como las de entrada / salida, funciones matemáticas, manejo de tablas, etc., que figuran en una librería del lenguaje, de manera que se puedan utilizar siempre que se quiera sin necesidad de programarlas otra vez.

Pero este tipo de lenguaje también trae aparejado ciertas desventajas como un incremento del tiempo de depuración, al necesitarse diferentes traducciones del programa fuente para conseguir el programa definitivo; no se aprovechan los recursos internos de la máquina que se explotan mucho mejor en lenguajes de máquina y ensambladores, aumento de la ocupación de memoria; y por último, el tiempo de ejecución de los programas es mucho mayor.

III.6 Aspectos Teóricos:

III.6.1 Sensores Ultrasónicos:

Los Ultrasonidos son ondas acústicas o sonoras cuya frecuencia está por encima del límite perceptible del ser humano, aproximadamente 20 KHz. Los sensores ultrasónicos emplean el fenómeno de la piezoelectricidad, que ocurre en determinados cristales. Al ser sometidos a tensiones mecánicas, en su masa adquieren una polarización eléctrica que produce una diferencia de potencial y cargas eléctricas en su superficie. Este efecto es reciproco, es decir, al aplicar una diferencia de potencial al sensor, éste se deformará mecánicamente como respuesta.

El movimiento efectuado depende de la forma y orientación del cuerpo con relación a los ejes de los cristales y la posición de los electrodos. Los electrodos metálicos se recubren con otros metales para unirlos al material piezoelectrico y aplicarles o extraerles la carga eléctrica. Como los materiales piezoelectricos son aisladores, los electrodos se convierten en placas de un capacitor. Por lo tanto, un elemento piezoelectrico que se emplea para convertir movimiento en señales eléctricas, puede considerarse como generador de carga y de forma general modelarse como un capacitor. La deformación mecánica genera una carga, y ésta se convierte en un voltaje definido que aparece entre los electrodos de acuerdo con la ley general de los capacitores.

$$V_c = \frac{Q}{C}$$

El efecto piezoelectrico es sensible a la dirección, ya que la tensión produce una polaridad definida en el voltaje, mientras que la compresión produce una opuesta.

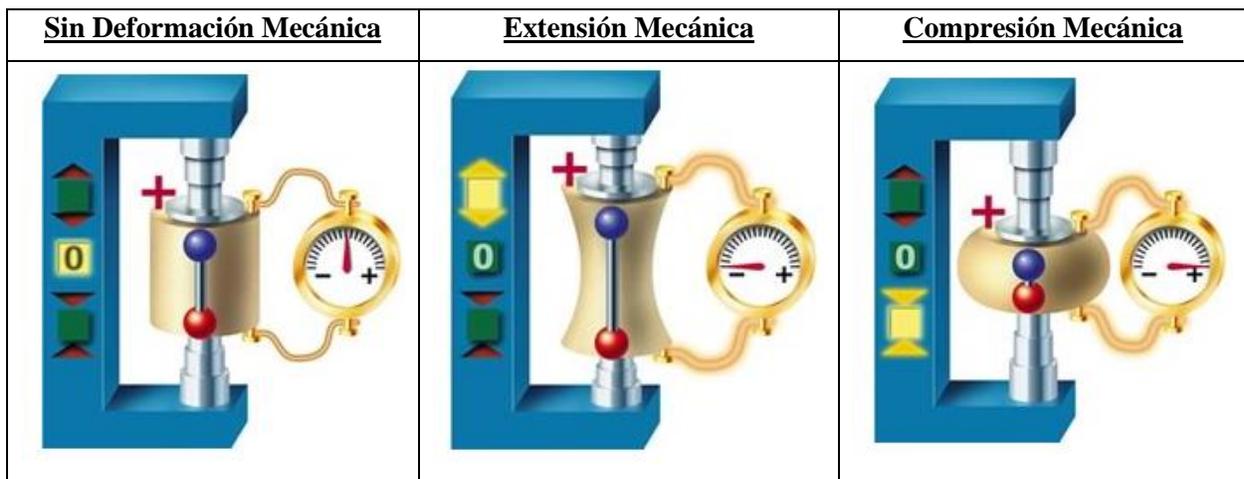


Figura III.8 – Deformaciones Mecánicas de un elemento Piezoelectrico.

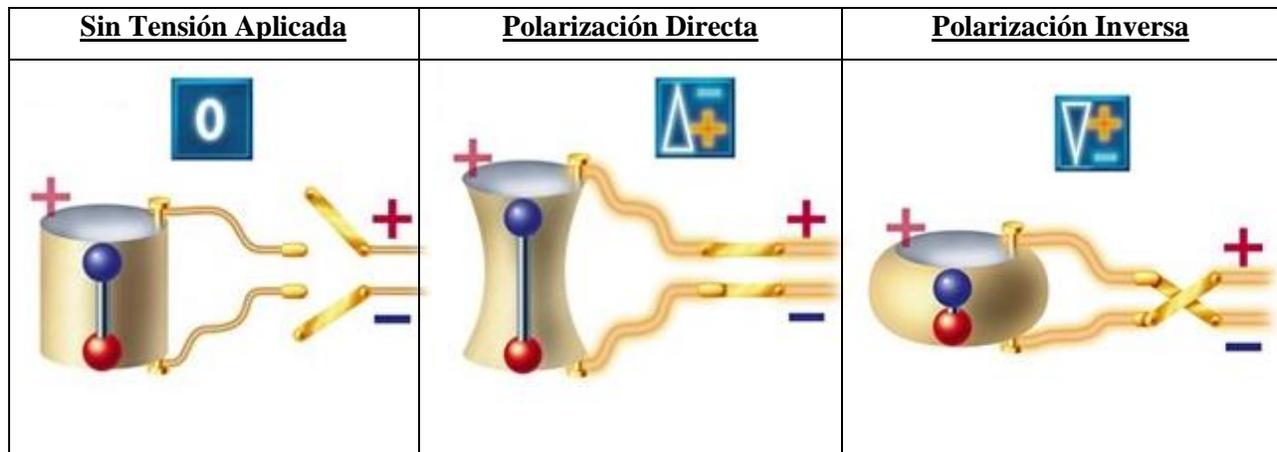


Figura III.9 – Polarización de un elemento Piezoeléctrico.

Si a un transductor piezoeléctrico, de un sensor ultrasónico, se le aplica en sus extremos (electrodos) un voltaje, el cristal sufre cambios en sus dimensiones, lo que ocasiona un cambio de presión en el medio que lo rodea (en este caso el aire), y viceversa, al ser sometido el cristal a un cambio de presión aparecen cargas eléctricas en sus extremos, donde se crea una diferencia de potencial. Por lo tanto, este tipo de transductor puede operar tanto como emisor o receptor ultrasónico.

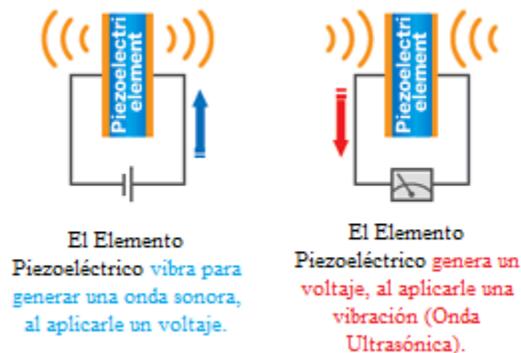


Figura III.10 – Generación de una onda ultrasónica debido al efecto piezoeléctrico.

III.6.1.a Sensores Ultrasónicos para medición de Distancia

El funcionamiento básico de los sensores ultrasónicos para la medición de distancia consiste en un Transmisor que emite un pulso de ultrasonido, el cual rebota sobre un determinado objeto y la reflexión de ese pulso es detectado por un Receptor de ultrasonidos.

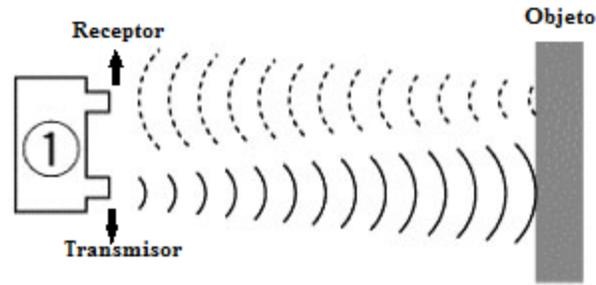


Figura III.11 – Esquema básico de un sensor de ultrasonido para medir distancia.

La mayoría de los sensores ultrasónicos de bajo costo se basan en la emisión de un pulso de ultrasonido cuyo lóbulo, o campo de acción, es de forma cónica.

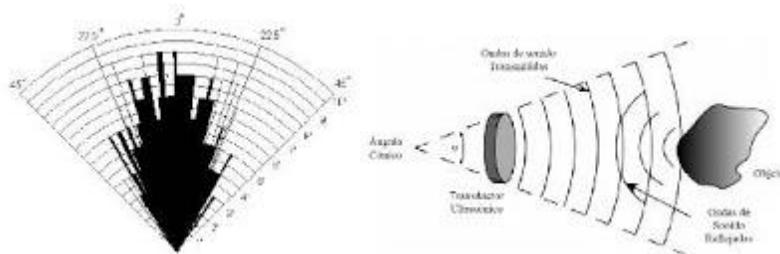


Figura III.12 –Perspectiva del ángulo de emisión del pulso ultrasónico.

Midiendo el tiempo que transcurre entre la emisión del sonido y la recepción del eco se puede establecer la distancia a la cual se encuentra el objeto que ha producido la reflexión de la onda sonora, mediante la fórmula:

$$Distancia = \frac{Velocidad \times Tiempo}{2}$$

Donde la Velocidad es la del sonido en el aire (343 m/s), medio de transmisión, el Tiempo es el transcurrido entre la emisión y recepción del pulso; y se divide por 2, ya que en ese tiempo se recorrió dos veces la distancia que se desea medir.

Otro material diferente al aire se toma como un objeto, incluyendo sólidos, líquidos y gases. Todos los objetos reflejan y absorben una porción de la onda, es decir, una parte de la onda que llega a la superficie del material es reflejada, mientras que otra parte de la onda penetra el material.

La amplitud de la onda reflejada es directamente proporcional a la superficie del objeto reflejante. El tamaño de la superficie, forma y orientación son también un factor contribuyente a la fuerza de la señal reflejada.

III.6.1.b Principales parámetros de los ultrasonidos para la medición de distancias

➤ Modo de operación

Modo Opuesto: Un sensor emite la onda de sonido y otro, montado en posición opuesta al emisor, recibe la onda.

Modo Difuso: El mismo sensor emite la onda de sonido y luego recibe el eco que rebota de un objeto.

➤ Margen de detección

El rango de detección es la distancia dentro de la cual el sensor ultrasónico detectará un objeto bajo fluctuaciones de temperatura y voltaje.

➤ Zona ciega

Los sensores ultrasónicos tienen una zona ciega inherente ubicada en la cara de detección. El tamaño de la zona ciega depende de la frecuencia del transductor. Los objetos ubicados dentro de la zona ciega no pueden ser detectados de manera confiable.

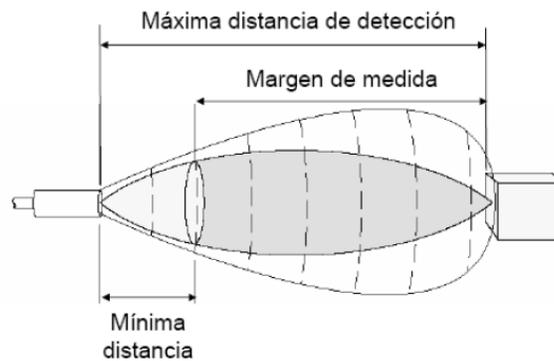


Figura III.13 – Distribución de márgenes de un sensor ultrasónico.

➤ Consideraciones sobre el objeto

Se deben tener en cuenta ciertas características de los objetos cuando se usan sensores ultrasónicos. Éstas incluyen la forma, el material, la temperatura, el tamaño y la posición del objeto. Los materiales suaves tales como telas o caucho esponjoso son difíciles de detectar por la tecnología ultrasónica difusa porque no reflejan el sonido adecuadamente.

III.6.1.c Transmisión y Recepción de ultrasonidos para la medición de Distancia

La distancia es calculada midiendo el tiempo transcurrido entre la emisión de la onda ultrasónica incidente y la reflejada, ya que se relacionan de forma proporcional. Por lo tanto, el obstáculo debería tener una sección transversal grande y no absorber el ultrasonido. El método más utilizado para la determinación de la distancia consiste en enviar trenes de pulsos de 40 KHz con periodos muy cortos y con la mayor amplitud posible, ya que la cantidad de energía que se transfiere al medio en forma de onda mecánica se maximiza; con lo cual aumenta la distancia de detección del sensor. Como los ecos deben recorrer mayor distancia, éstos serán recibidos por el receptor un periodo de tiempo posterior a las ondas directas, y no perturbarán a la hora de cuantificar la distancia.

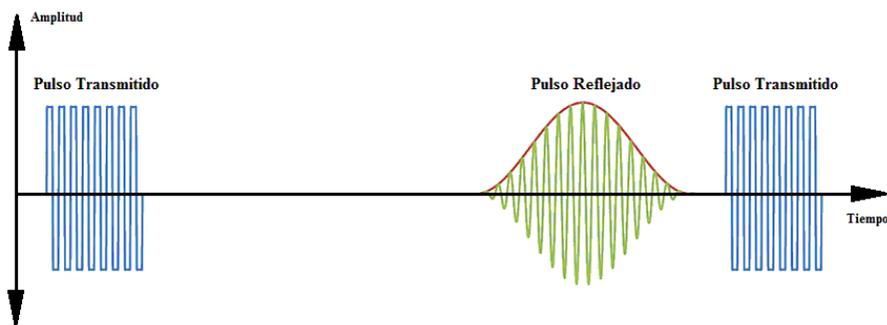


Figura III.14 – Diagrama de tiempo para la transmisión y recepción de ultrasonidos.

III.6.1.d Conclusiones

El sensor de ultrasonido posee un funcionamiento muy sencillo, sin embargo existen factores externos que influyen de manera determinante en las mediciones a realizar. Por lo tanto, es necesario considerar las diversas fuentes de interferencia que pueden afectar las mediciones para poder tratarlas de forma adecuada, minimizando su efecto. Entre los diversos factores que alteran las lecturas que se realizan con los sensores de ultrasonido cabe destacar:

- El campo de actuación del pulso que se emite desde un transductor de ultrasonido tiene forma cónica. El eco que se recibe como respuesta a la reflexión del sonido indica la presencia del objeto más cercano que se encuentra dentro del cono acústico y no especifica en ningún momento la localización angular del mismo. Aunque la máxima probabilidad es que el objeto detectado esté sobre el eje central del cono acústico, la probabilidad de que el eco se haya producido por un

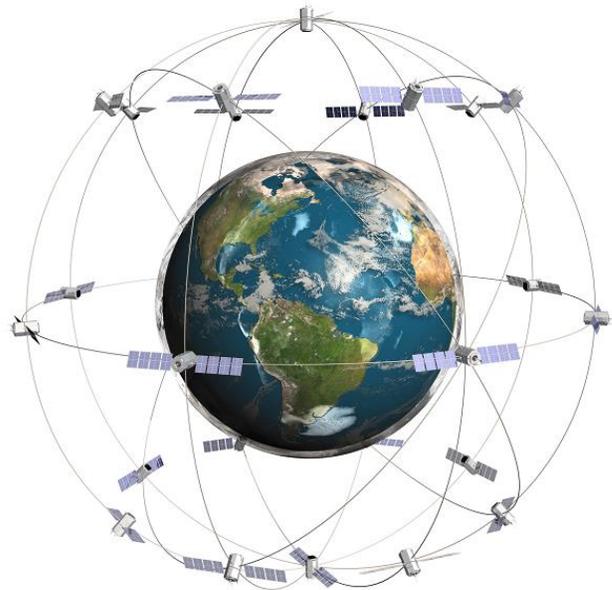
objeto presente en la periferia del eje central no es absolutamente despreciable, y debe ser tenido en cuenta.

- La cantidad de energía acústica reflejada por el obstáculo depende en gran medida de la estructura de su superficie. Para obtener una reflexión altamente difusa del obstáculo, el tamaño de las irregularidades sobre la superficie reflectora debe ser comparable a la longitud de onda de la onda de ultrasonido incidente.
- En los sensores de ultrasonido de bajo costo se utiliza el mismo transductor como emisor y receptor. Tras la emisión de ultrasonido se espera un determinado tiempo a que las vibraciones en el sensor desaparezcan y esté preparado para recibir el eco producido por el obstáculo. Esto implica que existe una distancia mínima d (proporcional al tiempo de relajación del transductor) a partir de la cual el sensor mide con precisión. Por lo general, todos los objetos que se encuentren por debajo de esta distancia d , serán interpretados por el sistema como que se encuentran a una distancia igual a la distancia mínima.
- La temperatura afecta la medida, ya que la velocidad de propagación de la onda varía dependiendo de la temperatura.
- Un factor de error muy común es el conocido como falsos ecos, estos pueden ocurrir cuando la onda choca con varios objetos antes de llegar al receptor.
- Las ondas de ultrasonido obedecen a las leyes de reflexión de las ondas, por lo que una onda de ultrasonido tiene el mismo ángulo de incidencia y reflexión respecto a la normal de la superficie. Esto implica que si la orientación relativa de la superficie reflectora con respecto al eje del sensor de ultrasonido es mayor que un cierto umbral, el sensor nunca recibirá el pulso de sonido que emitió.

III.6.2 El Sistema de Posicionamiento Global

III.6.2.a Descripción del sistema

La definición oficial del sistema GPS es: El Navstar Global Positioning System (GPS), donde GPS es el sistema y NAVSTAR es la constelación. Consiste en un sistema de navegación desarrollado por el Departamento de Defensa de los Estados Unidos que permite a las fuerzas armadas determinar de forma instantánea la posición, velocidad y hora de un objeto situado en las proximidades de la Tierra, en un sistema de referencia único y bajo cualquier condición meteorológica. Este sistema también es utilizado por civiles, pero con menor precisión.



El GPS establece para el sistema completo un total de 21 satélites en órbita, más 3 de repuesto.

Este conjunto de 24 satélites se distribuyen en seis planos orbitales con una inclinación de 55° respecto al plano ecuatorial terrestre, y están uniformemente repartidos a lo largo del ecuador, formando entre sí ángulos de 60° . Esta distribución garantiza que cualquier punto de la Tierra reciba en todo momento la señal de entre 4 y 12 satélites.

Cada satélite dispone de al menos cuatro osciladores atómicos encargados de generar la frecuencia fundamental de 10.23 MHz. El conjunto de los osciladores de todos los satélites más los osciladores del segmento de control en la Tierra, conforman la base de tiempos del sistema GPS.

El sistema GPS es mantenido por el segmento de control operacional (Operational Control Segment). Este está constituido por la estación central de control (Master Control Station) en Colorado, tres estaciones supervisoras y antenas terrestres. Las estaciones de seguimiento disponen de coordenadas muy precisas y están regularmente espaciadas en longitud.

Las estaciones reciben las señales de todos los satélites, con información respecto de los mismos. Esos datos contienen el Almanaque y la tabla de Efemérides. El Almanaque informa el estado de todos

los satélites de la red (nivel de salud, etc.), esta información suele variar aproximadamente cada 4 meses. La información del almanaque permite predecir la órbita y la posición de todos los satélites de forma aproximada. Cada satélite envía de forma individual los datos correspondientes al almanaque de todos los demás satélites. La tabla de Efemérides indica la posición precisa de los mismos. Los parámetros orbitales de los satélites se van actualizando a medida que su movimiento se ve alterado por la atracción del Sol y la Luna, la diferencia de gravedad entre distintas zonas de la corteza terrestre, viento solar, etc. Esta información de las efemérides normalmente cambia cada 4 horas. A diferencia del almanaque, en este caso cada satélite envía únicamente los datos propios de sus efemérides.

En base a estos datos la estación central de control hace una previsión de las efemérides y del comportamiento de los osciladores; y determina los parámetros que ha de contener el mensaje de navegación. Los datos del mensaje son enviados a las antenas terrestres y desde allí se transmiten a los satélites por la banda S. Esta banda comprende el rango frecuencial que va desde 1,5 a 5,2 GHz, cruzando el límite imaginario entre el UHF y el SHF a 3,0 GHz; y es parte de la banda de microondas del espectro electromagnético.

III.6.2.b Las señales

Las señales transmitidas por los satélites de GPS son complejas, ya que tienen que servir a un número ilimitado de usuarios, tanto militares como civiles, proporcionar medidas de distancia al satélite en tiempo real, la velocidad de desplazamiento en forma instantánea, medidas de fase precisas, mensaje en tiempo real, un modo de corregir el retardo ionosférico, permitir medidas simultáneas a varios satélites, proteger contra interferencias y tener tolerancia frente al efecto multitrayectoria. El efecto Multitrayectoria es un fenómeno de interferencia causado por señales GPS reflejadas en estructuras o superficies reflectoras las cuales, habiendo recorrido mayor distancia que la correcta, inducen errores de posición. Las características de las señales de los satélites GPS han sido diseñadas para reunir todos estos requisitos sin que el usuario tenga la necesidad alguna de transmitir.

III.6.2.b.1 Las ondas portadoras

Toda la información emitida por los satélites GPS es modulada sobre dos ondas portadoras de la banda L (comprende todas las ondas electromagnéticas cuya frecuencia está situada entre 1 GHz y 2 GHz) conocidas como L_1 y L_2 , cuya frecuencia se genera como múltiplo de la frecuencia fundamental.

$$\checkmark L_1 = 10,23 \text{ MHz} \times 154 = 1575,42 \text{ MHz}$$

$$\checkmark L_2 = 10,23 \text{ MHz} \times 120 = 1227,60 \text{ MHz}$$

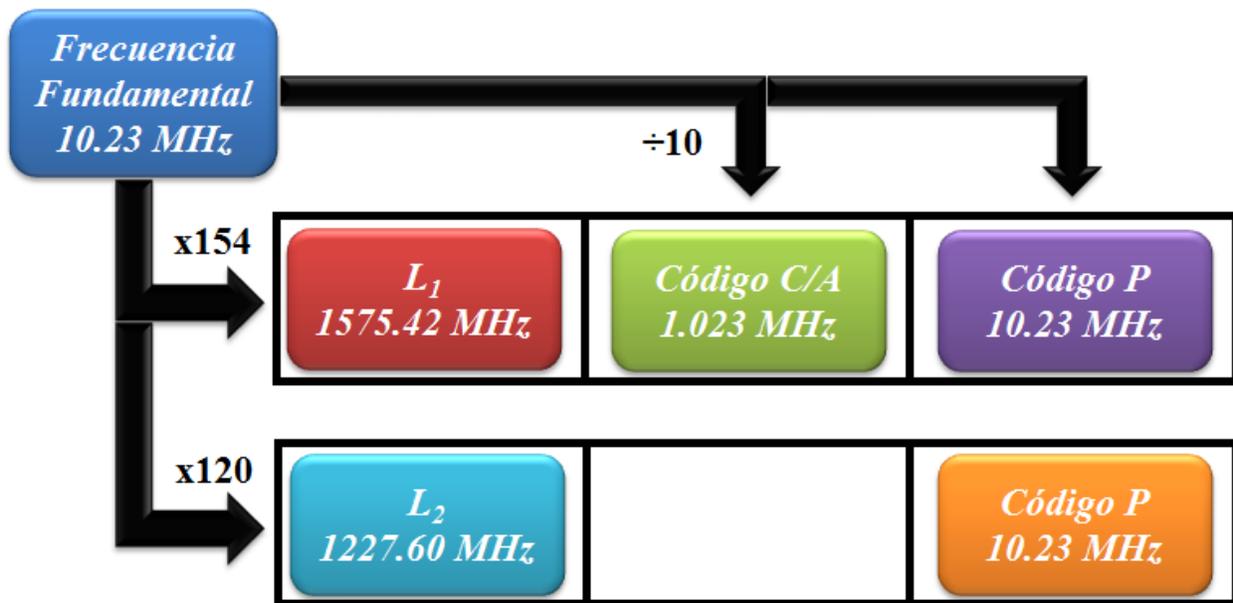


Figura III.15 – Estructura de la señal.

Se escogieron esas dos frecuencias por diversos motivos:

- Para poder transmitir toda la información necesaria sin pérdidas significativas se necesita un ancho de banda de 20 MHz, por lo cual se requiere una zona del espectro radioeléctrico que no se encuentre utilizada de forma desordenada.
- Para medir velocidades instantáneas en tiempo real con una precisión de 1 cm/s mediante la medición de desplazamientos en la frecuencia por efecto Doppler, se necesitan longitudes de onda de pocos decímetros.
- Cuanto más alta es la frecuencia, menor es la influencia del retardo ionosférico. Aun así, para la L_1 puede suponer hasta 30 m en el cenit (intersección de la vertical de un lugar y la esfera

celeste). La forma de eliminar esta influencia de la ionosfera es medir simultáneamente sobre otra portadora convenientemente desplazada de la anterior en el espectro radioeléctrico. De esta forma, mediante el empleo de una segunda portadora L_2 es posible eliminar la mayor parte de la influencia del retardo ionosférico.

- No es conveniente utilizar una frecuencia demasiado alta, porque a mayor frecuencia, mayor es la pérdida de señal para una misma frecuencia de transmisión.

III.6.2.b.2 Los códigos

Si los satélites transmitieran solamente las ondas portadoras L_1 y L_2 , los receptores no serían capaces de distinguir de qué satélite provienen, ni podrían determinar la distancia al satélite. Para solucionar este problema las señales deben ser moduladas, y para esto se emplean códigos pseudoaleatorios o códigos PRN (Pseudo Random Noise).

Los códigos PRN consisten en una secuencia de valores binarios que son generados por unos dispositivos electrónicos, esta cadena de dígitos es aparentemente aleatoria, pero en realidad se repite de forma periódica. Al ser conocida, puede ser reproducida por los receptores. A cada satélite se le asigna un código diferente; por lo tanto, correlacionando la señal, el receptor puede identificar al satélite y medir el tiempo que tarda la señal en recorrer la distancia existente entre el satélite y el receptor (tiempo de tránsito). Multiplicando el tiempo de tránsito por la velocidad de propagación se obtiene una estimación de la distancia al satélite.

Se transmiten dos tipos de códigos PRN:

- El código C/A (Clear/Acquisition). Es una secuencia de 1023 dígitos binarios transmitidos a 1.023 MHz. Por lo tanto, se repite cada milisegundo. Cada dígito supone 1 μ s y por lo tanto, a la velocidad de propagación de la señal supone aproximadamente 300 m, lo que constituye la longitud de onda del código C/A. Hay un total de 32 códigos C/A diferentes que se caracterizan por su baja correlación. Cada código es asignado a un único satélite. Como el código C/A se repite cada milisegundo, los receptores pueden correlacionar e identificar la señal rápidamente. La correlación consiste en adelantar o retrasar el código generado por el receptor de forma que coincida exactamente en el tiempo con el código que llega de la señal del satélite. Esta situación, una vez establecida, se mantiene a lo largo de la medición. La medición del retardo o adelanto

introducido en la réplica generada por el receptor permite determinar el tiempo de tránsito. El código C/A se transmite modulado sobre la portadora L_1 solamente.

- El código P (Precision). Su objetivo es mejorar la estimación del tiempo de tránsito llevada a cabo por el código C/A. El código P es una secuencia de 2.35×10^{14} dígitos binarios que transmitido a la frecuencia fundamental 10.23 MHz supondría casi 266 días. En consecuencia, el código completo se divide en tramos de una semana, y son asignados a un único satélite. Cada satélite comienza a transmitir su fracción de código P a las 0 horas de cada domingo (Tiempo Universal) y concluye a las 12 horas del sábado siguiente, comenzando de nuevo el proceso. Al ser transmitido a 10.23 MHz. cada dígito del código P supone aproximadamente 30 m. El código P se transmite modulado sobre las portadoras L_1 y L_2 . En la actualidad el código P se transmite encriptado mediante el proceso denominado Anti-Spoofing (AS) y por lo tanto, no puede ser directamente correlacionado. A pesar de ello, existen técnicas (Correlación cruzada, seguimiento Z, etc) que permiten a la mayor parte de los receptores modernos medir pseudodistancias precisas.

III.6.2.b.3 El mensaje

Para que el receptor pueda obtener la posición a partir de las pseudodistancias medidas mediante la correlación de los códigos, se necesita cierta información auxiliar como, por ejemplo, las coordenadas de los satélites y el error de los osciladores. Para que esto pueda llevarse a cabo en tiempo real se transmite el mensaje; que se modula en ambas portadoras y es superpuesto sobre los códigos a una frecuencia de 50 baudios.

Cada satélite transmite su propio mensaje que contiene fundamentalmente:

- Efemérides: consiste en el conjunto de parámetros que permiten determinar las coordenadas del satélite y el error del oscilador.
- Almanaque: consiste en unas efemérides aproximadas de todos los satélites de la constelación. Su misión es informar a los receptores de la situación aproximada de los satélites.
- Salud del satélite (HEALTH).
- Un indicador (HOW) para permitir correlacionar rápidamente el código P.

- Precisión estimada para la medida de la distancia (URA).
- Etc.

El mensaje completo consiste en 25 celdas de 1500 bits cada una. Como se transmite a 50 bits por segundo el tiempo necesario para transmitir el mensaje completo es de 12.5 minutos. Para permitir que el receptor determine rápidamente su situación las efemérides del satélite se transmiten cada 30 segundos.

III.6.2.c Cálculo de la posición geográfica

Para poder realizar el cálculo de la posición geográfica es necesario conocer en qué instante partió la señal del satélite, en qué instante llega al receptor y la velocidad de propagación de la señal. Para esto se necesitan 4 satélites, ya que si se sincroniza con un solo satélite, éste ubicará la posición sobre una esfera centrada en dicho satélite y de radio la distancia del receptor al satélite. Si se sincroniza con dos, como cada uno de ellos ubicará la posición del receptor en una esfera (cada una centrada en el satélite correspondiente), se puede deducir que la posición buscada se encuentra en la zona común de las 2 esferas. La intersección de dos esferas, en estas condiciones, es una circunferencia. Si se sincroniza con un tercer satélite, este ubicará la posición del receptor en una tercera esfera (centrada en él y de radio la distancia al mismo). Por lo tanto, la intersección de tres esferas en estas condiciones se reduce a dos puntos.

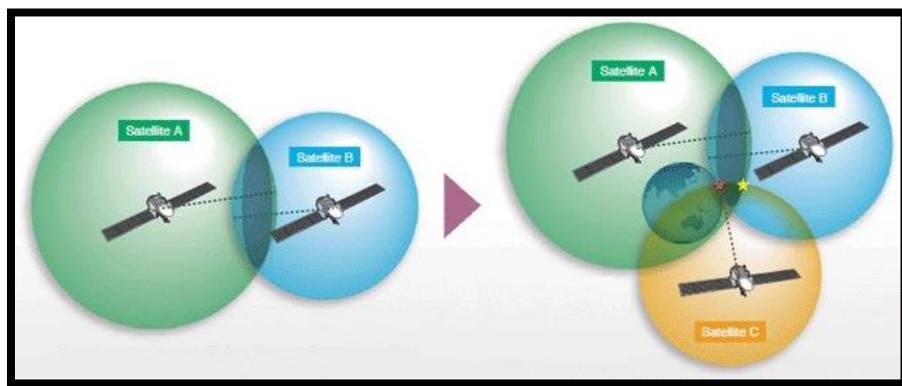


Figura III.16 – Intersección de las Esferas de los satélites.

Para decidir cuál de ellos es la posición verdadera, se podría efectuar una nueva medición con un cuarto satélite. Pero normalmente uno de los dos puntos posibles resulta ser muy improbable por su ubicación demasiado lejana de la superficie terrestre y puede ser descartado sin necesidad de mediciones posteriores.

Si el receptor tuviese un reloj perfectamente sincronizado con los relojes ubicados en los satélites, bastarían tres medidas para determinar las tres coordenadas de la posición. No obstante, el costo de un reloj de precisión es muy elevado, por lo cual se emplea un cuarto satélite para determinar el error en el reloj del receptor.

El receptor mide el tiempo que tarda en llegar la señal GPS de cada uno de los cuatro satélites, y multiplicando por la velocidad de la luz c , obtiene las medidas conocidas como PR_1, PR_2, PR_3, PR_4 . Teniendo en cuenta que se adquirieron las distancias a cada satélite $(X_i, Y_i, Z_i; i=1..4)$, con lo cual se llega a un sistema de ecuaciones con cuatro incógnitas:

$$(X_i - U_x)^2 + (Y_i - U_y)^2 + (Z_i - U_z)^2 = (PR_i - B * c)^2 ; i = 1..4$$

Donde U_x, U_y, U_z son las coordenadas buscadas y B es la corrección del error en el reloj del receptor.

IV. Proyecto:

Una vez investigados los diferentes sensores, módulos, microcontroladores, que se encuentran disponibles en el mercado; seleccionado los distintos dispositivos a utilizar; y estudiado los diferentes aspectos teóricos relacionados con dichos dispositivos, se procedió a realizar ensayos prácticos para comprender su funcionamiento y verificar las características descritas en las hojas de datos correspondientes.

IV.1 Sensor de Proximidad

En primer lugar, se evaluó el funcionamiento el sensor ultrasónico elegido. Se conectaron los pines “Trigger” y “Echo” a los pines de salida y entrada correspondientes del microcontrolador (PIC18F2550), utilizado en la materia optativa “Instrumentación Virtual”. Para esto se debieron consultar las hojas de datos tanto del sensor, como del microcontrolador.

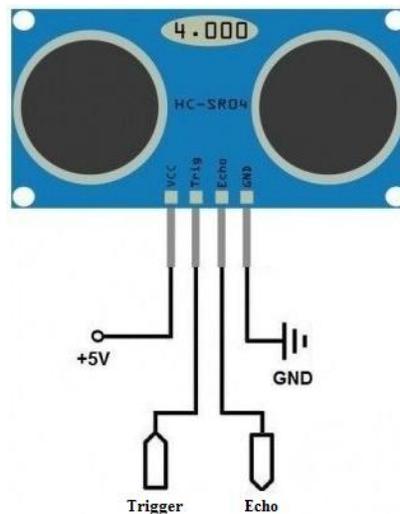


Figura IV.1 – Pines del Sensor Ultrasónico.

No se pudieron realizar simulaciones en programas como Proteus, debido a que no se encontraron las librerías necesarias. Por lo tanto, se procedió a construir el circuito directamente sobre una placa de desarrollo (Protoboard).

Para poder medir la distancia a un objeto, se debe medir el tiempo que transcurre desde que se genera el pulso de disparo hasta que se recibe el pulso de Eco. Al ser la señal de emisión y recepción pulsos, estos se pueden generar o recibir a través de los pines I/O (Entrada/Salida Digital) del microcontrolador. Los microcontroladores PIC poseen funciones conocidas como Interrupciones, que se pueden utilizar para detectar el pulso de Eco.

Las interrupciones son esencialmente llamadas a subrutinas generadas por los dispositivos físicos, en este caso el sensor ultrasónico, al contrario de las subrutinas normales de un programa en ejecución. Como el salto de subrutina no es parte de la secuencia de ejecución programada, el microcontrolador guarda el estado del procesador en una pila de memoria y ejecuta un código especial llamado "manejador de interrupciones" que atiende al periférico específico que generó la interrupción. Al terminar esa rutina, una instrucción especial le indica al procesador el fin de la atención de la interrupción. En ese momento, el microcontrolador restablece el estado anterior, y el programa que se estaba ejecutando antes de la interrupción, sigue su ejecución normal como si la interrupción no hubiese ocurrido.

Para poder medir distancias con este sensor, se debe enviar un pulso de disparo de al menos 10 μ seg, a través del pin Trigger; esto genera que el sensor envíe una ráfaga de 8 impulsos ultrasónicos; en ese preciso instante a través del pin Echo se genera un flanco positivo, y se mantendrá positivo hasta que se reciba una señal de eco o hasta que se cumplan los 36 mseg que corresponde al rango máximo de distancia del sensor, generando un flanco negativo. La duración del pulso de Eco corresponde al tiempo que tardaron las ondas en llegar al objeto y volver al sensor.

Por lo tanto, la distancia se puede calcular mediante la siguiente fórmula:

$$Distancia = \frac{Velocidad \times Tiempo}{2}$$

Donde la Velocidad es la del sonido en el aire (343 m/s), medio de transmisión, el Tiempo es el transcurrido entre la emisión y recepción del pulso; y se divide por 2, ya que en ese tiempo se recorrió 2 veces la distancia que se desea medir.

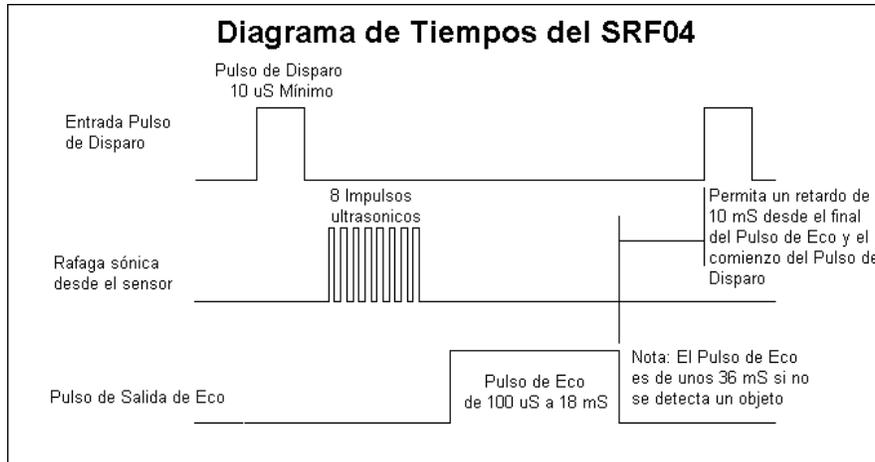


Figura IV.2 – Diagrama de tiempos del Sensor Ultrasónico.

Se eligió utilizar un rango de 25 cm a 150 cm. Los primeros 25 cm conforman una “zona muerta”, para el caso en que el usuario cruce el brazo por delante del dispositivo, y este no lo alerte. Y el límite máximo de 150 cm, debido a que a esa distancia el usuario tiene tiempo suficiente para corregir su trayectoria; detectar obstáculos más allá de esa distancia sería fastidioso para el usuario.

A continuación se puede observar el diámetro del área de la base del cono en la cual el sensor ultrasónico puede detectar obstáculos, siendo más precisa la zona verde que la amarilla.

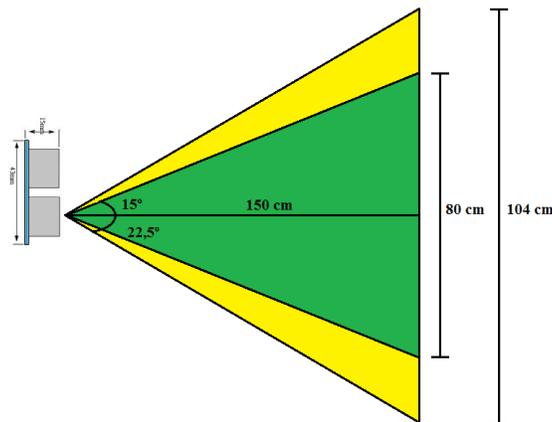


Figura IV.3 – Gráfica de Rangos del HC-SR04.

IV.2 Módulo GPS y GSM/GPRS

El SIM908-C es un módulo GSM/GPRS Quad-Band completo que trabaja con tecnología por satélite GPS. Posee un diseño compacto que integra GPRS y GPS en un solo componente, lo cual permite ahorrar significativamente tiempo y costo para desarrollar aplicaciones GPS.

Existe una placa de desarrollo de este módulo que permite utilizar todas las funciones disponibles del módulo de una manera sencilla, conectando la placa a una PC mediante puerto serie. De esta forma se puede configurar el módulo con las características requeridas, probar los comandos AT, y así poder depurar el programa final. Mi Co-Director fue muy amable en prestarme su placa de desarrollo para poder realizar dichas pruebas.



Figura IV.4 – Placa de desarrollo del SIM908.

Utilizando el puerto serie GPRS de la placa de desarrollo conectado a una PC se pudieron probar los diferentes comandos AT; empleando el programa Docklight. Estos comandos permiten configurar ciertas características del módulo, enviar y recibir mensaje, realizar llamadas, obtener las coordenadas geográficas, etc.

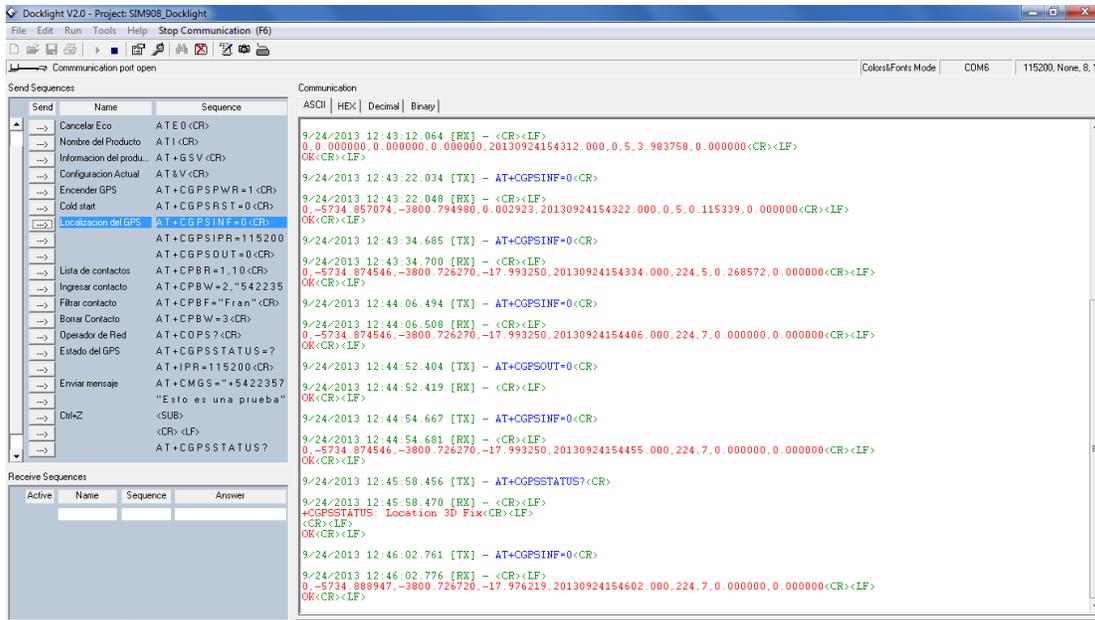


Figura IV.5 – Programa Docklight.

Como la idea del proyecto consistió en obtener las coordenadas geográficas, en las que se encuentra la persona en un preciso momento y luego enviar un mensaje de texto; se abocó la búsqueda de esos comandos específicos, aunque el módulo permite programar muchas más funciones.

IV.2.1 Comandos Utilizados

Configuración del GPS

- ✓ AT+CGSPWR=1: Control de encendido del GPS (0: Apagar, 1: Encender)
- ✓ AT+CGPSRST=0: Reseteo del GPS (0: Encendido en frio, 1: Encendido en modo autónomo)
- ✓ AT+CGPSIPR=9600: Setea la velocidad de Baud Rate (4800, 9600, 19200, etc).

Encendido en Modo Autónomo: Es la situación de arranque de un GPS que no tiene almacenado ningún almanaque, ni efemérides, por lo que se debe descargar esta información de los satélites. Normalmente el tiempo a cielo abierto y despejado suele ser superior a los 90 segundos.

Encendido en Frio (Cold Start): Es la situación de arranque de un GPS con un almanaque válido almacenado, por lo cual sólo debe descargar la información de los satélites. Los satélites envían la información de las efemérides a una velocidad de 50 bps, por lo tanto, el tiempo de adquisición suele ser de unos 45 segundos; en función del punto en el que se encuentre la transmisión de las tramas de información en el momento del arranque del GPS.

Encendido Cálido (Warm Start): Es la situación de arranque de un GPS con un almanaque y tablas de efemérides válidas almacenadas. En este caso sólo se necesita obtener una referencia de tiempo y sincronización con la transmisión de los satélites. El tiempo de adquisición suele ser entre 7 y 15 segundos.

Encendido en Caliente (Hot Start): Es la situación de arranque de un GPS con un almanaque y tablas de efemérides válidas almacenadas y una referencia de tiempo precisa. Esta situación suele darse en ciudades cuando “se pierde cobertura GPS” o cuando se atraviesa un túnel con un vehículo. El tiempo de adquisición suele ser entre 3 y 5 segundos.

Estado del GPS

- ✓ AT+CGPSSTATUS?: Pregunta el estado del GPS; existen 4 respuestas posibles.
 - “Location Unknown”: GPS no se encuentra operativo.
 - “Location Not Fix”: GPS se encuentra operativo, pero la posición no se ha fijado.
 - “Location 2D Fix”: La posición ha sido fijada en 2D (Coordenadas del Usuario).
 - “Location 3D Fix”: La posición ha sido fijada en 3D (Coordenadas y Altura del Usuario).

Envío de Mensaje de Texto

- ✓ AT+CMGF=1: Configura el sistema SMS en modo texto.
- ✓ La siguiente secuencia envía el “Texto” al número ingresado en el paso inicial.
 - AT+CMGS="\+542235730541\"
 - “Texto”
 - Ctrl+Z

- ✓ ATA=2: Contesta llamada.
- ✓ AT+CGPSINF=2: Obtiene la información de la localización actual del GPS (2: \$GPGGA).

Protocolo NMEA: (National Marine Electronics Association), permite enviar datos de posición entre los equipos electrónicos de la Marina. Los receptores de GPS han sido definidos con este protocolo de transmisión sencillo. Todas las sentencias NMEA empiezan con los caracteres \$GP.

Ej: \$GPGGA,163308.000,4019.5475,N,00347.4121,W,1,9,.086,686.3,M,51.7,M,,*46

- ❖ GGA → Global Positioning System Fix Data
- ❖ 163308.000 → Hora UTC en la cual se obtuvieron las coordenadas 16:33:08.000
- ❖ 4019.5475 → Latitud 40° 19.5475'
- ❖ N → Indicador N/S
- ❖ 00347.4121,W → Longitud 3° 47.4121'
- ❖ W → Indicador E/W
- ❖ 1 → Indicador de posición fija:

Position Fix Indicator	
0	Invalid
1	GPS fix (SPS)
2	DGPS fix
3	PPS fix
4	Real Time Kinematic
5	Float RTK
6	Estimated (dead reckoning) (2.3 feature)
7	Manual input mode
8	Simulation mode

- ❖ 9 → Número de satélites utilizados.
- ❖ 0.86 → Dilución horizontal de la posición.
- ❖ 686.3,M → Altitud sobre el nivel del mar, en metros.
- ❖ 51.7,M → Altura del geoide (nivel medio del mar), en metros, sobre WGS84 (World Geodetic System 84: Consiste en un patrón matemático de tres dimensiones que representa la tierra por medio de un elipsoide, un cuerpo geométrico más regular que la Tierra).
- ❖ *46 → Checksum, siempre comienza con *.
- ❖ <CR><LF> → Final de terminación mensaje.

Para que el conector de la tarjeta SIM funcionase, debió ser soldado al módulo ya que no venía integrado, se debió realizar los puentes de soldadura indicados en rojo que se pueden observar en la siguiente imagen.

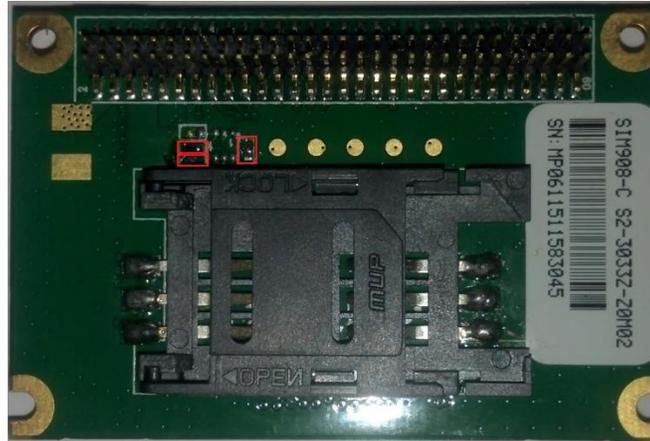


Figura IV.6 – Parte posterior del módulo SIM908-C.

IV.3 Programación del Microcontrolador:

Luego de haber encontrado los comandos necesarios para realizar las tareas buscadas, se procedió a transmitir los distintos comandos al SIM908-C, pero esta vez desde el Microcontrolador.

Para lograr este objetivo se programaron rutinas en Lenguaje C, y así poder evaluar este periférico. La programación se realizó cargando previamente en el microcontrolador un firmware llamado BootLoader. Este firmware funciona de la siguiente manera: cuando se alimenta el microcontrolador, este ejecuta el BootLoader, el cual espera 3 segundos a que llegue un nuevo programa a través del pin de EUSART, para cargar en la memoria no volátil del microcontrolador. Si esto no ocurre, ejecuta el último programa que se haya cargado en su memoria. Esto permite la programación sin retirar el microcontrolador de la placa en la que está siendo utilizado (*Programación In-Circuit*) utilizando solo 2 pines, aunque no permite la depuración de errores en esta condición (*Debug In-Circuit*).

Al elegir el PIC18F26K22, se debió configurar el bootloader propio de este PIC. Para esto se solicitó la ayuda de Walter Gemin, quien me proveyó del archivo de MPLab necesario para luego modificarlo, compilarlo y finalmente, cargárselo al PIC.

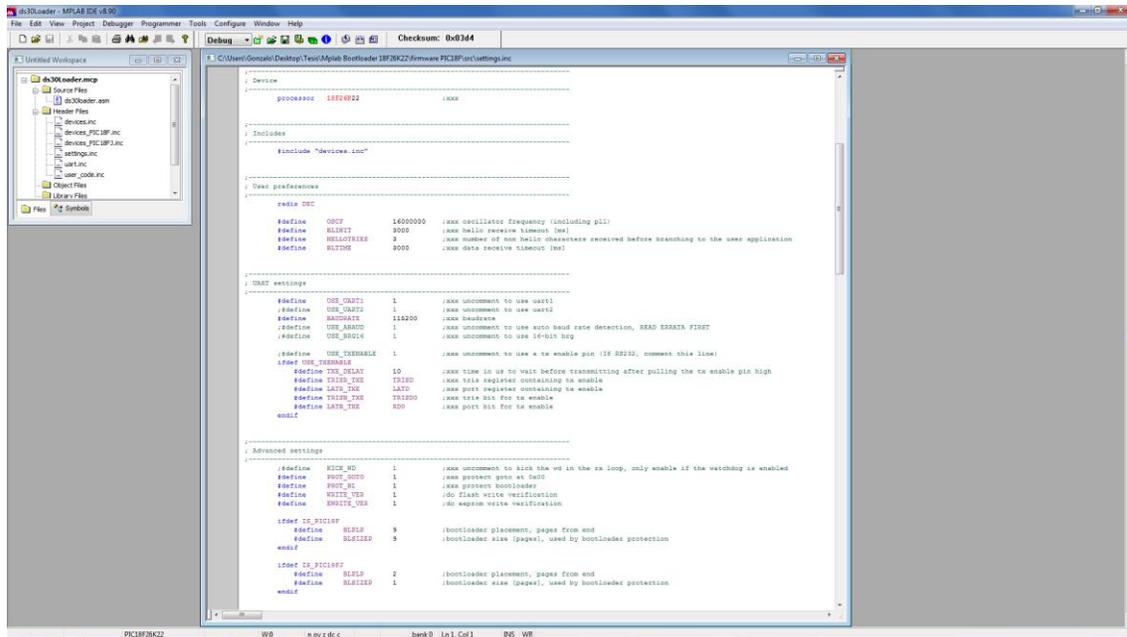


Figura IV.7 – Archivo ds30Loader.mcp correspondiente al PIC18F26K22.

Para poder cargar el bootloader se construyó el dispositivo Programador PICKit 2 de la marca Microchip, cuyo diagrama eléctrico es libre y se puede encontrar en internet. El Programador permite leer y grabar microcontroladores utilizando directamente el software original de Microchip: MPLAB o PICKit 2 v2.61; también posee un modo Debugger que se puede utilizar para depurar programas en tiempo real, correr el programa paso a paso, establecer breakpoints, observar los estados de las entradas, salidas y registros del microcontrolador.

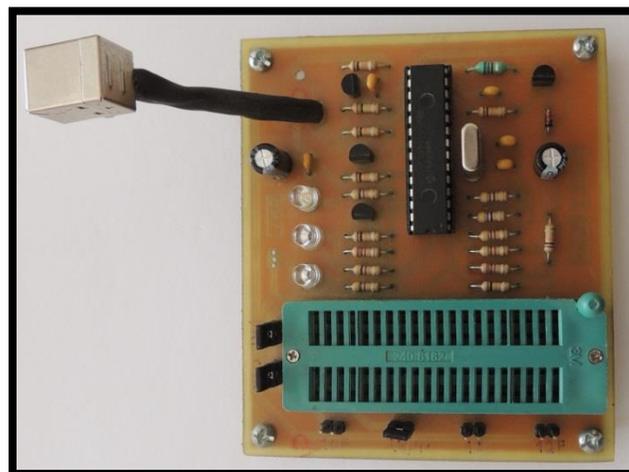


Figura IV.8 – Programador PICKit 2.

Para realizar el programa que controla los diferentes periféricos, se utilizó el software PIC-C Compiler; que consiste en un compilador C muy optimizado que contiene operadores estándar del lenguaje C y funciones incorporados en bibliotecas que son específicas a los registros de PIC, proporcionando a los desarrolladores una herramienta poderosa para el acceso al hardware de las funciones del dispositivo desde el nivel de lenguaje C.

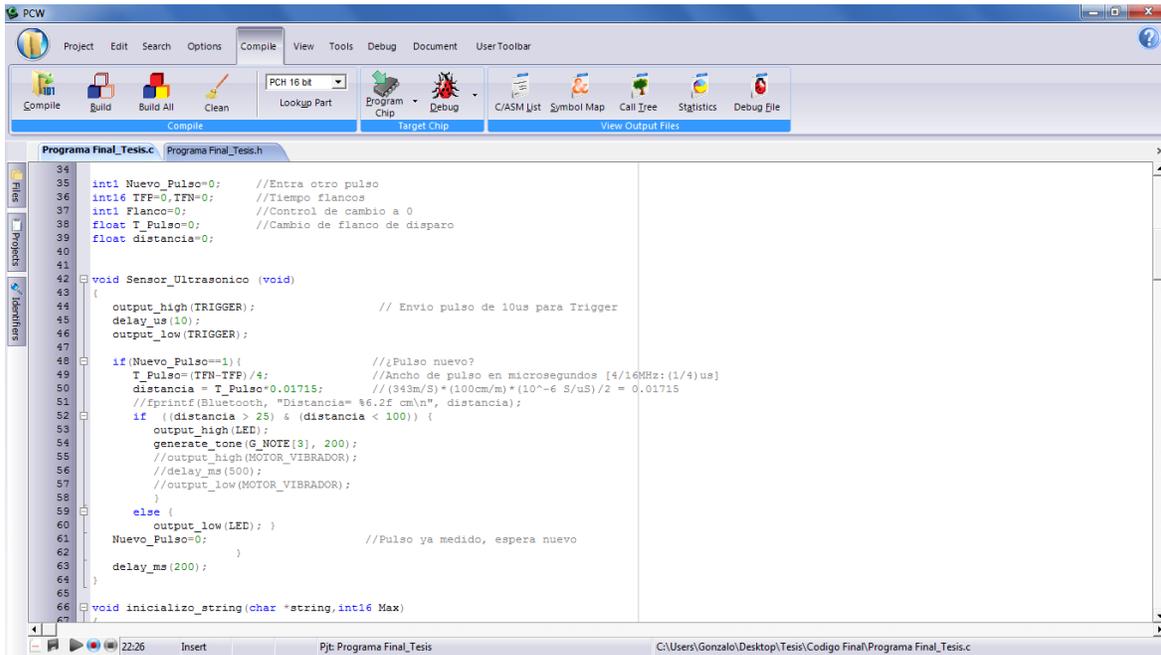


Figura IV.9 – PIC-C Compiler.

IV.4 Diseño:

Una vez evaluados todos los componentes a utilizar, se realizaron pruebas sobre una Protoboard, conectando los diferentes periféricos al microcontrolador. Esto permitió depurar errores eléctricos en el conexionado y encontrar la mejor distribución de los componentes. Luego, se diseñó la placa final mediante el programa “*Altium Designer*”, para posteriormente construir el prototipo de la placa de forma casera, transfiriendo el diseño planchando una impresión del circuito y quemando la placa con ácido cloruro-férrico. A lo largo del diseño se tuvo en cuenta la disposición de los componentes para su conexionado, de forma de reducir lo más posible las dimensiones de la placa.

A continuación, se muestra el diagrama eléctrico del prototipo inicial, que incluye el microcontrolador, sensor ultrasónico, módulo bluetooth, motor vibrador, buzzer, pulsadores, leds; con los resistores, capacitores e inductores correspondientes. En este diseño se le agregó el integrado MAX232, que permite la comunicación serie con la PC.

Al diseño se le agregaron 4 botones programables, los cuales se utilizaron para las pruebas con distintos fines, como encender el GPS, enviar un SMS, contestar una llamada, etc.

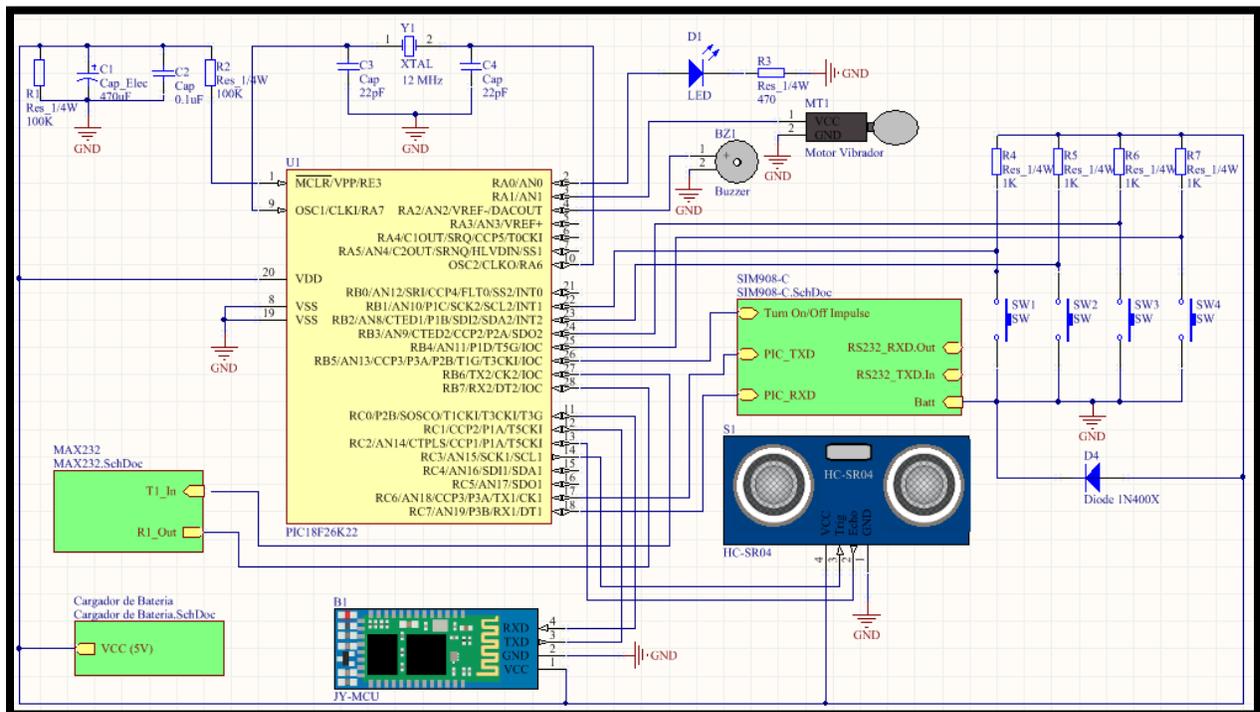


Figura IV.10 – Diagrama Eléctrico Global.

En el circuito se puede observar un bloque verde denominado “MAX232” que incluye el integrado MAX232 que permite conectar el circuito a una PC a través de un cable conversor RS232 a USB; para así poder programar el PIC de forma directa sin tener que retirarlo de la placa y poder depurar el programa, utilizando el software “ds30 Loader Gui”. Esta sección se obvió en el diseño final.

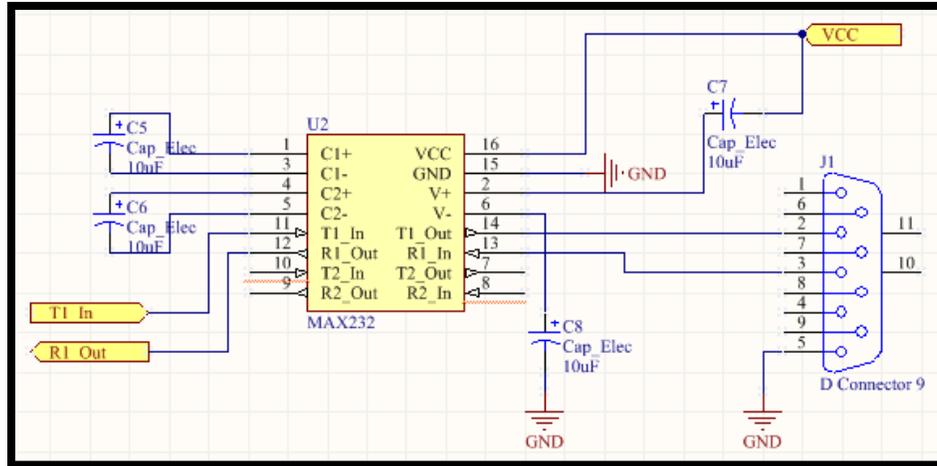


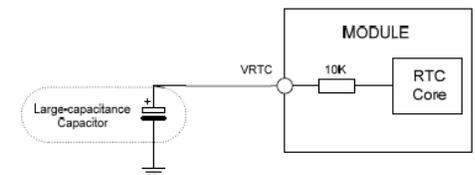
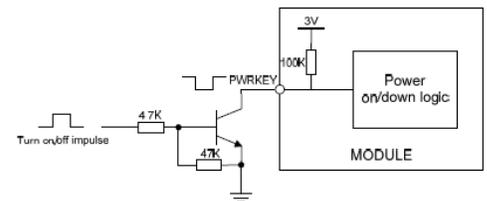
Figura IV.11 – Diagrama Eléctrico del Bloque MAX232.

La adición del módulo Bluetooth fue en un principio para poder comunicarse con un celular, y así poder optar entre usar el modulo GPS o utilizar el GPS integrado a un celular. Pero solo se utilizó este módulo como medio de comunicación inalámbrica para enviar y recibir datos a la PC.

El bloque verde denominado “SIM908-C” consiste en el módulo GPS y GSM/GPRS SIM908-C, el cual contiene los componentes necesarios para poder encenderlo con un impulso enviado desde el PIC, indicador de encendido, entrada y salida de audio, que permiten realizar llamadas telefónicas.

A continuación se detallan las conexiones que se realizaron obtenidas de la hoja de datos correspondiente.

- El módulo se puede encender aplicando un impulso positivo con una duración de al menos un segundo, en el pin PWRKEY.
- Se le agregaron 2 conectores estéreo, para poder realizar llamadas. Se debe tener en cuenta que las pistas de los dos canales del conector deben tener la misma longitud y grosor.
- La fuente de alimentación RTC del módulo puede ser proporcionado por un capacitor a través del pin VRTC, en caso que la batería de reserva se encuentre en estado de bajo voltaje.



- El pin NetLight se puede utilizar para indicar, mediante un led, el estado de la red. El estado de este pin aparece en la siguiente tabla:

Status	SIM908-C behavior
Off	SIM908-C is not running
64ms On/ 800ms Off	SIM908-C not registered the network
64ms On/ 3000ms Off	SIM908-C registered to the network
64ms On/ 300ms Off	PPP GPRS communication is established

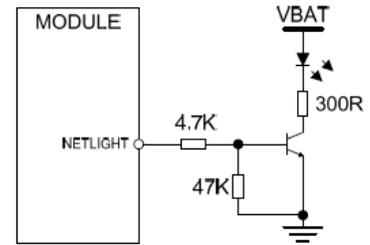


Figura IV.12 – Tabla del estado de pin NetLight.

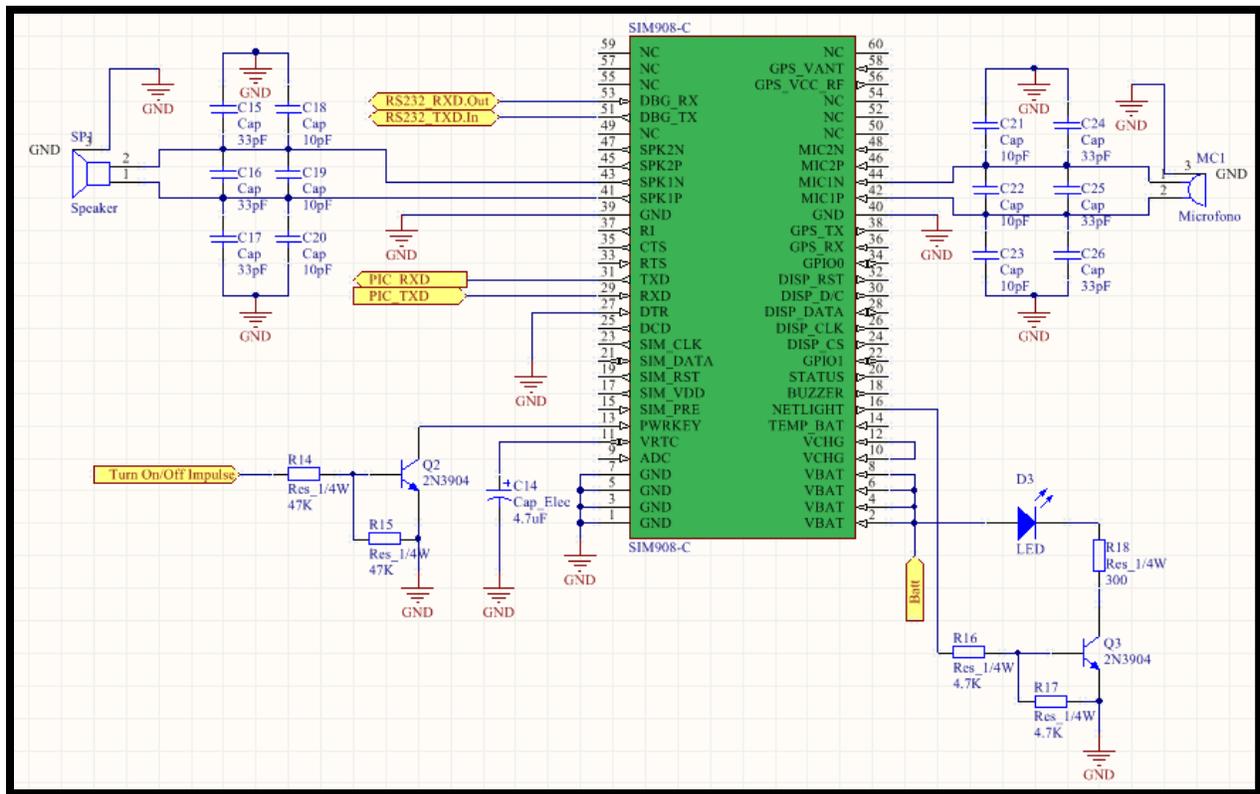


Figura IV.13 – Diagrama Eléctrico del bloque SIM908-C.

Debido a que el SIM908-C admite una fuente de alimentación entre 3.2V y 4.8V; se decidió utilizar una alimentación de 5V, como tensión principal, y agregar un diodo en el pin de entrada VBAT, que provea una caída de aproximadamente 0.6V para no quemar el módulo en cuestión.

Como el dispositivo debía ser portátil, se consideró que el circuito debía ser alimentado mediante una batería; en este caso se utilizó una batería de Tablet, la cual entrega una tensión de 3.7V y 3000mAh. Entonces, para poder obtener los 5V buscados fue necesario agregar un convertor DC-DC Boost.

El convertor DC-DC Boost (Elevador o Step-Up), es un circuito que produce una tensión de salida mayor o igual a la tensión de entrada. En el circuito del convertor Boost se puede observar que V_i es la tensión CC de entrada y que V_o es la tensión CC de salida. Se considera que el capacitor de salida es lo suficientemente grande para mantener constante la tensión en los bornes, despreciando el rizado de tensión. En este convertor siempre se cumple que V_o es mayor que V_i , razón por la cual se le llama “Elevador”. El interruptor S conmuta a una frecuencia $f = 1/T$. Normalmente en los convertidores DC-DC se distinguen dos modos de operación, según la corriente por el inductor L se anule en el período de operación T: Modo de Conducción Continua y Modo Conducción Discontinua.

Cuando el interruptor S está conduciendo la energía entregada por la fuente de entrada V_i es acumulada en el inductor L, y cuando el interruptor es bloqueado esa energía, junto con la proveniente de V_i , es transferida a la salida.

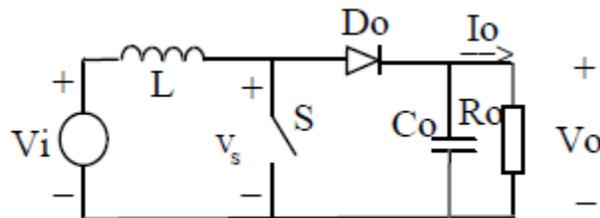


Figura IV.14 – Circuito Boost.

El siguiente diagrama comprende el cargador de batería y elevador de tensión, basado en un diseño de la marca Maxim. También se le agregó un conector USB y switch, para poder optar entre alimentar el circuito desde la batería o directamente a través del conector USB. La opción de alimentar el circuito de forma directa del conector USB, fue para la etapa de desarrollo, se obvió en el diseño final.

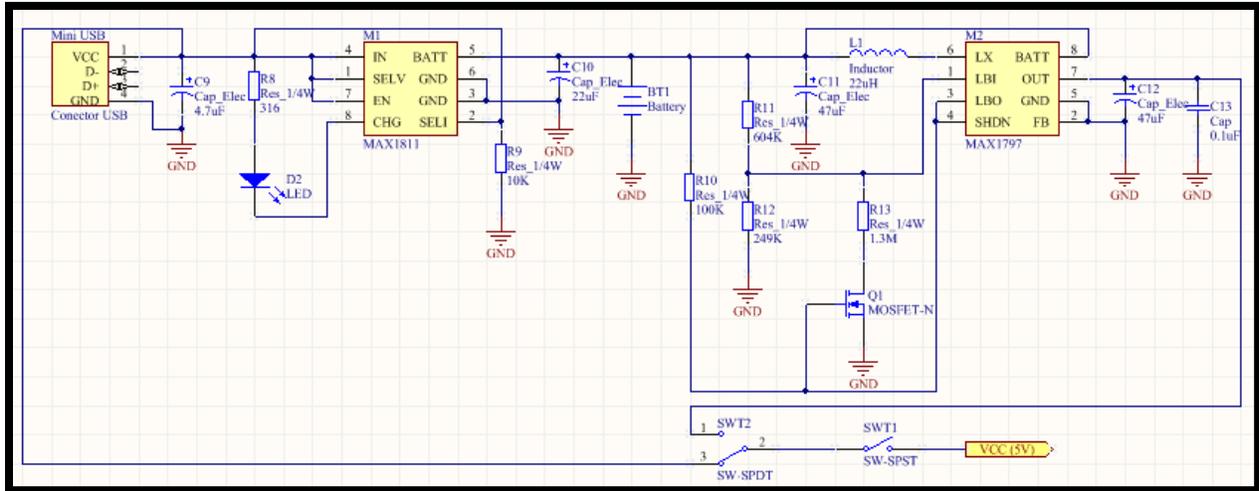


Figura IV.15 – Diagrama Eléctrico del bloque Alimentación y Cargador de Batería.

El programa “*Altium Designer*” permite visualizar un modelo 3D de placa final con todos los componentes, utilizando distintas librerías, para así ver la distribución y volumen que ocupan dichos componentes desde cualquier perspectiva.

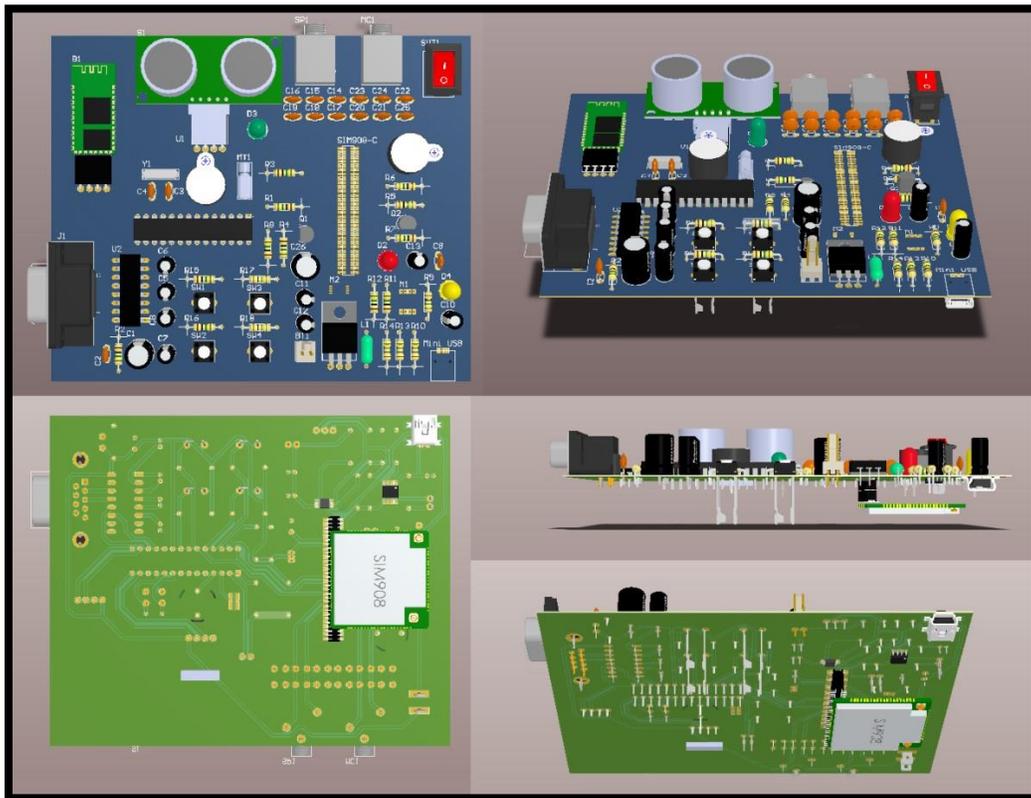


Figura IV.16 – Diseño 3D de la Placa Final.

IV.5 Prototipo:

A continuación se puede observar la placa de circuito impreso (PCB), con todos los componentes soldados y con los módulos insertados en sus respectivos zócalos. Y comparando con el diseño tridimensional anteriormente visto, se puede observar que son casi idénticos.

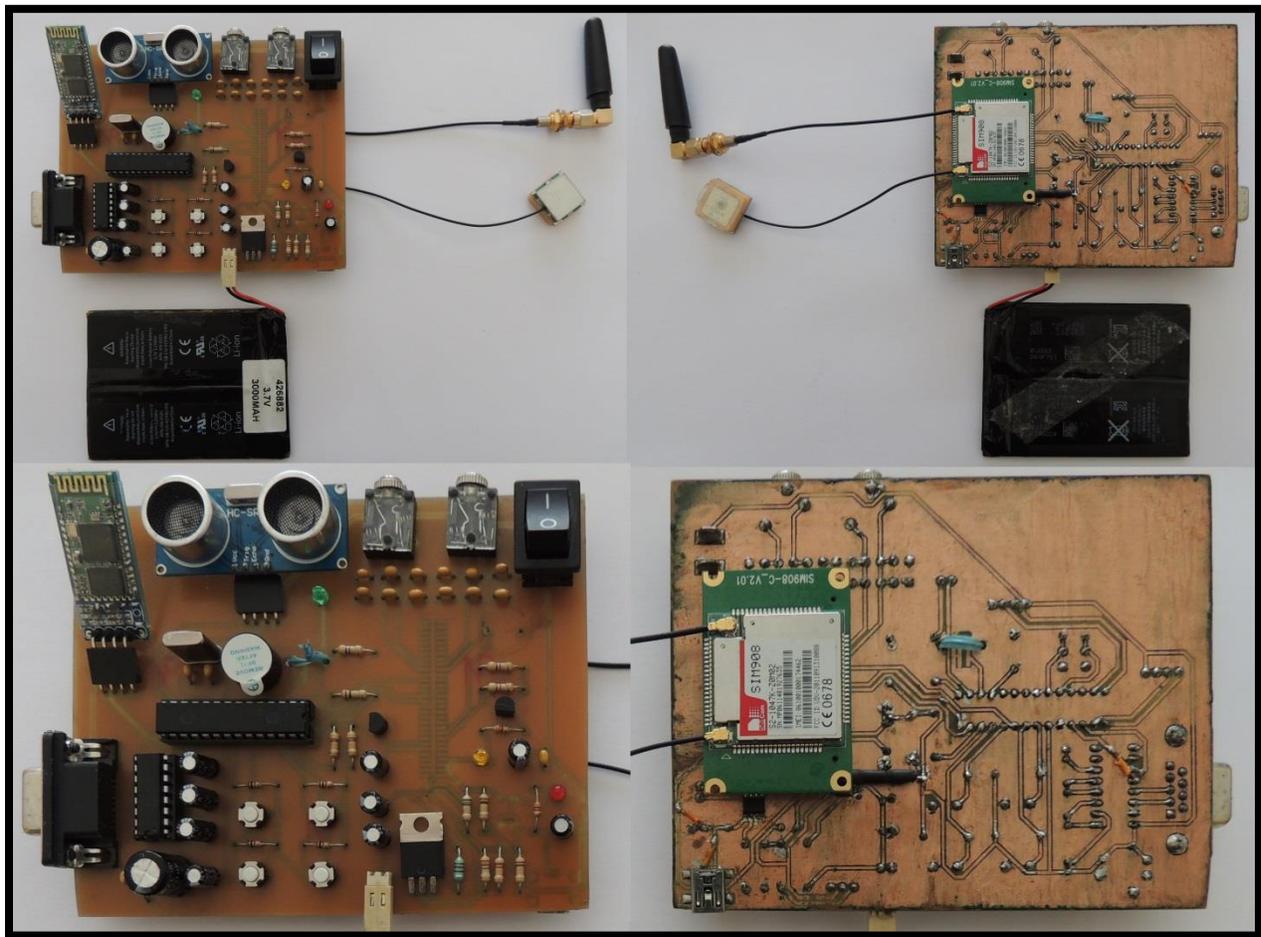


Figura IV.17 – Placa de Circuito Impreso Final.

Con este prototipo se pudo:

- ✓ Depurar todas las conexiones eléctricas.
- ✓ Verificar que todos los módulos funcionasen dentro de los parámetros especificados.
- ✓ Evaluar la comunicación con la PC a través de módulo bluetooth.
- ✓ Con el MAX232 se pudo evaluar los distintos programas, cargándolos directamente al PIC, para así depurar todos los bugs que se fueron encontrando.

- ✓ Evaluar distintas antenas para el SIM908-C.
- ✓ Evaluar los distintos modos de alerta del Sensor Ultrasónico: Led, Buzzer, Motor Vibrador.

Teniendo en cuenta los consumos de corriente de los distintos dispositivos utilizados, que más corriente demandan y en el momento que más consumen (al encenderse), se obtiene un total de 332 mA. El consumo más grande proviene del PIC, pero solo ocurre en el caso que se estén utilizando todas las entradas y salidas, y estén transmitiendo al mismo tiempo, que en el caso de este circuito no es así.

Pero si tenemos en cuenta los 332 mA con la batería utilizada de 3.7V y 3000mAH, este circuito podría operar durante 9 horas, en el peor caso de consumo.

Dispositivo	Activo	Reposo
HC-SR04 (Sensor Ultrasónico)	15 mA	
PIC18F26K22	200 mA (Max)	0,1 mA
SIM908-C	77 mA	1 mA
HC-06 (Modulo Bluetooth)	40 mA (Al Conectarse)	8 mA (En Comunicación)

Figura IV.18 – Tabla de Consumo de corriente de los distintos Dispositivos.

En la siguiente imagen se puede observar el mensaje que fue recibido al número de celular ingresado en la configuración del programa. Este SMS contiene un link de Google Maps con las coordenadas que detecto el módulo GPS incluidas. Se optó por hacerlo de esta forma, ya que la gran mayoría de usuarios de celulares poseen un dispositivo Smart, que permite el envío de datos, es decir, que pueden conectarse a internet en cualquier momento a través de las antenas de celular o wifi; y así poder observar la localización de la persona no vidente de forma gráfica con las calles detalladas. La ubicación mostrada es la Facultad de Ingeniería, ya que es ahí donde se realizó la prueba.

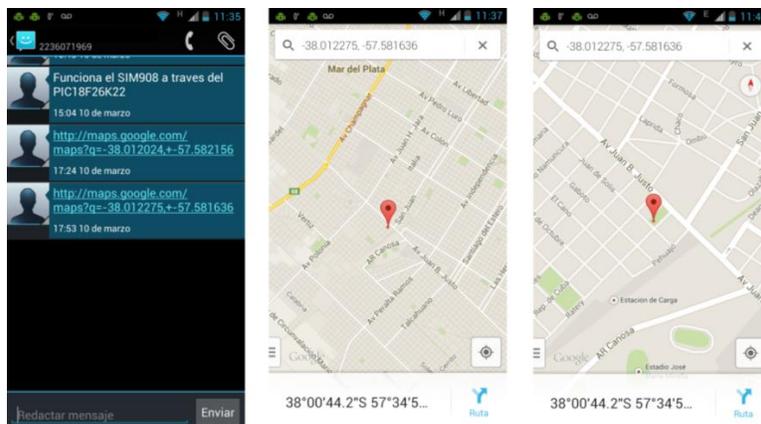


Figura IV.19 – Capturas de Celular.

Finalmente, se construyó un segundo prototipo, en el cual se eliminó la sección correspondiente al MAX232. Se pudo reducir las dimensiones del dispositivo, pero no lo suficiente. A continuación se puede observar el prototipo terminado con los módulos colocados en los zócalos correspondientes y las antenas de GSM y GPS.

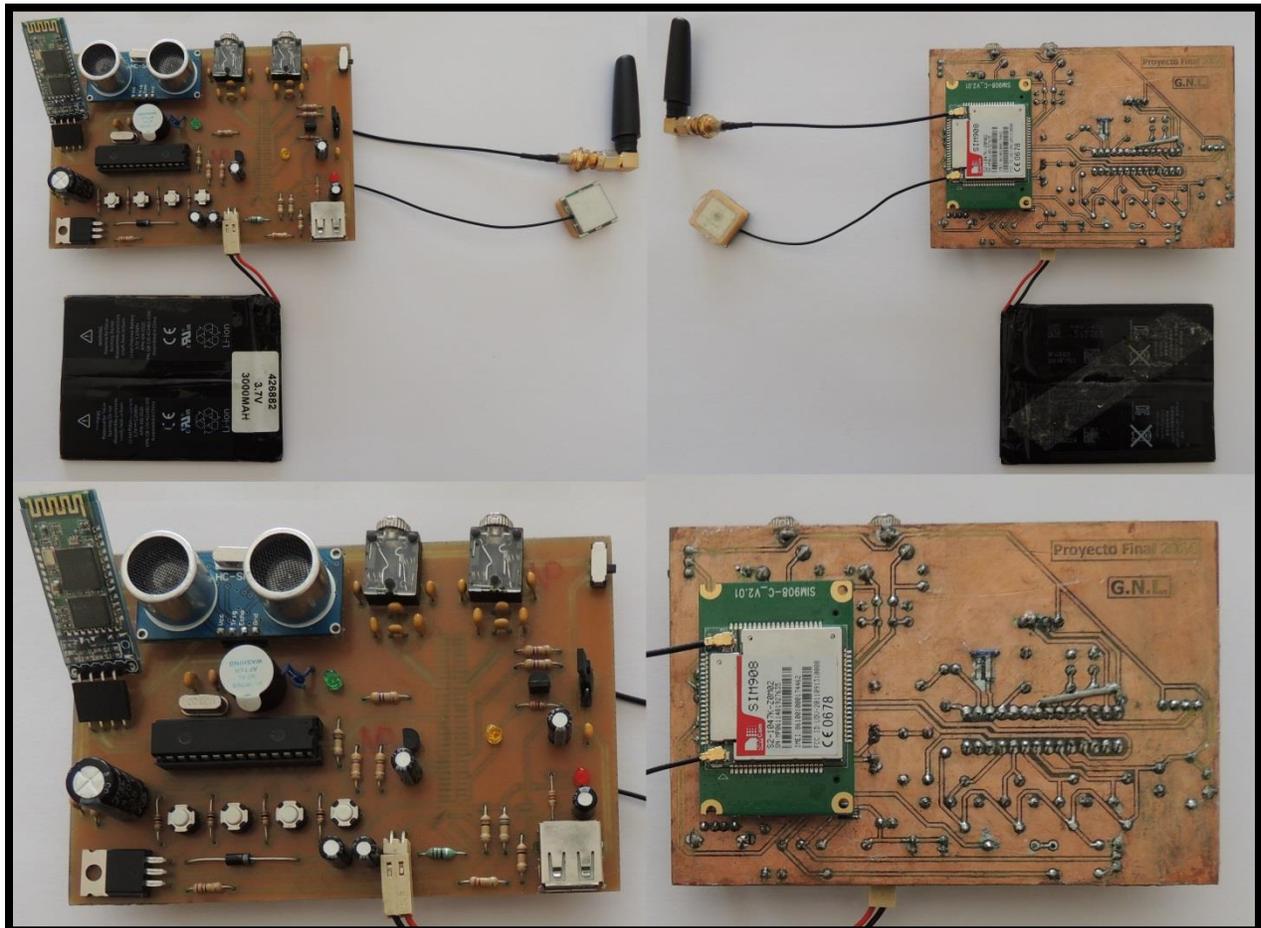


Figura IV.20 – Prototipo Final.

IV.6 Mejoras Futuras

A lo largo del desarrollo del proyecto, se observaron ciertos aspectos que se podrían mejorar y hacer más fácil la utilización del dispositivo para el usuario final, los cuales se consideraron interesantes para que otro estudiante pueda desarrollar posteriormente. A continuación se nombran los más importantes:

- Utilizar componentes SMD de montaje superficial, lo cual reduciría considerablemente su tamaño físico.
- Utilizar el módulo bluetooth para comunicarse con un celular y tener la opción de omitir el módulo SIM908-C con sus antenas.
- Realizar un programa, por ejemplo en VisualBasic, para poder ingresar más números de celulares en la memoria del SIM908-C; y así poder elegir a quien enviar el SMS.
- Poder enviar un SMS al dispositivo desde un celular, y que este le conteste la posición de la persona no vidente, sin que ella tenga que intervenir.
- Añadir alertas con mensajes de sonido, por ejemplo, utilizando el integrado APR9600.

V. Conclusión:

Durante este proyecto final se diseñó de forma teórica un circuito capaz de:

- ✓ Detectar obstáculos aéreos en un área aproximada de 80 cm² de diámetro a una distancia de 1.5 metros, alertando a la persona no vidente del objeto que puede encontrar a su paso, para poder alterar su trayectoria y evitar dicho objeto.
- ✓ Detectar la ubicación geográfica de la persona mediante satélites de GPS.
- ✓ Enviar a otra persona mediante mensaje de texto dicha ubicación.
- ✓ Establecer un enlace de comunicación entre el circuito y una PC mediante Bluetooth.

Luego de la búsqueda bibliográfica, y del análisis de los distintos elementos disponibles en el mercado se alcanzó un modelo optimizado del sistema, para finalmente implementarlo de forma práctica.

Debido a que no se encontraron las versiones virtuales de los módulos y sensores que se decidieron utilizar, para poder realizar simulaciones en programas como Proteus o Altium Designer, se decidió evaluar el diseño del circuito separado en etapas. Es decir, se armó cada etapa en una protoboard o se utilizó una placa de desarrollo para poder verificar el funcionamiento de cada dispositivo, para luego, fabricar el circuito completo y poder comprobar el comportamiento que tendrían todos los dispositivos interactuando juntos.

Cabe destacar que para poder alimentar el circuito con una batería típica de 3.7V, se debió incorporar un convertidor DC-DC Boost para poder elevar la tensión a 5V, que es requerida por el PIC, sensor ultrasónico y el módulo bluetooth. El módulo SIM908-C permite una tensión de alimentación máxima de 4.8V, por lo tanto se colocó un diodo entre los 5V y el pin de alimentación, ya que este provee una caída de 0.6V, aproximadamente.

Finalmente, luego de todo análisis y evaluación de los distintos integrados utilizados, se construyó un prototipo que fue capaz de cumplir con todos los objetivos planteados en el comienzo de este proyecto.

El dispositivo final, lamentablemente no se pudo probar con personas no videntes, debido a que el tamaño de la placa no se redujo lo suficiente para que fuese cómodo para el usuario final. La Directora de UMASDECA, me recalcó que a las personas no videntes no les gusta tener que colocarse varios dispositivos en distintas partes del cuerpo, ya que sería incómodo para el uso diario. Pero si se utilizan componentes de montaje superficial, se podría reducir el tamaño y el peso lo suficiente para que parezca la hebilla de un cinturón; también se podría utilizar una carcasa para ocultar el circuito, y así pasar de ser percibido.

Por último, con esta experiencia se logró adquirir práctica en el diseño y armado de circuitos; aprender a programar en un nuevo lenguaje; corroborar conceptos incorporados en materias estudiadas durante la carrera que permitieron resolver los diferentes problemas que se fueron encontrando a lo largo del desarrollo de este proyecto final.

VI. Bibliografía:

1. Gwirc, Sergio. “*Sensores Ultrasonicos: Respuesta a Distintas Formas de Ondas de Emisión*”. Centro de Investigación y Desarrollo en Electrónica e Informática (CITEI), Instituto Nacional de Tecnología Industrial (INTI): San Martin, Buenos Aires, 1998.
2. Blitz, Jack. “*Fundamentals of Ultrasonics*”. New York, Plenum Press, 1967.
3. Ferdeghini, Fernando. “*Sistema de Detección Combinado para Sensores Ultrasonicos*”. Centro de Investigación y Desarrollo en Electrónica e Informática (CITEI), Instituto Nacional de Tecnología Industrial (INTI): San Martin, Buenos Aires, 1998.
4. Bhatlawande, Shripad. “*Ultrasonic Spectacles and Waist-belt for Visually Impaired and Blind Person*”. Kharagpur, India, 2012.
5. Marpaung ,Julius. “*An Implementation of an Ultrasonic Device for the Visually Impaired*”. Stillwater, Oklahoma, 2006.
6. García, Javier. “*Aprenda lenguaje ANSI C como si estuviera en primero*”. San Sebastián: Universidad de Navarra, 1998.
7. Schildt, Herbert. “*C++: The Complete Reference, 4th Edition*”. Estados Unidos de América, 2003, 1058 pp.
8. Corrales V., Santiago. “*Electrónica Práctica con Microcontroladores PIC*”. Ecuador, 2006, 181 pp.
9. García Breijo, Eduardo. “*Compilador C CCS y simulador Proteus para Microcontroladores PIC*”. Barcelona, 2008, 270 pp.
10. Grewal, Mohinder. “*Global Positioning Systems, Inertial Navigation, and Integration*”. 2001, 409 pp.
11. El-Rabbany, Ahmed. “*Introduction to GPS*”. Norwood, Massachusetts, 2002.

VII. Anexo

A continuación se anexa el programa de este proyecto final realizado en Lenguaje C.

Programa Final Tesis.c

```
#include <Programa Final_Tesis.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <TONES.C>

#define TRIGGER          PIN_C3
#define LED              PIN_A0
#define MOTOR_VIBRADOR  PIN_A1
#define BUZZER          PIN_A2
#define PWRKEY          PIN_B5
#define BOTON_1         PIN_B0
#define BOTON_2         PIN_B1
#define BOTON_3         PIN_B3
#define BOTON_4         PIN_B4

//Vectores Globales
unsigned char Estado[40];
unsigned char buffer[100];
char Trama_GPS[100];
char Latitud[14];
char Longitud[14];

//Variables Globales
unsigned int8 char_in;
int8 Listo;

//Inicializo Variables
int1 Nuevo_Pulso=0;
int16 TFP=0,TFN=0;
int1 Flanco=0;
float T_Pulso=0;
float distancia=0;

//Entra otro pulso.
//Tiempo de los flancos Positivo y Negativo.
//Control de cambio a 0.
//Cambio de flanco de disparo.

void Sensor_Ultrasonico (void)
{
    output_high(TRIGGER);
    delay_us(10);
    output_low(TRIGGER);
    // Envio pulso de 10us para Trigger
}
```

```

if(Nuevo_Pulso==1){ //¿Pulso nuevo?
  T_Pulso=(TFN-TFP)/4; //Ancho de pulso en microsegundos [4/16MHz:(1/4)us]
  distancia = T_Pulso*0.01715; //(343m/S)*(100cm/m)*(10^-6 S/uS)/2 = 0.01715
  //fprintf(Bluetooth, "Distancia= %6.2f cm\n", distancia);
  if ((distancia > 25) & (distancia < 150)) {
    output_high(LED);
    generate_tone(G_NOTE[3], 200);
    //output_high(MOTOR_VIBRADOR);
    //delay_ms(500);
    //output_low(MOTOR_VIBRADOR);
  }
  else {
    output_low(LED); }
  Nuevo_Pulso=0; //Pulso ya medido y queda a la espera de un nuevo pulso.
}
delay_ms(200);
}

void Configuracion_GPS()
{
  fprintf(SIM908,"AT+CGSPWR=1\r"); //Enciendo el GPS.
  delay_ms(200);
  fprintf(SIM908,"AT+CGPSRST=0\r"); //Reseteo en Cold Start.
  delay_ms(200);
  fprintf(SIM908,"AT+CGPSIPR=9600\r"); //Configuro el Baud rate del GPS.
  delay_ms(200);
}

void GPS_Listo()
{
  int8 i=0;

  while (Listo==0)
  {
    i++;
    //fprintf(Bluetooth, "Localizando...%i\n",i);
    delay_ms(5000);
    fprintf(SIM908,"AT+CGPSSTATUS?\r");
  }
}

void Conversion (char Trama_GPS[])
{
  int16 latg[3],latm[9],longg[3],longm[9];
  float latitudgoogle,longitudgoogle;

  strncpy(latg,&Trama_GPS[9],2); // Convierte a coordenadas de Googlemaps.

```

```

strncpy(latm,&Trama_GPS[11],9);
strncpy(longg,&Trama_GPS[23],2);
strncpy(longm,&Trama_GPS[25],9);
latitudgoogle=(atof(latm))/60;
longitudgoogle=(atof(longm))/60;
sprintf(&Latitud[2],"%0.6f",latitudgoogle);
Latitud[0]=latg[0];
Latitud[1]=latg[1];
sprintf(&Longitud[2],"%0.6f",longitudgoogle);
Longitud[0]=longg[0];
Longitud[1]=longg[1];
}

void Envio_SMS() //Función para enviar mensajes SMS.
{
    char ctrlz = 26; //combinación de teclas "ctrl+z".

    fprintf(SIM908,"AT+CMGF=1\r"); //configura el tipo de formato de mensaje de texto SMS
    delay_ms(400); //formato TEXTO).
    fprintf(SIM908,"AT+CMGS=\"+542235730541\"\r");
    delay_ms(200);
    fprintf(SIM908,"http://maps.google.com/maps?q=-%s,+%s",Latitud,Longitud);
    delay_ms(200);
    fprintf(SIM908,"%c",ctrlz); //Se envía el mensaje.
}

#int_RDA2
void RDA2_isr(void)
{
    char_in++;
    buffer[char_in]=fgetc(SIM908);
    if (buffer[char_in]==0x0A) //<CR> <LF> = 0D 0A = \r \n.
    {
        buffer[char_in-1]="\0";
        //fputs(buffer,Bluetooth);
        if (strcmp(Estado,buffer)==0)
        {
            Listo=1;
        }
        if (char_in>60)
        {
            strcpy(Trama_GPS,&buffer[0]);
        }
        char_in=-1;
    }
}
}

```

```

#int_CCP1
void CCP1_int() //Función Interrupción.
{
    if(Flanco==0){ //Flanco Positivo.
        TFP=CCP_1; //Carga del valor del registro CCPR1 en flanco positivo.
        setup_ccp1(CCP_CAPTURE_FE); //Configuración modo Captura en flanco Negativo.
        Flanco=1; } //Control de cambio de flanco.
    else { //Flanco Negativo.
        TFN=CCP_1; //Carga del valor del registro CCPR1 en flanco Negativo.
        setup_ccp1(CCP_CAPTURE_RE); //Configuración modo Captura en flanco Positivo.
        Flanco=0; //Control de cambio de flanco.

        if(Nuevo_Pulso==0){ //Fin de pulso.
            Nuevo_Pulso=1; } //Nuevo pulso a medir.
        }
    }
}

void main()
{
    enable_interrupts(GLOBAL);
    setup_ccp1(CCP_CAPTURE_RE); // Configuro CCP1 para que capture el flanco Positivo.
    setup_timer_1(T1_INTERNAL); // Seteo timer 1.
    enable_interrupts(INT_CCP1); // Seteo la interrupción para el flanco Negativo.
    enable_interrupts(INT_RDA2);

    Listo=0;
    char_in=-1;
    Estado="+CGPSSTATUS: Location 3D Fix\0";

    while(TRUE)
    {
        Sensor_Ultrasonico (); //El sensor Ultrasónico siempre está activo.
        if (input(BOTON_1)==0)
        {
        }
        if (input(BOTON_2)==0) //Enciende el modulo SIM908-C.
        {
            output_low(PWRKEY);
            delay_ms(100);
            output_high(PWRKEY);
            delay_ms(1000);
            output_low(PWRKEY);
            delay_ms(7000);
        }
        if (input(BOTON_3)==0) //Obtiene las coordenadas, las convierte y las envía.
        {
            Configuracion_GPS();

```

```

delay_ms(200);
GPS_Listo();
if (Listo==1)
{
    delay_ms(200);
    fprintf(SIM908,"AT+CGPSINF=2\r");    //Localización del GPS: $GPGGA.
    delay_ms(200);
    disable_interrupts(INT_RDA2);
    delay_ms(2000);
    Conversion(Trama_GPS);
    //fprintf(Bluetooth,"Latitud: %s\n",Latitud);
    //fprintf(Bluetooth,"Longitud: %s\n",Longitud);
    Envio_SMS();
    Listo=0;
}
delay_ms(300);    //Importante no quitar!!!
}
if (input(BOTON_4)==0)
{
    fprintf(SIM908,"ATA=2\r");    //Atiende la llamada
}
}
}

```

Programa Final Tesis.h

```

#include <18F26K22.h>
#define adc=16

#FUSES HS           //HS oscillator (medium power 4-16 MHz).
#FUSES H4           //High speed osc with HW enabled 4X PLL
#FUSES PLLEN        //4X HW PLL disabled, 4X PLL enabled in software
#FUSES PRIMARY      //Primary clock is system clock when scs=00
#FUSES NOFCMEN      //Fail-safe clock monitor disabled
#FUSES IESO         //Internal External Switch Over mode disabled
#FUSES NOBROWNOUT   //No brownout reset
#FUSES NOPBADEN     //PORTB pins are configured as digital I/O on RESET
#FUSES NOLVP        //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOXINST      //Extended set extension and Indexed Addressing mode disabled
                    // (Legacy mode)
#FUSES WDT_NOSLEEP  //PORTB pins are configured as analog input channels on RESET
#FUSES PBADEN       //Timer1 configured for low-power operation
#FUSES LPT1OSC
#FUSES NODELAYINTOSC
#FUSES TIMER3CO
#FUSES CCP2B5

```

```
#FUSES MCLR //Master Clear pin enabled
#FUSES WDT1 //Watch Dog Timer uses 1:1 Postscale
#FUSES NOPUT //No Power Up Timer
#FUSES BORV45 //Brownout reset at 4.5V
#FUSES NOLPT1OSC //Timer1 configured for higher power operation
#FUSES STVREN //Stack full/underflow will cause reset
#FUSES NODEBUG //No Debug mode for ICD
#FUSES PROTECT //Code protected from reading
#FUSES NOCPB //No Boot Block code protection
#FUSES NOCPD //No EE protection
#FUSES NOWRT //Program memory not write protected
#FUSES NOWRTC //configuration not registers write protected
#FUSES NOWRTB //Boot block not write protected
#FUSES NOWRTD //Data EEPROM not write protected
#FUSES NOEBTR //Memory not protected from table reads
#FUSES NOEBTRB //Boot block not protected from table reads

#USE delay(clock=48000000)
#USE rs232(baud=115200,parity=N,UART1,bits=8,stream=Bluetooth)
#USE rs232(baud=9600,parity=N,UART2,bits=8,stream=SIM908)
```