

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACION DE TECNOLOGOS

IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO DE ALERTA DE OBSTÁCULOS PARA PERSONAS NO VIDENTES MEDIANTE EL USO DE UN SENSOR.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRONICA Y TELECOMUNICACIONES

GOMEZ BASTIDAS PABLO MIGUEL

pabloasoma@hotmail.com

TRUJILLO FONSECA ANDRES JULIAN

andrestrujillo102010@hotmail.com

DIRECTOR: ING. PABLO LOPEZ MERINO

pablo.lopez@epn.edu.ec

QUITO, DICIEMBRE 2011

DECLARACION

Nosotros, Andrés Julián Trujillo Fonseca y Pablo Miguel Gómez Bastidas, declaro bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Andrés Julián Trujillo Fonseca.

Pablo Miguel Gómez Bastidas.

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por los señores GOMEZ BASTIDAS PABLO MIGUEL y TRUJILLO FONSECA ANDRES JULIAN, bajo mi supervisión.

Ing. Pablo López Merino.

DIRECTOR DE PROYECTO

AGRADECIMIENTO.

A Dios por permitirme cumplir una mas de mis metas profesionales ya que el a estado presente en todo momento brindándome inteligencia y sabiduría.

A mis padres quienes me han brindado todo el apoyo que he necesitado y por ser un ejemplo de perseverancia, dedicación y humildad.

Agradezco a la Escuela Politécnica Nacional por permitirme aprender las bases teóricas y técnicas que me han servido para la elaboración de este trabajo y que me servirán para todo mi vida profesional.

Al Ing. Pablo López por su dedicación y apoyo total al realizar este trabajo de titulación.

A mi familia y amigos en general que de uno u otro modo me han brindado su apoyo para la culminación de mi vida estudiantil y comienzo de mi vida profesional.

Andrés Trujillo

DEDICATORIA

A mis padres Emilio y Marina quienes han estado conmigo siempre apoyándome y brindándome su cariño lo que me ha permitido llegar muy lejos en la vida y por lo que los amo y siempre estaré agradecido.

A mi hermano, esposa e hijos a quienes les debo mucho por todo el apoyo que me han dado todos estos años.

Andrés Trujillo

AGRADECIMIENTO.

A Dios por permitirme cumplir con las metas que me propuesto en mi vida.

A mi padre que me ha enseñado a no rendirme y levantarme de todo fracaso que se me obstaculizado en mi vida.

A la Escuela Politécnica Nacional que me han entregado todos los conocimiento para ejercer en mi vida profesional.

Al Ing. Pablo López que nos apoyó con su paciencia y dedicación en nuestro proyecto de tesis.

A Andrés Trujillo que sin su apoyo y dedicación no se hubiera culminado en la realización de nuestro proyecto.

Pablo Gómez

DEDICATORIA

Con mucho cariño para mi padre Miguel, quien es mi fortaleza para continuar con mis metas.

Para mi familia quien se ha preocupado por cada paso que realizo.

Pablo Gómez

INDICE

INDICE	vii
INDICE DE FIGURAS	xiii
INDICE DE TABLAS	xvi
INTRODUCCION	1
RESUMEN	2
CAPÍTULO I	4
FUNDAMENTOS TEÓRICOS	4
1.1. INTRODUCCIÓN	4
1.2. Instrumentos y sistemas de ayuda a la movilización de personas no videntes	5
1.2.1. INSTRUMENTOS DE BAJA TECNOLOGÍA	5
1.2.1.1. Bastón blanco	5
1.2.1.1.1. Técnica de uso	6
1.2.1.1.1.1. Técnica diagonal	7
1.2.1.1.1.2. Técnica rítmica	7
1.2.2. Sistemas electrónicos de ayuda a la movilidad de personas invidentes	8
1.2.2.1. Eye stick, bastón con sensor para ayudar a los invidentes	9
1.2.2.2. Bastón electrónico para ciegos	10
1.2.2.2.1. Ocho años de investigación	11
1.2.2.2.2. Apoyo GPS	11
1.2.2.3. Kinect ayuda para ciegos	12
1.2.2.4. Una vara para ciegos	13
1.2.2.5. Visión de murciélago para ciegos	14
1.3. PROTOCOLO DE COMUNICACIÓN SERIAL	16
1.3.1. DEFINICION DE COMUNICACIÓN SERIAL	16
1.3.1.1. Codificación no auto reloj	19

1.3.1.2. Codificación auto reloj.....	20
1.3.1.3. Método asíncrono.....	20
1.3.1.4. Método síncrono.....	21
1.4 MICROCONTROLADORES.....	21
1.4.1. INTRODUCCIÓN.....	21
1.4.2. ¿QUÉ ES UN MICROCONTROLADOR?.....	22
1.4.3. APLICACIONES DE LOS MICROCONTROLADORES.....	23
1.4.4. TIPOS DE ARQUITECTURAS DE MICROCONTROLADORES.....	23
1.4.4.1. Arquitectura Von Neumann.....	23
1.4.4.2. Arquitectura Harvard.....	24
1.4.5. ORGANIZACIÓN INTERNA.....	26
1.4.5.1. El procesador.....	26
1.4.5.1.1. CISC.....	26
1.4.5.1.2. RISC.....	26
1.4.5.1.3. SISC.....	27
1.4.5.2. Tipos de memorias.....	27
1.4.5.3. Puertas de entrada y salida.....	28
1.4.5.4. Reloj principal.....	28
1.4.6. RECURSOS ESPECIALES.....	28
1.4.6.1. Temporizadores o timers.....	29
1.4.6.2. Perro guardián o watchdog.....	30
1.4.6.3. Protección ante fallo de alimentación o brownout.....	30
1.4.6.4. Estado de reposo o de bajo consumo.....	30
1.4.6.5. Conversor A/D (CAD).....	31
1.4.6.6. Conversor D/A (CDA).....	31
1.4.6.7. Comparador analógico.....	31
1.4.6.8. Modulador de anchura de impulsos o PWM.....	31
1.4.6.9. Puertos digitales de E/S.....	32
1.4.6.10. Puertas de comunicación.....	32
1.4.6.10.1 COMUNICACIÓN USART.....	33
1.4.7. LA FAMILIA DE LOS PIC.....	34

1.4.7.1. Introducción.....	34
1.4.7.2. PIC16F877A.....	35
1.4.7.2.1. Introducción.....	35
1.4.7.2.2. Características.....	36
1.4.7.2.2.1. Características generales.....	36
1.4.7.2.2.2. Características internas.....	37
1.4.7.2.3. Diagrama de bloques de su arquitectura interna.....	38
1.4.7.2.4. Diagrama de pines.....	40
1.4.7.2.5. Descripción de cada pin del PIC16F877A.....	40
1.4.7.2.6. Dispositivos periféricos.....	43
1.4.7.2.7. Memorias.....	44
1.4.7.2.7.1 La memoria EEPROM ò memoria de programa.....	44
1.4.7.2.7.2. La memoria RAM.....	44
1.4.7.2.7.3. Configuración de los puertos del PIC.....	45
1.5. TRANSDUCTORES.....	47
1.5.1. INTRODUCCIÓN.....	47
1.5.2 SENSORES DE DISTANCIA.....	47
1.5.3 SENSORES ULTRASÓNICOS.....	48
1.5.3.1 Funcionamiento básico de los ultrasonidos.....	48
1.5.3.2. Sensor ultrasónico SRF05.....	49
1.5.3.2.1. Modo 1 (compatible con SRF04).....	49
1.5.3.2.1.1 Diagrama de tiempos del srf05 en modo 1.....	50
1.5.3.2.2. Modo 2 (pin único para la señal de activación y eco).....	51
1.5.3.2.2.1 Diagrama de tiempos del srf05 en modo 2.....	51
1.5.3.2.3. Cómo calcular la distancia.....	52
1.5.3.2.4. Los otros 5 pines del sensor.....	53
1.5.3.2.5. Patrón y el ancho del haz.....	53
1.6. MODULO DE VOZ USB-SD MP3 WT9501M03.....	54
1.6.1. CARACTERISTICAS.....	54
1.6.2. DESCRIPCION DE PINES.....	55
1.6.3. MODO DE FUNCIONAMIENTO.....	56

1.6.3.1. Modo pulsador.....	56
1.6.3.2. Modo de control serial.....	56
1.6.3.2.1. Descripción del código de operación.....	57
1.6.3.2.2. Código de inicio (startcode).....	58
1.6.3.2.3. Tamaño de la trama (data length).....	59
1.6.3.2.4. Código de finalización (endcode).....	59
1.7 LCD.....	59
1.7.1. INTRODUCCION.....	59
1.7.2. CARACTERISTICAS.....	59
1.7.3. LCD ALFANUMERICOS.....	61
1.7.3.1. Introducción.....	62
1.7.3.2. Conexiones.....	62
1.7.3.3. Funcionamiento.....	65
CAPITULO II.....	66
CONSTRUCCIÓN DEL HARDWARE DEL PROYECTO.....	66
2.1. INTRODUCCIÓN.....	66
2.2. ARQUITECTURA DE PROYECTO.....	66
2.2.1. PRINCIPIO FILOSOFICO DEL CONTROL DEL SISTEMA ELECTRÓNICO.....	66
2.3. DIAGRAMA DE BLOQUES DEL PROYECTO.....	67
2.4. ANÁLISIS DEL SISTEMA.....	67
2.5. SISTEMA DE CONTROL.....	68
2.6. CONEXIONES A LA TARJETA PRINCIPAL.....	68
2.6.1. CONEXIÓN DE LOS OSCILADORES.....	68
2.6.2. CONEXIÓN DEL SENSOR SRF 05.....	69
2.6.3. CONEXIÓN DEL MÓDULO DE VOZ.....	71
2.6.4. CONEXIÓN DEL LCD.....	72
2.7. CARGADOR DE BATERIAS.....	72
2.8. CONEXIÓN DEL REGULADOR DE VOLTAJE (7805A).....	74
2.9. SISTEMA DE REPRODUCCIÓN DE MENSAJES DE VOZ.....	74

2.9.1. INTRODUCCIÓN	74
2.9.2. DESCRIPCIÓN DEL MÓDULO DE VOZ (WT9501M03).....	75
2.9.2.1. DIRECCIONAMIENTO DE LA MEMORIA SD	75
2.9.3. PROCESO DE GRABACIÓN DE MENSAJES	75
2.10. DISEÑO DEL CIRCUITO ESQUEMATICO	77
2.11. DISEÑO DEL CIRCUITO IMPRESO DE LA PLACA.....	79
2.11.1. PASOS PARA LA CREACIÓN DEL CIRCUITO ESQUEMÁTICO.....	79
2.12. REALIZACION DEL CIRCUITO IMPRESO	85
2.13. DESCRIPCION DE LA PLACA PRINCIPAL.....	92
2.14. DESCRIPCION DEL CONTROL ULTRASONIDO.....	93
CAPITULO III	94
SOFTWARE PARA EL SISTEMA DE ALERTA DE OBSTÁCULOS PARA NO VIDENTES.....	94
3.1. INTRODUCCIÓN.....	94
3.2. PROCESOS PARA PROGRAMAR UN MICROCONTROLADOR.....	94
3.3. LENGUAJE DE PROGRAMACION	95
3.3.1 MIKROC™	95
3.3.1.1. Características de MikroC™	95
3.3.2. LENGUAJE DE PROGRAMACION.....	97
3.3.2.1. Estructura de programa.....	97
3.3.2.2. Comentarios.....	99
3.3.2.3. Operadores aritméticos.....	99
3.3.2.4. Operadores de relación	100
3.3.2.5. Operadores de bits.....	100
3.3.2.6. Operadores lógicos.....	101
3.3.2.7. Tipos de variables.....	102
3.3.2.8. Sentencias.....	102
3.3.2.8.1. Sentencias de etiqueta.....	103
3.3.2.8.2. Sentencias de selección.....	103

3.3.2.8.3. Sentencias de iteración.....	104
3.3.2.8.4. Sentencias de salto.....	105
3.4 DIAGRAMAS DE FLUJO DEL PROGRAMA.....	107
3.5. CODIGO DE PROGRAMA PARA EL PIC16F887A.....	111
CAPITULO IV.....	126
PRUEBAS DE FUNCIONAMIENTO DEL PROYECTO.....	126
4.1. INTRODUCCION.....	126
4.2 PRUEBAS DE FUNCIONAMIENTO.....	126
4.2.1. FUNCIONAMIENTO DEL MÓDULO LCD.....	126
4.2.2. FUNCIONAMIENTO DEL SENSOR ULTRASONIDO SRF 05.....	127
4.2.3. FUNCIONAMIENTO DEL MÓDULO DE VOZ.....	128
CAPITULO V.....	130
CONCLUSIONES Y RECOMENDACIONES.....	130
5.1. CONCLUSIONES.....	130
5.2. RECOMENDACIONES.....	131
BIBLIOGRAFIA.....	132
ANEXOS.....	134
ANEXO 1: DESCRIPCION DEL MICROCONTROLADOR PIC16F877A.....	135
ANEXO 2: DESCRIPCION DEL MODULO DE VOZ WT9501M03.....	145
ANEXO 3: DESCRIPCION DEL SENSOR SRF05.....	157
ANEXO 4: MANUAL DE USUARIO.....	162

INDICE DE FIGURAS

Figura 1.1 Bastón Blanco.....	6
Figura 1.2 Eye Stick.....	9
Figura 1.3 Bastón Electrónico para Ciegos.....	10
Figura 1.4 Kinect ayuda para ciegos.....	12
Figura 1.5 Kinect ayuda para ciegos.....	13
Figura 1.6 Una Vara para Ciegos.....	13
Figura 1.7 Una Vara para Ciegos.....	14
Figura 1.8 Visión de murciélago para ciegos.....	14
Figura 1.9 Codificación no auto reloj.....	20
Figura 1.10 Arquitectura Von Neumann.....	24
Figura 1.11 Arquitectura Harvard.....	25
Figura 1.12 Diagrama de bloques del PIC16F877A.....	39
Figura 1.13 Diagrama de pines del PIC16F877A.....	40
Figura 1.14 Memoria de programa del PIC16F877A.....	44
Figura 1.15 Sensores de distancia.....	47
Figura 1.16 Sensores ultrasónicos.....	48
Figura 1.17 Pines del SRF05 en modo 1.....	50
Figura 1.18 Diagrama de tiempos del SRF05 en modo 1.....	50
Figura 1.19 Pines del SRF05 en modo 2.....	51
Figura 1.20 Diagrama de tiempos del SRF05 en modo 2.....	52
Figura 1.21 Haz de radiación del SRF05.....	53
Figura 1.22 Modulo de voz WT9501M03.....	54
Figura 1.23 Pines del módulo WT9501M03.....	55
Figura 1.24 LCD alfanumérico.....	61
Figura 1.25 Esquema de conexiones en modo 8 bits.....	64
Figura 1.26 Esquema de conexiones en modo 4 bits.....	64
FIGURA 1.27 Timing.....	65

Figura 2.1. Diagrama del Proyecto.....	67
Figura 2.2 Análisis del sistema.....	68
Figura 2.3 Conexión del Oscilador tipo XT.....	69
Figura 2.4 Conexión del sensor SRF 05.....	71
Figura 2.5 Conexión del módulo de voz.....	71
Figura 2.6 (Conexión del LCD).....	72
Figura 2.7 Circuito del Cargador de Baterías.....	73
Figura 2.8 Conexión del Regulador de Voltaje.....	74
Figura 2.9 Proceso de grabación de los mensajes de voz.....	75
Figura 2.10 Circuito esquemático del sistema electrónico.....	78
Figura 2.11 Ejecutando Eagle 5.11.0.....	79
Figura 2.12 Ventana del Eagle.....	80
Figura 2.13 Ventana para el diseño de un nuevo esquema.....	80
Figura 2.14 Herramienta Add.....	81
Figura 2.15 Herramienta Wire.....	81
Figura 2.16 Esquema finalizado.....	81
Figura 2.17 Ordenar los elementos.....	82
Figura 2.18 Auto ruteado.....	82
Figura 2.19 Configuración para el auto ruteado.....	83
Figura 2.20 Circuito Board.....	83
Figura 2.21 Configuración del display.....	84
Figura 2.22 Pistas de la placa principal.....	84
Figura 2.23 Configuración de la impresión del circuito.....	85
Figura 2.24 Pista impresa de la placa principal.....	86
Figura 2.25 Pista impresa de la placa de la mano para el sensor.....	86
Figura 2.26 Placa a utilizar.....	86
Figura 2.27 Placa limpia y lista para la transferencia.....	87
Figura 2.28 Placa lista para el planchado.....	87
Figura 2.29 Proceso de planchado.....	88
Figura 2.30 Quemado de la placa con ácido.....	88
Figura 2.31 Placa limpia.....	89

Figura 2.32 Taladro para perforar la placa.....	89
Figura 2.33 Screen impreso.....	90
Figura 2.34 Retirar la lámina.....	90
Figura 2.35 Screen transferido a la placa.....	90
Figura 2.36 Elementos soldados.....	91
Figura 2.37 Placa finalizada y colocados todos sus elementos.....	91
Figura 2.38 Circuito para ser ensamblado.....	92
Figura 2.39 Control de Ultrasonido.....	93
Figura 3.1 Procesos para programar un microcontrolador.....	95
Figura 3.2 Ventana de MikroC.....	97
Figura 3.3 Estructura de un programa en MikroC.....	98
FIGURA 3.4 Diagrama de flujo del sistema de control.....	107
FIGURA 3.5 Diagrama de flujo del sistema de adquisición.....	108
FIGURA 3.6 Diagrama de flujo del sistema de adquisición.....	109
FIGURA 3.7 Diagrama de flujo del sistema de mensajes de voz.....	110
Figura 4.1 Prueba del LCD.....	127
Figura 4.2 Prueba con el flexómetro a 54cm.....	127
Figura 4.3 Controlador Ultrasonido.....	128
Figura A4.1 Partes del módulo principal.....	164
Figura A4.2 Partes del módulo de adquisición.....	166
Figura A4.3 Detección de obstáculos frontales.....	169
Figura A4.4 Detección de huecos y gradas.....	170

INDICE DE TABLAS

Tabla 1.1 Características generales del PIC16F877A.....	37
Tabla 1.2 Características internas del PIC16F877A.....	38
Tabla 1.3 Descripción de los pines del PIC16F877A.....	43
Tabla 1.4 Selección de bancos.....	45
Tabla 1.5 Descripción de pines del módulo WT9501M03.....	56
Tabla 1.6 Función de pines en el modo pulsador.....	56
TABLA 1.7 Protocolo de comunicación usado por el WT9501M03.....	57
TABLA 1.8 Descripción del código de operación.....	58
Tabla 1.9 Pines del LCD alfanumérico.....	63
Tabla 2.1 Conexión del Sensor srf05.....	70
Tabla 2.2 Código USART para la reproducción de mensajes.....	77
Tabla 3.1 Operadores aritméticos.....	100
Tabla 3.2 Operadores de relación.....	100
Tabla 3.3 Operadores de bits.....	101
Tabla 3.4 Operadores lógicos.....	101
Tabla 3.5 Tipo de variables.....	102
Tabla 4.1 Detección de obstáculo del sensor.....	128
Tabla 4.2 Reproducción del mensaje.....	129
Tabla A4.1 Mensajes de audio.....	167

INTRODUCCION

El sistema electrónico de alerta de obstáculos para personas no videntes mediante el uso de un sensor está constituido por el sistema de adquisición formado por el sensor ultrasónico que es el que tiene la tarea de detectar la distancia de los objetos que estén frente a él. Gracias a que el sistema de adquisición se encuentra en una caja pequeña similar a un mando de televisión, se puede realizar lecturas en la dirección que el usuario desee, pero la desventaja es que el sistema no le informará que tipo de obstáculo se encuentra en esa dirección por lo que el usuario se deberá guiar por su intuición, por ejemplo si apunta hacia abajo y toma una lectura y después apunta a la misma altura pero más adelante y la distancia tiene una variación, entonces es más seguro de que tiene un hueco o una grada en frente dependiendo de la variación si aumenta o disminuye respectivamente. Además de que tiene un pulsador en la parte superior el cual sirve para realizar las lecturas de la distancia, caso contrario el sistema estaría realizando lecturas cada 3 s lo que sería molesto para el usuario.

El cerebro del sistema se ubica en una caja de metal que es llevada por el usuario en un bolso tipo canguro, teniendo 2 bolsillos a cada lado del cuerpo, el uno para la batería que alimenta al sistema y el otro para la caja con el cerebro del sistema. En esta caja esta la placa principal que contiene al PIC el cual se encarga de controlar el correcto funcionamiento de todo el sistema, calculando la distancia con los datos entregados por el sistema de adquisición y escogiendo el código correcto que va a enviar al módulo de voz y a su vez enviar los datos necesarios para ver la distancia por el LCD.

El LCD se usó solo con fines de mantenimiento para el técnico que esté a cargo de resolver problemas con el sistema, es por ese que se lo ubicó dentro de la caja y no fuera.

El módulo de voz por su parte se encarga de reproducir los archivos que están almacenados en la tarjeta SD. Dependiendo del código que recepte por parte del PIC

el módulo de voz accederá a la memoria SD y escogerá el número del archivo requerido, este audio se puede escuchar a través de los audífonos incluidos en el proyecto.

RESUMEN

Para la realización del sistema electrónico de alerta de obstáculos para personas no videntes mediante el uso de un sensor se empezó por hacer un estudio del tipo de ayudas a no videntes ya existentes, y lo que nos encontramos fue con sistemas prototipos y en fase de experimentación, que no son comercializados por lo que una persona no vidente no tiene acceso a ellos.

Para el diseño de los diagramas esquemáticos se dividió al sistema en tres partes. La primera formada por la placa principal donde se ubican el microcontrolador y el LCD así como también los conectores para que se acoplen los demás componentes del sistema. La segunda parte es el sistema de adquisición que contiene al sensor y a un pulsador el cual sirve para activar las lecturas de la distancia. Y por último la tercera parte está formada por el cargador de la batería de 6 voltios.

Primero se procedió a hacer las pruebas en el protoboard, luego los cambios requeridos y con su aprobación se pasó a realizar las placas y a soldar los elementos respectivos, afortunadamente no se tuvo ningún inconveniente ya que el circuito funcionaba a la perfección. Por último en esta etapa del proyecto se realizó las cajas para las tres partes del sistema y un canguro modificado para que la placa principal y la batería se las lleve en la cintura del usuario.

El diseño del software representó un gran reto, se probó cada sistema por separado en el siguiente orden: el sensor para lo cual se realizó un programa con el fin de calcular la distancia que se la visualizo en el LCD y fue comparada con la medida de un flexómetro, luego de varias calibraciones se logró detectar la distancia hasta 4 metros. Luego de esto se probó el módulo de voz programando varios códigos para seleccionar ciertos archivos y probar si por los audífonos se escuchaba los mismos, afortunadamente no se tuvo ningún inconveniente. El reto estuvo en unir los dos

programas con lo que tuvimos unos inconvenientes los que fueron exitosamente superados.

Finalmente se procedió a hacer las pruebas del sistema en varios aspectos como el que la medida visualizada en el display sea correcto o que los mensajes de audio sean escogidos correctamente.

CAPÍTULO I.

FUNDAMENTOS TEÓRICOS.

1.1. INTRODUCCIÓN.

Las personas invidentes no se sienten seguras al caminar por si solas o sin algún tipo de ayuda debido a que no pueden ver qué tipo de obstáculos tienen a su alrededor por lo que no interactúan de forma normal con el medio en el que se desenvuelve diariamente.

Esto ha sido motivo para que a través del tiempo se desarrollen diversas herramientas con el fin de que no lleven una vida limitada, es así como se fueron ingeniando herramientas que aprovechan los otros sentidos que le quedan a los invidentes como son el tacto y el oído principalmente.

Estos sentidos han sido aprovechados para que el invidente pueda leer mediante el tacto con la escritura braille, y también pueda caminar por un ambiente conocido con el uso de varias herramientas de guía que aprovechan la propia intuición de la persona.

Las herramientas guía que son más comunes son el perro guía, que es un perro adiestrado para que guíe a la persona a través de obstáculos frontales y en el suelo evitándolos. La otra es el común bastón blanco cuya aparición se remonta luego de la segunda guerra mundial que es cuando surgen las técnicas de orientación y movilidad. En un hospital de Estados Unidos fue donde se llevaba a cabo un programa de rehabilitación para veteranos ciegos, el entonces sargento Richard Hoover (quien se desempeñaba como Director de Rehabilitación Física, Orientación y Recreación) observa que los ciegos se movían con pesados y cortos bastones que les servían como apoyo pero que les prestaban escasos servicios a la hora de anticipar obstáculos. Idea entonces un bastón largo, liviano, y la técnica de uso que

hoy lleva su nombre y que le permitió a los ciegos del mundo desplazarse en forma autónoma y segura.

Pero estas herramientas tienen sus limitaciones como el no saber si hay un obstáculo elevado como las ramas de un árbol por ejemplo, para lo cual se han ido diseñando varios sistemas que aprovechan distintas tecnologías con el fin de que la persona invidente se pueda desenvolver de mejor forma pero cada sistema electrónico tiene sus ventajas y desventajas.

A continuación se presenta una descripción de varias herramientas y sistemas usados y que están en proceso de fabricación y distribución en la actualidad.

1.2. INSTRUMENTOS Y SISTEMAS DE AYUDA A LA MOVILIZACIÓN DE PERSONAS NO VIDENTES.

1.2.1. INSTRUMENTOS DE BAJA TECNOLOGÍA.

1.2.1.1. Bastón blanco.¹

Para la mayoría de la gente, incluso para muchos profesionales, el bastón blanco es un instrumento con el cual las personas ciegas se identifican y que les sirve para detectar obstáculos. Lo que se ignora es que el bastón se utiliza con una técnica determinada, que debe ser adecuado a la altura de la persona, que hay diferentes modelos y que no sólo lo utilizan las personas totalmente ciegas.

Lo conocemos como bastón largo, o de Hoover, o de movilidad. Generalmente se fabrica con tubos de aluminio hueco recubierto con material plástico. En el extremo inferior tiene una puntera metálica recambiable y en el superior una empuñadura que idealmente debe ser de goma para facilitar la toma. Puede ser rígido o plegable. Este último modelo trae en su interior un elástico grueso que posibilita su plegado generalmente en cuatro tramos tal como podemos observar en la figura 1.1. Con respecto a las ventajas y desventajas de uno y otro modelo podemos decir que el

¹http://www.juntadeandalucia.es/averroes/caidv/interedvisual/ftp/fm_personas_c_y_uso_baston.rtf

rígido es más durable y transmite mejor las sensaciones táctiles mientras que el plegable se destaca por su portabilidad siendo ideal para quien no necesita usarlo de forma permanente (por ejemplo quien posee ceguera nocturna).

En cuanto a la medida, debe llegar hasta la apófisis xifoides del esternón, siendo las medidas más comunes 1.05; 1.10; 1.15 y 1.20 metros. Es muy importante respetar la altura apropiada para cada persona ya que un bastón muy corto no permitirá anticipar lo suficiente los obstáculos u obligará a posturas incorrectas con el consiguiente perjuicio físico mientras que un bastón muy largo resultará incómodo y tampoco permitirá la toma correcta.



Figura 1.1 Bastón Blanco.¹

1.2.1.1.1. Técnica de uso.

El entrenamiento en el uso del bastón largo debe ser progresivo, continuado y lo suficientemente prolongado como para cerciorarnos de que la persona lo utiliza en forma adecuada y segura. Existen técnicas pre-bastón que deben enseñarse previamente (como la de rastreo o la del brazo cruzando el cuerpo) pero no nos extenderemos en ellas por no ser objetivo del presente trabajo. Las técnicas con bastón son básicamente dos:

- Técnica diagonal.
- Técnica rítmica.

1.2.1.1.1.1. Técnica diagonal.

Es la que se utiliza para deambular en interiores desconocidos. Consiste en colocar el bastón en forma diagonal, delante del cuerpo a modo de parachoques y no de explorador. Se toma colocando la parte interna de la muñeca hacia abajo, con el dedo índice extendido y colocando el bastón a unos 30° del cuerpo de manera que la punta quede (sin tocar el suelo) delante del pie del lado contrario al que sostiene el bastón. Es la técnica que se usa por ejemplo para circular en un shopping, en un edificio público, en un hospital.

1.2.1.1.1.2. Técnica rítmica.

Es la que le permitirá a la persona desplazarse en forma segura e independiente en exteriores conocidos y desconocidos. Consiste en mover en forma rítmica el bastón delante del cuerpo mientras se camina, con el fin de detectar obstáculos en el suelo.

Para ello es importante que:

- La toma se efectúe de forma correcta, es decir con la muñeca apoyada en el centro del cuerpo, el dedo índice en extensión (a fin de posibilitar una buena percepción táctil e imprimirle direccionalidad al movimiento), ubicando el bastón extendido hacia delante de modo que la punta quede delante del pie que comenzará la marcha.
- La posición del brazo sea la adecuada, es decir que esté con el hombro relajado en posición primaria (sin que se extienda hacia delante ni hacia atrás, ni esté elevado ni caído), el brazo al costado y el antebrazo apoyado delante del cuerpo formando un ángulo de 90° con respecto al brazo de forma de posibilitar la correcta toma.
- El movimiento de la muñeca se realice en forma recta de derecha a izquierda evitando movimientos circulares que imprimirían al bastón una dirección incorrecta.

- El arco sea el adecuado, es decir levemente más ancho que el ancho del cuerpo de modo que al moverse el bastón anticipe en forma efectiva el sitio en que la persona va a pisar. El bastón debe tocar el suelo en los extremos derecho e izquierdo del arco levantándose levemente del piso (en el caso de la técnica de dos puntos) o deslizándose (en el caso de la técnica de contacto constante).
- El ritmo se realice de modo que el bastón toque el suelo del lado derecho mientras que el pie izquierdo se adelanta y viceversa.

Sólo si se tienen en cuenta todos estos aspectos a la hora de enseñar la técnica, la marcha será cómoda y segura. Es importante corregir errores de entrada ya que si se automatiza el movimiento de forma incorrecta provocará vicios posturales que repercutirán negativamente en la salud física de la persona y en su deambular. Ahora bien, en las personas adultas, suele ocurrir que cuando se inicia la enseñanza de la técnica rítmica se evidencien problemas que quizás hayan pasado desapercibidos anteriormente como por ejemplo, imposibilidad de caminar en línea recta, falta de equilibrio, giros incorrectos, etc.

Es importante investigar si son la lógica y pasajera consecuencia de la falta de visión, en cuyo caso revertirán con simples ejercicios o si se trata de problemas neurológicos en donde se deberá hacer las consultas médicas correspondientes siendo la intervención de un kinesiólogo o psicomotricista de mucha ayuda.

1.2.2. SISTEMAS ELECTRONICOS DE AYUDA A LA MOVILIDAD DE PERSONAS INVIDENTES.

Actualmente se están desarrollando varios proyectos cada uno de estos utiliza una tecnología diferente pero todos tienen como finalidad ayudar a la movilidad de los invidentes en el entorno en que se desenvuelven.

A continuación se van a mencionar varios de los proyectos que han sido, son y prometen ser de gran utilidad.

1.2.2.1. Eye stick, bastón con sensor para ayudar a los invidentes².

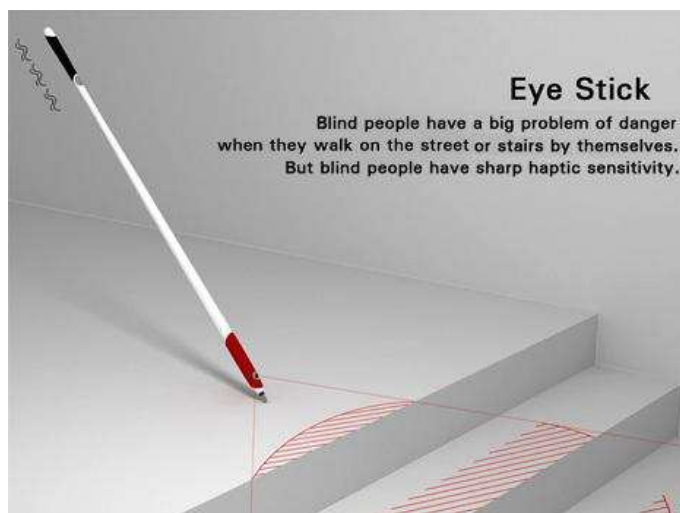


Figura 1.2 Eye Stick².

Seguramente las personas con discapacidades visuales, que utilizan bastones para sentir el tipo de camino que se encuentre frente a ellos son muy útiles. Sin embargo una pequeña ayuda no está demás, pues eso es lo que hace el Eye Stick que cuenta con un sensor en la parte inferior, que permite determinar qué tipo de terreno es el que se encuentra a continuación, emitiendo un sonido y pequeñas vibraciones para que la persona invidente pueda detenerse. Vemos en la figura 1.2 como se usa este bastón al encontrarnos frente a unas escaleras.

El sensor es capaz de detectar escaleras, banquetas, objetos grandes e incluso coladeras, ayudando a la persona a saber que hay algo que podría ser peligroso o que no permite continuar.

Este tipo de dispositivos están siendo muy bien aprovechados en objetos para la gente que tiene algún tipo de discapacidad, aunque no se ha invertido mucho tiempo en la investigación.

²<http://espegizmo.com/2009/09/08/general/eye-stick-baston-con-sensor-para-ayudar-a-los-invidentes/>

1.2.2.2. Bastón electrónico para ciegos.



Figura 1.3 Bastón Electrónico para Ciegos.

En la figura 1.3 vemos al bastón electrónico para ciegos, que mide la distancia de los objetos mediante rayos laser y transmite la información correspondiente al usuario a través de sonidos o vibraciones emitidos por un pequeño dispositivo del tamaño de un mando de televisión que se lleva adherido a la mano, comienza a implantarse después de haber pasado dos años en fase de experimentación.

El bastón electrónico tiene externamente la misma apariencia que el bastón blanco alargado empleado por los ciegos para orientarse en sus desplazamientos por entornos abiertos, con la diferencia de que cuanto más se aproxima el usuario a un obstáculo, más intensa es la señal emitida por el aparato.

El bastón electrónico no sustituye al tradicional, sino que más bien lo complementa, ya que potencia la capacidad de esta herramienta para evitar los muros, anticipar el comienzo de la acera o descubrir un bache en el firme, e incluso señalar a los videntes la proximidad de una persona ciega.

Hay dos modelos del bastón electrónico: uno denominado Tom Puce (pequeño), que detecta objetos a cuatro metros, alerta mediante vibraciones y cuesta 762 euros. El otro modelo, más sofisticado y caro (2.286 euros), se llama Teletacto y describe el

espacio con mayor amplitud (hasta 15 metros), ya que el barrido del láser permite detectar los perfiles y reconocer las formas.

1.2.2.2.1. Ocho años de investigación.

El bastón ha sido creado por el físico e investigador del CNRS y de la Universidad Paris-Sud/Orsay, René Farsi, quien ha dedicado ocho años de su vida a perfeccionar esta tecnología, probada ya con 60 invidentes franceses, y que continúa desarrollándose con diferentes variables nuevas.

El bastón electrónico no es de fácil manejo, ya que necesita un período de formación que cuesta 1.800 euros, que se suman al precio del producto. Lo más complicado es que los usuarios aprendan a interpretar las señales emitidas por el dispositivo y a construir referencias a partir de estos símbolos.

La eficacia comprobada en el período experimental es del 60%, pero el hecho de que el 40% de los que lo prueban no consigan orientarse no deja de ser un obstáculo para su plena implantación social.

De ahí que los objetivos de sus creadores sean todavía modestos: en Francia hay 300.000 ciegos de los que 50.000 utilizan bastón. De ellos, 10.000 personas no pueden prescindir de este apoyo para su vida cotidiana y lo que pretenden los inventores del bastón electrónico es que en el plazo de diez años 5.000 ciegos se beneficien para la adquisición de un bastón electrónico por persona.

1.2.2.2.2. Apoyo GPS.

Estas previsiones son de por sí bastante elocuentes del relativo potencial alcance social del bastón electrónico. La mayoría de los usuarios prefiere el modelo más simple y económico, pero los más capaces se encuentran un completo soporte a su movilidad: consiguen que el oído supla a la visión en la orientación sobre el entorno, permitiendo al usuario deducir los comportamientos de locomoción más adecuados.

El proyecto sin embargo es más ambicioso, ya que pretende incorporar al bastón electrónico el soporte de localización geográfica vía satélite (GPS), con la finalidad de que el usuario pueda tener todas las referencias de un entorno para orientarse, de la misma forma que hoy lo consigue un conductor para llegar a un destino desconocido.

1.2.2.3. Kinect ayuda para ciegos³.

NAVI [Navigational Aids for the Visually Impaired], es el nombre que se le dio a un sistema que sirve como ayuda de guía para las personas ciegas.



Figura 1.4 Kinect ayuda para ciegos³.

Esto hará que el Sistema NAVI, al ir caminando te diga que obstáculos tienes al frente. Funciona [como vemos en la figura 1.4], colocando el famoso Kinect a un casco, conectado por un dispositivo Bluetooth y un cinturón vibro-táctil Arduino, y la mochila donde va cargado el sistema NAVI. Te dirá el sistema mediante voz lo que tienes al frente, incluso puede darte consejos de que dirección tomar.

Los responsables son Michael Zollner y Stephan Huber en la Universidad Konstanz, en Alemania.

³<http://beatgeek.net/kinect-ayuda-para-ciegos/>

Zollner y Huber, dicen que hasta el momento es muy pesado, ya que tener que llevar Kinect prácticamente a la cabeza, cables y de más, es fatigado y estorboso, pero seguirán trabajando en NAVI, para más adelante poder optimizarlo y que sea de gran ayuda.



Figura 1.5 Kinect ayuda para ciegos³.

1.2.2.4. Una vara para ciegos⁴.



Figura 1.6 Vara para Ciegos⁴.

Cuando la ciencia hace aportes a enfermedades o padecimientos de algún tipo es una doble celebración, y en este caso tenemos a la tecnología ayudando a los ciegos a moverse con más facilidad y evitando los obstáculos. En la figura 1.6 vemos a la vara para ciegos como un aparato parecido a un mando de televisión.

⁴<http://www.impresionante.net/10-03-2008/general/una-vara-para-ciegos>

Diseñado por Woo Jin Han esta vara para ciegos hace cierta referencia al clásico bastón blanco, sólo que un poco de tecnología que permite detectar los obstáculos sin necesidad de golpearlos.

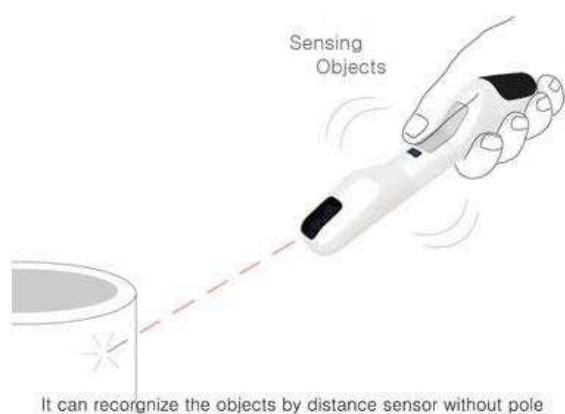


Figura 1.7 Una Vara para Ciegos⁴.

El tubo tiene sensores incorporados que miden la distancia de los objetos y avisan por medio de vibraciones al portador de su cercanía, para los evite tal como podemos observar en la figura 1.7.

1.2.2.5. Visión de murciélago para ciegos⁵.



Figura 1.8 Visión de murciélago para ciegos⁵.

⁵<http://www.neoteo.com/vision-de-murcielago-para-ciegos-14686>

Se trata de un invento que va a permitir mayor independencia y movilidad a los ciegos. Desde la Universidad de Valencia llega un aparato que dispone de rayos láser, video y hasta de GPS para situar y ubicar a los invidentes mediante sonidos que le descubren donde existen obstáculos. Y de paso, pueden entrar en una obra sin infringir la ley de prevención de riesgos laborales.

En el futuro, aparte de entretenerse en diseñar inventos imposibles, también ayuda a los más discriminados y pone a su disposición lo último en tecnología para hacerles la vida más llevadera. En este caso, el milagro viene de la mano de la Universidad Politécnica de Valencia, que coordina el desarrollo de los prototipos que podrán ser utilizados de forma masiva en 2 o 3 años. El sistema se basa en el uso de varias tecnologías bien diferenciadas cuyo fin último consiste en posicionar al invidente y situarlo en el espacio de tal modo que sepa en todo momento donde se encuentran los obstáculos con los que se enfrenta. El proyecto CASBLIP (acrónimo en inglés de Sistema Cognitivo de Ayuda a Personas Ciegas) combina en sus dos modalidades 4 tecnologías modificadas y mejoradas: láser infrarrojo, imágenes de vídeo y dos aplicaciones de GPS. En total participan en el prototipo siete instituciones europeas: el instituto Francesco Cavazza y el DBSV -dos grandes organizaciones de ciegos de Italia y Alemania-, Siemens y las universidades de Bristol (Inglaterra) y La Marche, de Ancona (Italia), además de la canaria de La Laguna y la Politécnica de Valencia.

Existen dos prototipos diferentes que conforman el sistema integral de detección de objetos. Por un lado se encuentran las gafas M1 que se observan en la figura 1.8, que emiten un rayo láser que permite localizar los objetos a 5 metros. Han sido diseñadas por Siemens (a partir de los usados en coches). Al mirar hacia una zona, mide las distancias al objeto que haya en el camino. La principal novedad es el sistema de traducción de esos objetos en tres dimensiones, captados por el láser, en 64 señales acústicas diferentes que hacen que la persona ciega pueda ubicar esos objetos a través de los auriculares. El sistema acústico "engaña al cerebro", de manera que los sonidos son recibidos como si vinieran del exterior. Es como si

cualquier persona que no tiene problemas de visión cerrara los ojos e intentara situar dónde ha caído una moneda.

Por otro lado tenemos el casco M2, que parece de albañil y que incorpora sendas cámaras con sistema de estereovisión, posee un alcance de 15 metros. Detecta objetos en movimiento (sólo si hacen ruido) y avisa al invidente de que se está acercando o alejando. Por ejemplo, si viene un coche será detectado pero si aparece una bicicleta silenciosa no. Este sistema además ayuda a los ciegos en "zonas complejas con pocos espacios", sin huecos. Le muestra "un pasillo libre" en un área conflictiva. Como si cualquiera, en una habitación oscura, pudiera notar dónde se abre una puerta.

Los usuarios de esta nueva tecnología se mostraban poco propicios a probarla porque alegaban que el sonido de los auriculares no les dejaría escuchar otros sonidos, que le son fundamentales para su supervivencia, pero luego, con algo de entrenamiento, pudieron comprobar que apenas suponía un escollo. Ahora muchos invidentes han probado el invento y se declaran sumamente satisfechos con los resultados.

1.3. PROTOCOLO DE COMUNICACIÓN SERIAL⁶.

1.3.1. DEFINICION DE COMUNICACIÓN SERIAL.

La comunicación serial es un protocolo muy común (no hay que confundirlo con el Bus Serial de Comunicación, o USB) para comunicación entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. La mayoría de las computadoras incluyen dos puertos seriales RS-232. La comunicación serial es también un protocolo común utilizado por varios dispositivos para instrumentación; existen varios dispositivos compatibles con GPIB que incluyen un puerto RS-232. Además, la comunicación serial puede ser utilizada para adquisición de datos si se usa en conjunto con un dispositivo remoto de muestreo.

⁶<http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>

El concepto de comunicación serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias. Por ejemplo, la especificación *IEEE 488* para la comunicación en paralelo determina que el largo del cable para el equipo no puede ser mayor a 20 metros, con no más de 2 metros entre cualesquier dos dispositivos; por el otro lado, utilizando comunicación serial el largo del cable puede llegar a los 1200 metros.

Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

- **Velocidad de transmisión (baudrate):** Indica el número de bits por segundo que se transfieren, y se mide en baudios (bauds). Por ejemplo, 300 baudios representa 300 bits por segundo. Cuando se hace referencia a los ciclos de reloj se está hablando de la velocidad de transmisión. Por ejemplo, si el protocolo hace una llamada a 4800 ciclos de reloj, entonces el reloj está corriendo a 4800 Hz, lo que significa que el puerto serial está muestreando las líneas de transmisión a 4800 Hz.
- **Bits de datos:** Se refiere a la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits. Las cantidades más comunes de bits por paquete son 5, 7 y 8 bits. El número de bits que se envía depende en el tipo de información que se transfiere. Por ejemplo, el ASCII estándar tiene un rango de 0 a 127, es decir, utiliza 7 bits; para ASCII extendido es de 0 a 255, lo que utiliza 8 bits. Si el tipo de datos que se está transfiriendo es texto simple (ASCII estándar), entonces es suficiente con utilizar 7 bits por paquete para la comunicación. Un paquete se refiere a una transferencia de byte, incluyendo los bits de inicio/parada, bits de datos, y paridad. Debido a que el número actual de bits depende en el protocolo que se seleccione, el término paquete se usará para referirse a todos los casos.

- **Bits de parada:** Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits. Debido a la manera como se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj, es posible que los dos dispositivos no estén sincronizados. Por lo tanto, los bits de parada no sólo indican el fin de la transmisión sino además dan un margen de tolerancia para esa diferencia de los relojes. Mientras más bits de parada se usen, mayor será la tolerancia a la sincronía de los relojes, sin embargo la transmisión será más lenta.
- **Paridad:** Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible. Para paridad par e impar, el puerto serial fijará el bit de paridad (el último bit después de los bits de datos) a un valor para asegurarse que la transmisión tenga un número par o impar de bits en estado alto lógico. Por ejemplo, si la información a transmitir es 011 y la paridad es par, el bit de paridad sería 0 para mantener el número de bits en estado alto lógico como par. Si la paridad seleccionada fuera impar, entonces el bit de paridad sería 1, para tener 3 bits en estado alto lógico. La paridad marcada y espaciada en realidad no verifican el estado de los bits de datos; simplemente fija el bit de paridad en estado lógico alto para la marcada, y en estado lógico bajo para la espaciada. Esto permite al dispositivo receptor conocer de antemano el estado de un bit, lo que serviría para determinar si hay ruido que esté afectando de manera negativa la transmisión de los datos, o si los relojes de los dispositivos no están sincronizados.

A diferencia de la comunicación en paralelo, en la comunicación en serie se hace necesario establecer métodos de sincronización para evitar la interpretación errónea de los datos transmitidos. Para ilustrar esto consideremos la siguiente información en serie:

...0100110001001100100...

Esta información puede interpretarse de diversas maneras, (aún si se recibe a la velocidad adecuada) dependiendo del punto de inicio de separación de datos, por ejemplo, una posible interpretación sería como sigue:

... 0100110001001100100...

Que interpretado como códigos ASCII corresponde a los caracteres '1' y '2'. Sin embargo, otra posible interpretación es:

... 0100110001001100100

Que corresponde a los caracteres 'b' y 'd'.

Una manera de resolver el problema anterior es la sincronización de bits que puede realizarse por varios métodos:

- Enviar por una línea adicional una señal de reloj que indique el centro de las celdas de bits en la línea de datos (datos no - auto reloj).
- Enviar con cada bit y por la misma línea de datos información que permita extraer la señal de reloj (datos auto reloj).
- Lograr mediante alguna estrategia que los relojes de transmisión y de recepción se mantengan en fase continuamente.

1.3.1.1. Codificación no auto reloj.

En la figura 1.15 se muestran las tres codificaciones de una línea de datos:

- **RZ.-** Una celda de bit es 1 si contiene un impulso positivo y un 0 si no lo contiene.
- **NRZ.-** La celda contiene un 1 o 0 de acuerdo al nivel de la señal (constante) en la celda.
- **NRZI.-** La celda de bit contiene un 1 si hay una transición y un 0 si no la hay.

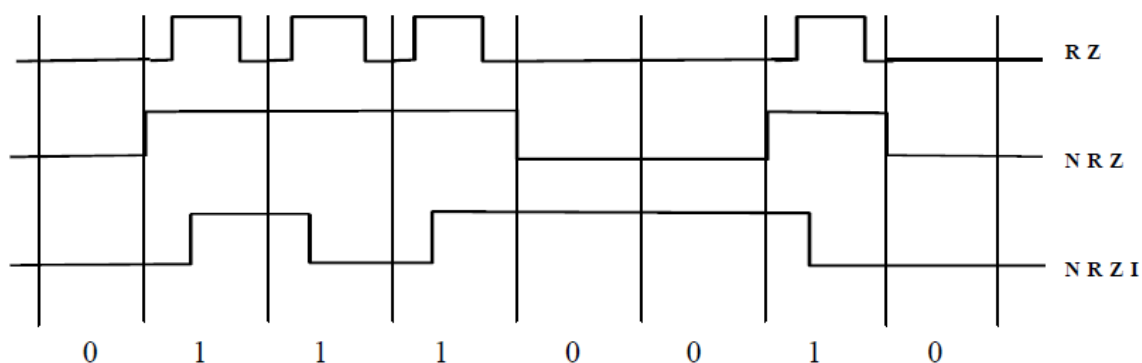


Figura 1.9 Codificación no auto reloj.

Como puede verse en la figura 1.9, en estos sistemas (RZ, NRZ y NRZI) las secuencias de ceros no contienen ninguna transición que permita ubicar la situación de las celdas de bit. De hecho, el formato NRZ no la contiene ni en los unos.

1.3.1.2. Codificación auto reloj.

Algunos métodos que contienen en la misma línea de datos información adicional para determinar la velocidad del reloj a costa de disminuir la cantidad de información útil a la mitad que los métodos no-auto reloj.

1.3.1.3. Método asíncrono.

Cada carácter va señalizado mediante dos bits: un bit de inicio y un bit de parada, estos dos bits permiten al receptor reconocer el inicio y el final de cada carácter. La especificación RS404 de EIA (Electronic Industries Association) define las características del método asíncrono para transmisión en serie de acuerdo a las siguientes reglas:

- Cuando no se envían datos la línea debe mantenerse en estado 1.
- Cuando se va a mandar un carácter se envía primero un bit de inicio de valor 0.

- A continuación se envían todos los bits del carácter a transmitir al ritmo marcado por el reloj de transmisión.
- Después del último bit del carácter enviado se envía un bit de parada de valor 1.

1.3.1.4. Método síncrono.

Cada mensaje o bloque de transmisión va precedido de unos caracteres de sincronismo. Así, cuando el receptor identifica una configuración de bits igual a la de los caracteres de sincronismo da por detectado el inicio y el tamaño de los datos.

Observación: Para el usuario de un microcontrolador que posee una USART o sistema similar la manera detallada como el sistema logra establecer la comunicación resulta transparente a él, ya que sólo tiene que configurar el protocolo del transmisor y del receptor para que estos logren la comunicación adecuada, es decir, el usuario usualmente sólo debe configurar:

- Tipo de comunicación (síncrona o asíncrona).
- Velocidad de transmisión en Baudios (bits por segundo)
- Longitud de los datos.
- Bits de inicio y de parada, bits de paridad, etc.

1.4 MICROCONTROLADORES.

1.4.1. INTRODUCCIÓN⁷.

Los microcontroladores son computadores digitales integrados en un chip que cuentan con un microprocesador o unidad de procesamiento central (CPU), una memoria para almacenar el programa, una memoria para almacenar datos y puertos de entrada salida. A diferencia de los microprocesadores de propósito general, como

⁷http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf

los que se usan en los computadores PC, los microcontroladores son unidades autosuficientes y más económicas.

El funcionamiento de los microcontroladores está determinado por el programa almacenado en su memoria. Este puede escribirse en distintos lenguajes de programación. Además, la mayoría de los microcontroladores actuales pueden reprogramarse repetidas veces.

Por las características mencionadas y su alta flexibilidad, los microcontroladores son ampliamente utilizados como el cerebro de una gran variedad de sistemas embebidos que controlan máquinas, componentes de sistemas complejos, como aplicaciones industriales de automatización y robótica, domótica, equipos médicos, sistemas aeroespaciales, e incluso dispositivos de la vida diaria como automóviles, hornos de microondas, teléfonos y televisores.

Frecuentemente se emplea la notación μC o las siglas MCU (por microcontroller unit para referirse a los microcontroladores).

1.4.2. ¿QUÉ ES UN MICROCONTROLADOR?⁸

Un microcontrolador es un circuito integrado que contiene toda la estructura (arquitectura) de un microcomputador, o sea CPU, RAM, ROM y circuitos de entrada y salida. Los resultados de tipo práctico, que pueden lograrse a partir de estos elementos, son sorprendentes.

Algunos microcontroladores más especializados poseen además convertidores análogos digital, temporizadores, contadores y un sistema para permitir la comunicación en serie y en paralelo.

Al principio, los controladores estaban formados exclusivamente por componentes discretos. Más tarde, se emplearon procesadores rodeados de memorias, circuitos de E/S,... sobre una placa de circuito impreso (PCB).

⁸<http://ingenio-upp.blogdiario.com/>

Actualmente, los controladores integran todos los dispositivos antes mencionados en un pequeño chip. Esto es lo que hoy conocemos con el nombre de microcontrolador.

1.4.3. APLICACIONES DE LOS MICROCONTROLADORES⁹.

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo.

Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc. Y otras aplicaciones con las que seguramente no estaremos tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema. Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

1.4.4. TIPOS DE ARQUITECTURAS DE MICROCONTROLADORES⁹.

1.4.4.1. Arquitectura Von Neumann.

La arquitectura tradicional de computadoras y microprocesadores está basada en la arquitectura Von Neumann, en la cual la unidad central de proceso (CPU), está conectada a una memoria única donde se guardan las instrucciones del programa y los datos tal como se ilustra en la figura 1.10. El tamaño de la unidad de datos o

⁹<http://www.monografias.com/trabajos12/microco/microco.shtml>

instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU. Así un microprocesador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits (bytes) de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria. Y el tener un único bus hace que el microprocesador sea más lento en su respuesta, ya que no puede buscar en memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior. Resumiendo todo lo anterior, las principales limitaciones que nos encontramos con la arquitectura Von Neumann son:

La limitación de la longitud de las instrucciones por el bus de datos, que hace que el microprocesador tenga que realizar varios accesos a memoria para buscar instrucciones complejas.

La limitación de la velocidad de operación a causa del bus único para datos e instrucciones que no deja acceder simultáneamente a unos y otras, lo cual impide superponer ambos tiempos de acceso.

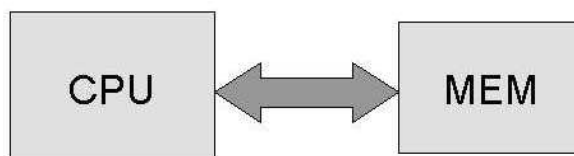


Figura 1.10 Arquitectura Von Neumann⁹

1.4.4.2. Arquitectura Harvard.

La arquitectura Harvard tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes tal como se observa en la figura 1.11. Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y en la otra se almacena datos (Memoria de Datos). Ambos buses son totalmente independientes y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones.

Reducido, o RISC (Reduced Instrucción Set Computer), el set de instrucciones y el bus de memoria de programa pueden diseñarse de tal manera que todas las instrucciones tengan una sola posición de memoria de programa de longitud. Además, al ser los buses independientes, la CPU puede acceder a los datos para completar la ejecución de una instrucción, y al mismo tiempo leer la siguiente instrucción a ejecutar.

Ventajas de esta arquitectura:

- El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.
- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación.

Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontraran físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

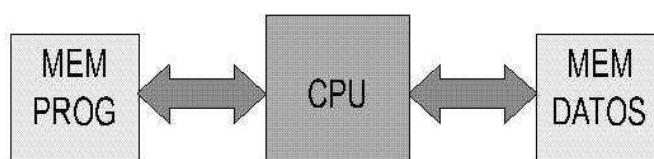


Figura 1.11 Arquitectura Harvard⁹

1.4.5. ORGANIZACIÓN INTERNA⁹.

A continuación pasamos a describir los elementos más comunes en todo tipo de microcontroladores y sistemas.

1.4.5.1. El procesador.

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software. Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado. Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

1.4.5.1.1. CISC.

Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros, es decir, que si las tuviésemos que implementar con instrucciones básicas, acabaríamos con dolor de cabeza.

1.4.5.1.2. RISC.

Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

1.4.5.1.3. SISC.

En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es específico, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

1.4.5.2. Tipos de memorias.

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

No existen sistemas de almacenamiento masivo como disco duro o disquetes. Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero los diseñadores con microcontroladores trabajan con capacidades

de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes.

1.4.5.3. Puertos de entrada y salida.

Los puertos de Entrada y Salida (E/S) permiten comunicar al procesador con el mundo exterior, a través de interfaces, o con otros dispositivos. Estos puertos son la principal utilidad de las patas o pines de un microprocesador. Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

1.4.5.4. Reloj principal.

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Esta señal del reloj es el motor del sistema y la que hace que el programa y los contadores avancen.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía y de calor generado.

1.4.6. RECURSOS ESPECIALES.

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para

aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el costo, el hardware y el software. Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o Timers.
- Perro guardián o Watchdog.
- Protección ante fallo de alimentación o Brownout.
- Estado de reposo o de bajo consumo (Sleep mode).
- Conversor A/D (Analógico - Digital).
- Conversor D/A (Digital - Analógico).
- Comparador analógico.
- Modulador de anchura de impulsos o PWM (Pulse Wide Modulation).
- Puertas de E/S digitales.
- Puertas de comunicación.

A continuación pasamos a ver con un poco más de detalle cada uno de ellos.

1.4.6.1. Temporizadores o timers.

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso. Cuando se desean contar acontecimientos que se

materializan por cambios de nivel o flancos en algunos pines del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

1.4.6.2. Perro guardián o watchdog.

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicia el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro Guardián consiste en un contador que, cuando llega al máximo, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que resetee al Perro Guardián de vez en cuando antes de que provoque el reset. Si falla el programa o se bloquea (si cae en bucle infinito), no se refrescará al Perro guardián y, al completar su temporización, provocará el reset del sistema.

1.4.6.3. Protección ante fallo de alimentación o brownout.

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo (brownout). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor. Esto es muy útil para evitar datos erróneos por transiciones y ruidos en la línea de alimentación.

1.4.6.4. Estado de reposo o de bajo consumo.

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj

principal y se congelan sus circuitos asociados, quedando sumido en un profundo sueño. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo. Para hacernos una idea, esta función es parecida a la opción de Suspend en el menú para apagar el equipo (en aquellos PCs con administración avanzada de energía).

1.4.6.5. Conversor A/D (CAD).

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde los pines del circuito integrado.

1.4.6.6. Conversor D/A (CDA).

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por uno de los pines del chip. Existen muchos circuitos que trabajan con señales analógicas.

1.4.6.7. Comparador analógico.

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de los pines de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra. También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

1.4.6.8. Modulador de anchura de impulsos o PWM.

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

1.4.6.9. Puertos digitales de E/S.

Todos los microcontroladores destinan parte de su puerto a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos.

Las líneas digitales de las Puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

1.4.6.10. Puertos de comunicación.

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- UART, adaptador de comunicación serie asíncrona. (Ej. Puerto Serie)
- USART, adaptador de comunicación serie síncrona y asíncrona.
- Puerto paralelo esclavo para poder conectarse con los buses de otros microprocesadores.
- USB (Universal Serial Bus), que es un moderno bus serie para los PC.
- Bus I2C, que es un interfaz serie de dos hilos desarrollado por Philips.
- CAN (Controller Area Network), para permitir la adaptación con redes de conexión multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU se usa el J1850.

Tanto el I2C en televisores, como el Bus CAN en automóviles, fueron diseñados para simplificar la circuitería que supone un bus paralelo de 8 líneas dentro de un

televisor, así como para librar de la carga que supone una cantidad ingente de cables en un vehículo.

1.4.6.10.1 COMUNICACIÓN USART¹⁰.

USART es el acrónimo de Universal Synchronous/Asynchronous Receiver Transmitter, que traducido al español viene a ser algo parecido a Transmisor y Receptor Sincrónico/Asincrónico Universal.

Se trata de un periférico para la transmisión de datos en formato serie, utilizando técnicas de transmisión sincrónica o asincrónica, según se configure el periférico.

Este periférico generalmente se confunde con algunos de los estándares de comunicación que lo utilizan para la interconexión entre equipos terminales de datos (ETD) y equipos de circuito de datos (DCE), dentro de estos estándares el más popular es el EIA-232, conocido también como RS-232; se utiliza para la interconexión mediante otros estándares como el RS-485 y el RS-422.

Sus salidas, generalmente son del tipo TTL, aunque actualmente también se pueden encontrar otros niveles lógicos dentro de la gama de valores aceptados por el estándar IEEE, para sistemas digitales. Para comunicar dispositivos distintos, mediante cables de conexión, es obligatorio utilizar, por ejemplo, un MAX232 o MAX485 para adaptar los niveles de tensión a los utilizados por RS-232 o RS-485.

La característica más destacable de este tipo de periféricos es que utiliza solamente dos terminales para el envío y recepción de datos, en cualquiera de los dos modos de trabajo.

- En el caso de las comunicaciones sincrónicas uno de los terminales se comporta como reloj (CLK) y el otro como datos (DT). Este tipo de comunicaciones generalmente demanda el uso de más E/S del dispositivo, por ejemplo un microcontrolador, para gestionar el uso del canal de comunicaciones, la ventaja

¹⁰<http://www.ucontrol.com.ar/wiki/index.php?title=USART>

que tiene radica en que no se requiere realizar una configuración previa de los dispositivos conectados al medio de comunicación debido a que la señal de sincronismo viaja por el medio de comunicación.

- Para las comunicaciones asincrónicas, se destina un terminal a la transmisión (Tx) y otro a la recepción (Rx), en este caso el sincronismo se hace dentro de cada equipo y la interfaz solo define el uso de un bit de start y otro de stop, para indicar el inicio y fin de transmisión de un byte, es por eso que todos los equipos interconectados deben estar configurados para el mismo bit-rate. Las ventajas más importantes de este modo de comunicación radica en que no se requiere destinar más entradas salidas a completar algunas interfaces como la RS-232, en su forma más reducida y en que se puede transmitir y recibir al mismo tiempo con un solo periférico.

1.4.7. LA FAMILIA DE LOS MICROCONTROLADORES PIC¹¹.

1.4.7.1. Introducción.

Los PIC son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument.

El nombre actual no es un acrónimo. En realidad, el nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller (controlador de interfaz periférico).

El PIC original se diseñó para ser usado con la nueva CPU de 16 bits CP16000. Siendo en general una buena CPU, ésta tenía malas prestaciones de E/S, y el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de E/S a la CPU. El PIC utilizaba micro código simple almacenado en ROM para realizar estas tareas; y aunque el término no se usaba por aquel entonces, se trata de un diseño RISC que ejecuta una instrucción cada 4 ciclos del oscilador.

¹¹http://es.wikipedia.org/wiki/Microcontrolador_PIC

En 1985 la división de microelectrónica de General Instrument se separa como compañía independiente que es incorporada como filial (el 14 de diciembre de 1987 cambia el nombre a Microchip Technology y en 1989 es adquirida por un grupo de inversores) y el nuevo propietario canceló casi todos los desarrollos, que para esas fechas la mayoría estaban obsoletos. El PIC, sin embargo, se mejoró con EPROM para conseguir un controlador de canal programable. Hoy en día multitud de PICs vienen con varios periféricos incluidos (módulos de comunicación serie, UARTs, núcleos de control de motores, etc.) y con memoria de programa desde 512 a 32.000 palabras (una palabra corresponde a una instrucción en lenguaje ensamblador, y puede ser 12, 14 o 16 bits, dependiendo de la familia específica de PICmicro).

1.4.7.2. PIC16F877A.

1.4.7.2.1. Introducción.

Microcontroladores PIC16CXX/FXX de Microchip Estos micros pertenecen a la gama media y disponen de un set de 35 instrucciones, por eso lo llaman de tipo RISC (Reduced Instruction Set Computer) en español sería "Computador con Set de Instrucciones Reducido" pocas instrucciones pero muy poderosas, otras son de tipo CISC (Complex Instruction Set Computer - Computador con Set de Instrucciones Complejo), demasiadas instrucciones, y lo peor, difíciles de recordar.

Esta familia de microcontroladores se divide en tres rangos según la capacidad de los microcontroladores. El más bajo lo compone la familia 16C5X. El rango medio lo componen las familias 16C6X/ 7X/ 8X, algunos con conversores A/D, comparadores, interrupciones, etc. La familia de rango superior lo componen los 17CXX.

EL Pic16F877 pertenece a la gama media de Microcontroladores donde encontraremos modelos desde 18 hasta 68 Pines su repertorio de instrucciones es de 35, de 14 bits cada una (Todas las instrucciones son de un solo ciclo con excepción de las ramificaciones del programa que son de dos ciclos). Disponen de interrupciones y una pila de 8 niveles que permite el anidamiento de subrutinas. Se le

puede encontrar en un tipo de encapsulado de 40 pines (ò patas) PDIP el cual es el más utilizado para su utilización didáctica en proyectos escolares.

Cuenta con una arquitectura Harvard en la que son independientes la memoria de instrucciones y la memoria de datos y cada una dispone de su propio sistema de buses para el acceso.

Posee Rango amplio de tensiones de funcionamiento:

- Comercial: 2.0 a 5.5 volts.
- Industrial: 2.0 a 5.5 volts.

Consumo muy bajo:

- < 2 mA típico a 5 volt, 4MHz.
- 15 μ A típico a 2 volt, 32KHz.
- >0.5 μ A típico de corriente en reposo a 2 volts.

1.4.7.2.2. Características.

1.4.7.2.2.1. Características generales.

Sus características más importantes se encuentran en la tabla 1.1:

CARACTERÍSTICAS	16F877A
Frecuencia máxima	DX-20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos E/S	A,B,C,D,E
Número de pines	40
Interrupciones	15
Timers	3

Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones paralelo	PSP
Modulo Analógico a Digital de 10 bit	8 canales de entrada
Juego de instrucciones	35 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard
CPU	Risc
Módulos Comparador/comparador/Pwm	2

Tabla 1.1 Características generales del PIC16F877A.

1.4.7.2.2.2. Características internas.

En la tabla 11.2 encontramos las características internas de la familia de los PIC clasificadas en Procesador, Memoria y Periféricos.

Procesador	Descripción	Características
Procesador	Núcleo	RISC, Arq. Harvard, 20 MHz, 5 MIPS
	Interrupciones	14 fuentes posibles de interrupción
	Reloj	0-20 MHz
	Reset	Master Clear, Brown Out, Watchdog, Power On
	Instrucciones	35 instrucciones de 14 bits
	Memoria	M. de programa
Memoria	M. de datos RAM	368 registros de 8 bits
	M. de datos EEPROM	256 registros de 8 bits

	Pila	8 palabras de 13 bits
	M. de datos ext. EEPROM	Hasta 256 Kbytes
Periféricos	Puertos programables de E/S	Hasta 33 bits, pueden ser usados por otros periféricos
	Timers/Counters	2 de 8 bits y 1 de 16 bits
	Puertos de captura/comparación de datos	2 de 8 bits
	Moduladores de ancho de pulso (PWM)	2 de 8 bits
	Convertor Analógico/Digital de 10 bits	Con un MPX de 8 canales para 8 entradas diferentes
	Puerto serie sincrónico	Configurable en modo SPI e I2C
	USART	Para conexiones RS 232
	Parallel Slave Port	8 bits + 3 bits de control

Tabla 1.2 Características internas del PIC16F877A.

1.4.7.2.3. Diagrama de bloques de su arquitectura interna.

En la figura 1.12 vemos la arquitectura interna de uno de los PICs más usados últimamente en diseños orientados en proyectos es el PIC16F877A.

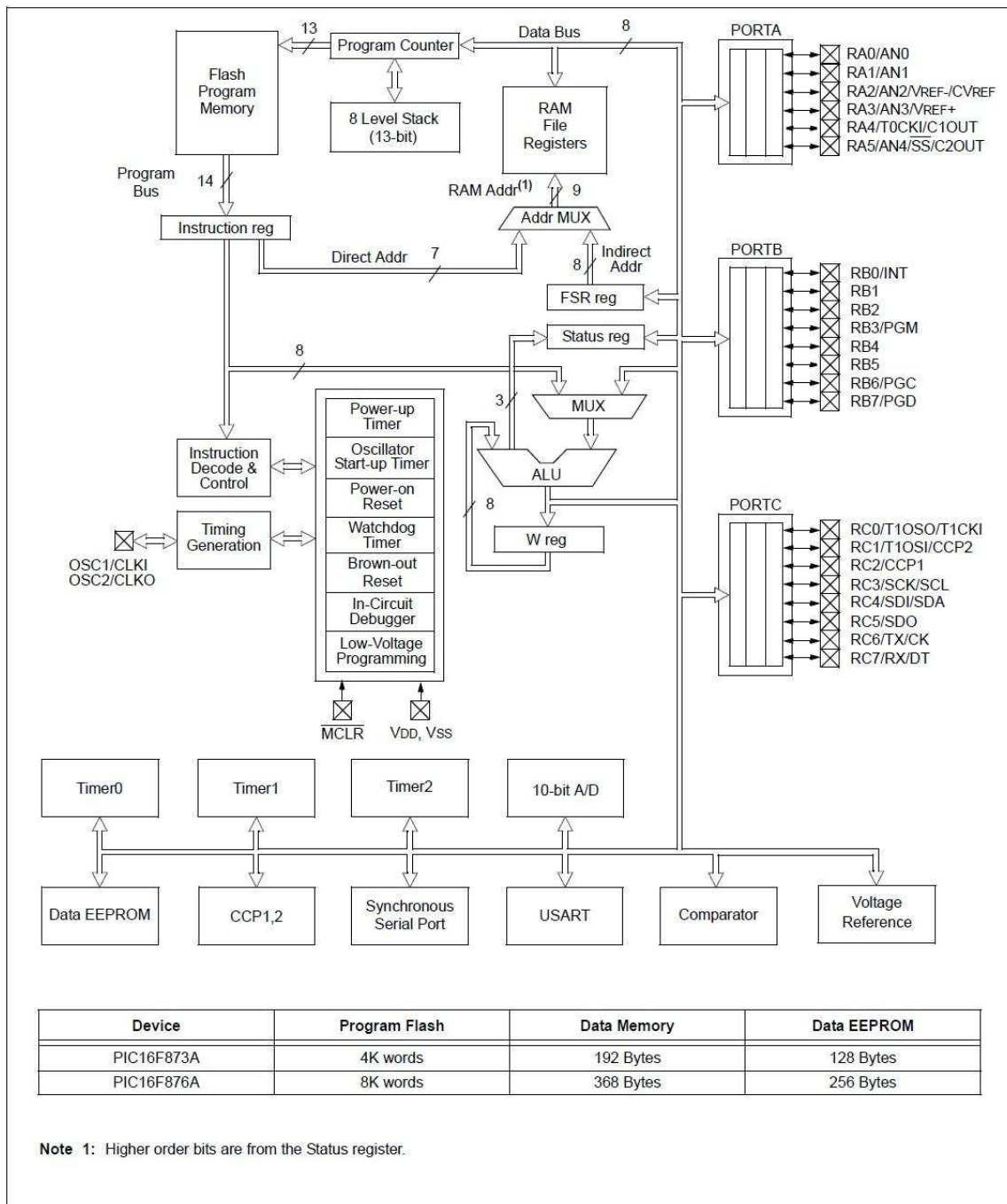


Figura 1.12 Diagrama de bloques del PIC16F877A.

1.4.7.2.4. Diagrama de pines.

En la figura 1.13 tenemos la descripción de pines del PIC16F877A.

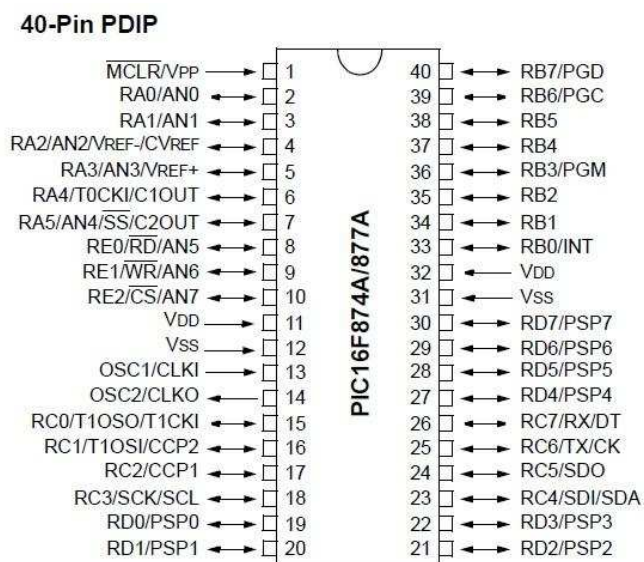


Figura 1.13 Diagrama de pines del PIC16F877A.

1.4.7.2.5. Descripción de cada pin del PIC16F877A.

En la tabla 1.3 se da una descripción general de cada pin, su uso, su número de pin, el tipo de pin y el tipo de buffer.

NOMBRE DEL PIN	PIN	TIPO	TIPO DE BUFFER	DESCRIPCIÓN
OSC1/CLKIN	13	I	ST/MOS	Entrada del oscilador de cristal / Entrada de señal de reloj externa
OSC2/CLKOUT	14	O	-	Salida del oscilador de cristal
MCLR/Vpp/THV	1	I/P	ST	Entrada del Master clear (Reset) o entrada de voltaje de programación o modo de control high voltaje test
PORTA				
				PORTA es un puerto I/O bidireccional

RA0/AN0	2	I/O	TTL	RA0: puede ser salida analógica 0
RA1/AN1	3	I/O	TTL	RA1: puede ser salida analógica 1
RA2/AN2/ Vref-	4	I/O	TTL	RA2: puede ser salida analógica 2 o referencia negativa de voltaje
RA3/AN3/Vref+	5	I/O	TTL	RA3: puede ser salida analógica 3 o referencia positiva de voltaje
RA4/T0CKI	6	I/O	ST	RA4: puede ser entrada de reloj el timer0.
RA5/SS/AN1	7	I/O	TTL	RA5: puede ser salida analógica 4 o el esclavo seleccionado por el puerto serial síncrono.
PORTB				
PORTB				PORTB es un puerto I/O bidireccional. Puede ser programado todo como entradas
RB0/INT	31	I/O	TTL/ST	RB0 puede ser pin de interrupción externo.
RB1	34	I/O	TTL	
RB2	35	I/O	TTL	
RB3/PGM	36	I/O	TTL	RB3: puede ser la entada de programación de bajo voltaje
RB4	37	I/O	TTL	Pin de interrupción
RB5	38	I/O	TTL	Pin de interrupción
RB6/PGC	39	I/O	TTL/ST	Pin de interrupción. Reloj de programación serial
RB7/PGD	40	I/O	TTL/ST	
PORTC				
PORTC				PORTC es un puerto I/O bidireccional
RCO/T1OSO/T1CKI	15	I/O	ST	RCO puede ser la salida del oscilador timer1 o la entrada de reloj del timer1
RC1/T1OS1/CCP2	16	I/O	ST	RC1 puede ser la entrada del oscilador timer1 o salida PWM 2
RC2/CCP1	17	I/O	ST	RC2 puede ser una entrada de captura y comparación o salida PWN
RC3/SCK/SCL	18	I/O	ST	RC3 puede ser la entrada o salida serial de reloj síncrono para modos SPI e I2C

RC4/SD1/SDA	23	I/O	ST	RC4 puede ser la entrada de datos SPI y modo I2C
RC5/SD0	24	I/O	ST	RC5 puede ser la salida de datos SPI
RC6/Tx/CK	25	I/O	ST	RC6 puede ser el transmisor asíncrono USART o el reloj síncrono.
RC7/RX/DT	26	I/O	ST	RC7 puede ser el receptor asíncrono USART o datos síncronos
PORTD				
PORTD				PORTD es un puerto bidireccional paralelo
RD0/PSP0	19	I/O	ST/TTL	
RD1/PSP1	20	I/O	ST/TTL	
RD2/PSP2	21	I/O	ST/TTL	
RD3/PSP3	22	I/O	ST/TTL	
RD4/PSP4	27	I/O	ST/TTL	
RD5/PSP5	28	I/O	ST/TTL	
RD6/PSP6	29	I/O	ST/TTL	
RD7/PSP7	30	I/O	ST/TTL	
PORTE				
PORTE				PORTE es un puerto I/O bidireccional
RE0/RD/AN5	8	I/O	ST/TTL	RE0: puede ser control de lectura para el puerto esclavo paralelo o entrada analógica 5
RE1/WR/AN	9	I/O	ST/TTL	RE1: puede ser escritura de control para el puerto paralelo esclavo o entrada analógica 6
RE2/CS/AN7	10	I/O	ST/TTL	RE2: puede ser el selector de control para el puerto paralelo esclavo o la entrada analógica 7.
Vss				
Vss	12, 31	P	-	Referencia de tierra para los pines lógicos y de I/O
Vdd				
Vdd	11,	P	-	Fuente positiva para los pines lógicos y de

	32			I/O
NC	-	-	-	No está conectado internamente

Tabla 1.3 Descripción de los pines del PIC16F877A.

1.4.7.2.6. Dispositivos periféricos.

- Timer0: Temporizador-contador de 8 bits con pre-escalador de 8 bits
- Timer1: Temporizador-contador de 16 bits con pre-escalador que puede incrementarse en modo sleep de forma externa por un cristal/clock.
- Timer2: Temporizador-contador de 8 bits con pre-escalador y post-escalador.
- Dos módulos de Captura, Comparación, PWM (Modulación de Anchura de Impulsos).
- Conversor A/D de 10 bits.
- Puerto Serie Síncrono Master (MSSP) con SPI e I2C (Master/Slave).
- USART/SCI (Universal Synchronous Asynchronous Receiver Transmitter) con 9 bit.
- Puerta Paralela Esclava (PSP) solo en encapsulados con 40 pines.

1.4.7.2.7. Memorias.

1.4.7.2.7.1 La memoria EEPROM ò memoria de programa.

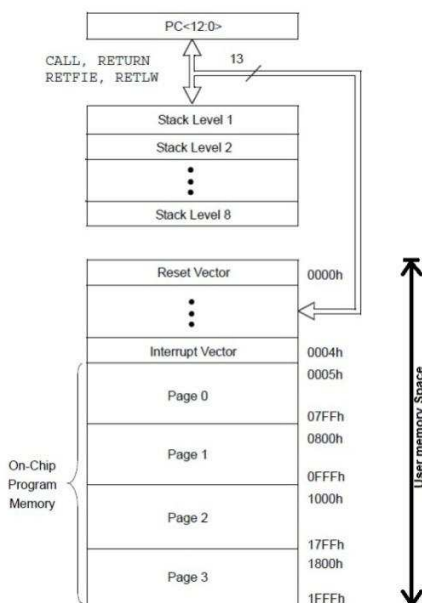


Figura 1.14 Memoria de programa del PIC16F877A.

En la figura 1.14 vemos el diagrama de la memoria EEPROM, el espacio marcado como "User memory Space" es el espacio de memoria donde ira nuestro programa, desde 0000h hasta la 1FFFh.

"Reset Vector" es la primera dirección a la que se dirige el PIC al encenderlo o al resetearlo.

"PC" y los "Stack Level" son empleado por el Pic y nosotros no tenemos acceso a ellos.

1.4.7.2.7.2. La memoria RAM.

La memoria RAM no solo se usa para almacenar nuestras variables, también se almacenan una serie de registros que configuran y controlan el PIC. Podemos

observar en la tabla 1.5 que esta memoria está dividida en 4 Bancos (banco 0,1, 2, 3). Antes de acceder a un registro de esta memoria tenemos que preguntarnos en que banco estamos, generalmente se trabaja en el banco 0 cada banco contiene sus registros que sirven para la configuración y control de la memoria RAM. Estos registros se usan para controlar los pines del PIC, consultar los resultados de las operaciones de la ALU (unidad aritmética lógica), cambiar del banco de memoria entre otras cosas.

La selección del banco de memoria se hace por medio de los bits RP1 y RP0 del registro STATUS (6,5).en la tabla 1.4 se observa que valor deben tener los bits RP1 y RP0 para seleccionar el banco de memoria.

RP1:RP0	BANK
0 0	0
0 1	1
1 0	2
1 1	3

Tabla 1.4 Selección de bancos.

1.4.7.2.7.3. Configuración de los puertos del PIC.

Para poder hacerlo es necesario conocer la tabla de registros de la memoria de datos, la cual está dividida en el BANCO 0, BANCO 1, BANCO 2, BANCO 3.

Los registros importantes en la configuración de los puertos son:

STATUS dirección **03h**

PORTA dirección **05h**

PORTB dirección **06h**

PORTC dirección **07h**

PORTD dirección **08h**

PORTE dirección **09h**

TRISA dirección **85h**

TRISB dirección **86h**

TRISC dirección **87h**

TRISD dirección **88h**

TRISE dirección **89h**

Al configurar los puertos se deberá tener en cuenta que:

Si se asigna un CERO (0) a un pin, éste quedará como salida y...

Si le asignas un UNO (1), quedará como entrada.

Esta asignación se hace en:

TRISA para los pines del **PUERTO A.**

TRISB para los pines del **PUERTO B.**

TRISC para los pines del **PUERTO C.**

TRISD para los pines del **PUERTO D.**

TRISE para los pines del **PUERTO E.**

1.5. TRANSDUCTORES¹².

1.5.1. INTRODUCCIÓN.

Un transductor es un dispositivo capaz de transformar magnitudes físicas o químicas, en magnitudes eléctricas por ejemplo el parlante, que transforma una señal eléctrica en señal acústica. Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, como por ejemplo la temperatura.

1.5.2 SENSORES DE DISTANCIA.

En la figura 1.15 se tiene la clasificación de los tipos de sensores de distancia que existen tanto para cortas distancias y largas distancias, teniendo a los de ultrasonido como sensor de larga distancia.

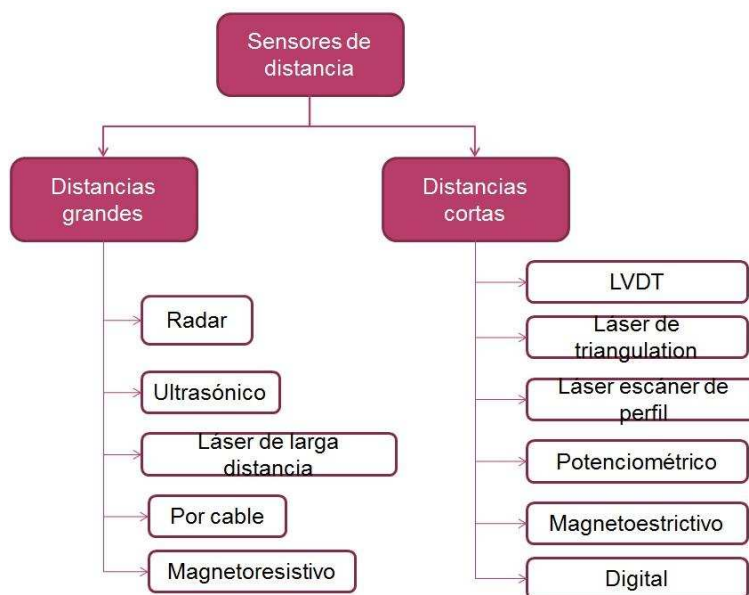


Figura 1.15 Sensores de distancia¹².

¹²<http://www.slideshare.net/diego5wh/sensores-de-distancia>

1.5.3. SENSORES ULTRASÓNICOS¹³.

1.5.3.1 Funcionamiento básico de los ultrasonidos.

Los ultrasonidos son antes que nada sonido, exactamente igual que los que oímos normalmente, salvo que tienen una frecuencia mayor que la máxima audible por el oído humano. Ésta comienza desde unos 16 Hz y tiene un límite superior de aproximadamente 20 KHz, mientras que nosotros vamos a utilizar sonido con una frecuencia de 40 KHz. A este tipo de sonidos es a lo que llamamos Ultrasonidos.

El funcionamiento básico de los ultrasonidos como medidores de distancia se muestra de una manera muy clara en la figura 1.16, donde se tiene un receptor que emite un pulso de ultrasonido que rebota sobre un determinado objeto y la reflexión de ese pulso es detectada por un receptor de ultrasonidos:

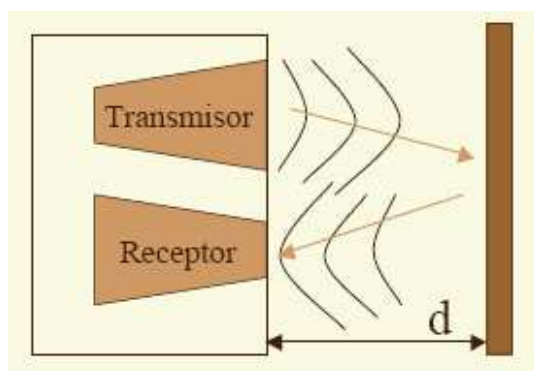


Figura 1.16 Sensores ultrasónicos¹³.

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

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

¹³http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores_de_distancias_con_ultrasonidos.pdf

$$d = \frac{1}{2} V \times t$$

Donde V es la velocidad del sonido en el aire y t es el tiempo transcurrido entre la emisión y recepción del pulso.

1.5.3.2. Sensor ultrasónico SRF05¹⁴.

SRF05 es un nuevo sensor de distancias pensado para ser una actualización del clásico SRF04 con el que es compatible, pero además añadiendo nuevas funciones y características. En el modo estándar, el SRF05 se comporta igual que el SRF04 con la diferencia de que el rango de trabajo se ha aumentado de 3 a 4 metros. Esto significa que todo el software que funciona con el SRF04, funciona con el SRF05. Por otro lado, el SRF05 cuenta con un nuevo modo de trabajo que emplea un solo pin para controlar el sensor y hacer la lectura de la medida. Lo que se hace es mandar un impulso para iniciar la lectura y luego poner el pin en modo entrada. Después basta con leer la longitud del pulso devuelto por el sensor, que es proporcional a la distancia medida por el sensor. El SRF05 es mecánicamente igual al SRF04, por lo que puede ser un sustituto de este.

El sensor SRF05 incluye un breve retardo después del pulso de eco para dar a los controladores más lentos como Basic Stamp y Picaxe el tiempo necesario para ejecutar sus pulsos en los comandos. El sensor SRF05 tiene dos modos de funcionamiento, según se realicen las conexiones.

1.5.3.2.1. Modo 1 (compatible con SRF04, señal de activación y eco independientes).

Este modo utiliza pines independientes para la señal de inicio de la medición y para retorno del eco, siendo el modo más sencillo de utilizar. Todos los ejemplos de códigos para el sensor SRF04 funcionarán para SRF05 en este modo. Para utilizar este modo, simplemente deberá dejar sin conectar el pin de modo - el SRF05 integra

¹⁴<http://superrobotica.com/S320111.htm>.

una resistencia pull-up en este pin. En la figura 1.17 se muestra la descripción de pines del sensor en modo 1.

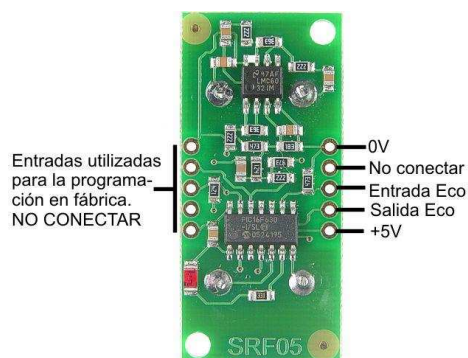


Figura 1.17 Pines del SRF05 en modo 1.

1.5.3.2.1.1 Diagrama de tiempos del srf05 en modo 1.

En la figura 1.18 se observa el diagrama de tiempos del sensor en modo 1, donde un tiempo después del pulso de disparo el sensor emite la ráfaga ultrasónica, luego de un cierto tiempo en la que el sensor recepta el eco genera un pulso cuyo ancho depende del tiempo de ida y vuelta de la ráfaga ultrasónica, este varía entre 100 μ S a 25 mS y de un máximo de 36 mS si no se detecta algún objeto.

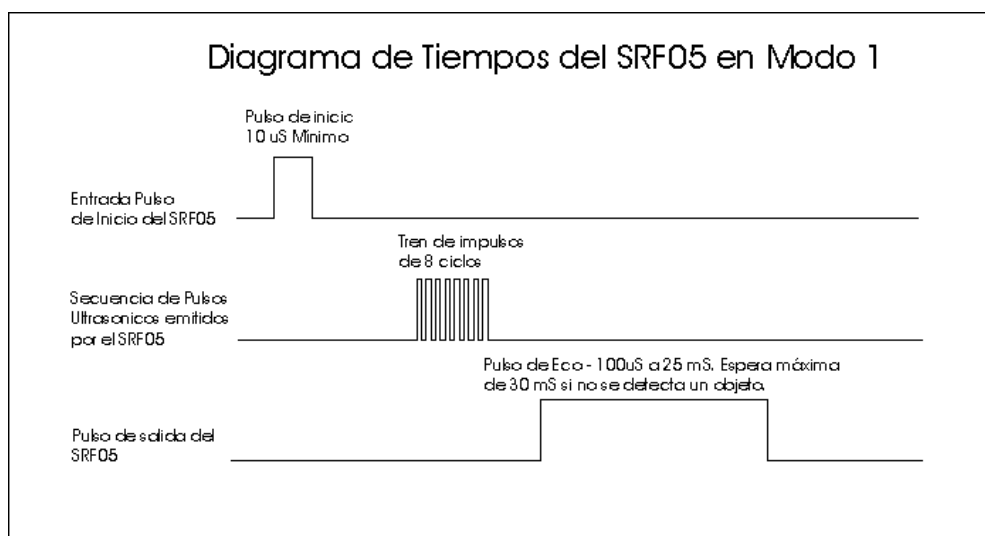


Figura 1.18 Diagrama de tiempos del SRF05 en modo 1.

1.5.3.2.2. Modo 2 (pin único para la señal de activación y eco).

Este modo utiliza un único pin para las señales de activación y eco, y está diseñado para reducir el número de pines en los microcontroladores. Para utilizar este modo, hay que conectar el pin de modo al pin de tierra de 0v tal como se observa en la figura 1.19. La señal de eco aparecerá en el mismo pin que la señal de activación. El SRF05 no elevará el nivel lógico de la línea del eco hasta 700uS después del final de la señal de activación. Dispone de ese tiempo para cambiar el pin del disparador y convertirlo en una entrada para preparar el código de medición de pulsos. El comando PULSIN integrado en la mayor parte de los controladores del mercado lo hace automáticamente.

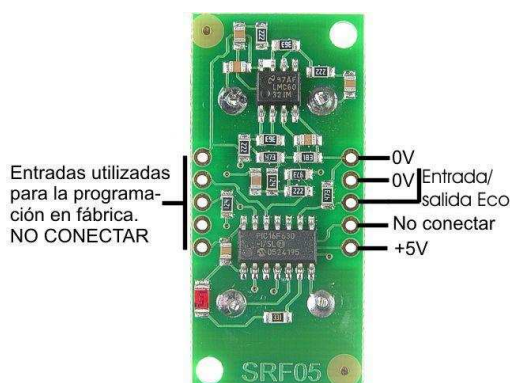


Figura 1.19 Pines del SRF05 en modo 2.

1.5.3.2.2.1. Diagrama de tiempos del srf05 en modo 2.

En la figura 1.20 se observa el diagrama de tiempos del sensor en modo 2, el cual es bastante similar al de modo 1 a diferencia de que el pulso de eco sale por el mismo pin por el que entra la señal de activación.

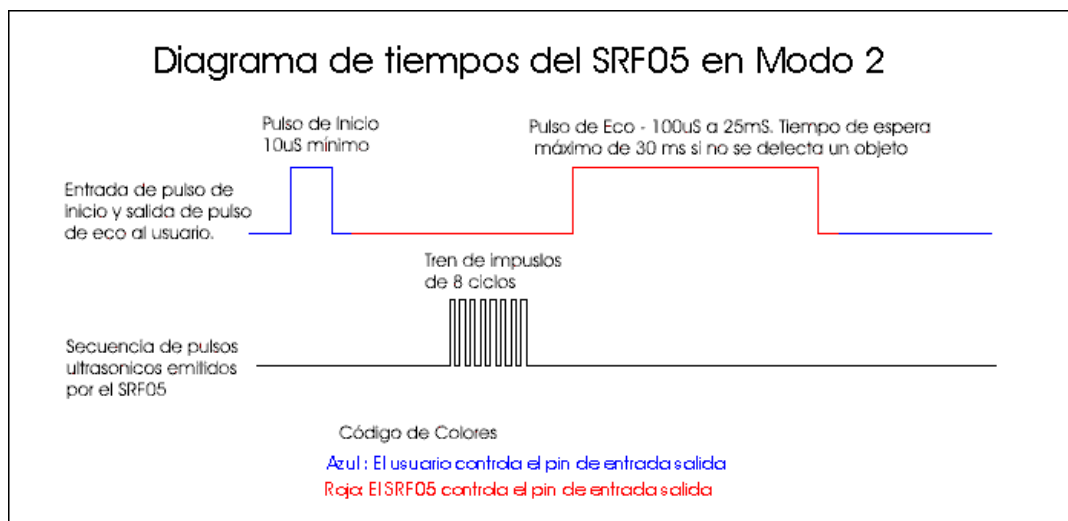


Figura 1.20 Diagrama de tiempos del SRF05 en modo 2.

1.5.3.2.3. Cómo calcular la distancia.

Se mostraron los diagramas de tiempo para el sensor de distancias por ultrasonido SRF05 para cada modo. Se deberá suministrar un breve pulso de al menos 10µs para disparar la entrada de comienzo del cálculo de distancia. El SRF05 transmitirá una ráfaga de 8 ciclos de ultrasonidos a 40kHz elevando el nivel lógico de la señal del eco (o la línea de activación en el modo 2). Entonces el sensor "escucha" un eco, y en cuanto lo detecta, vuelve a bajar el nivel lógico de la línea de eco. La línea de eco es por lo tanto un pulso, cuyo ancho es proporcional a la distancia respecto al objeto. Registrando la duración del pulso es posible calcular la distancia en pulgadas/centímetros o en cualquier otra unidad de medida. Si no se detectase nada, entonces el SRF05 baja el nivel lógico de su línea de eco después de 30ms.

El SRF05 puede activarse cada 50ms, o 20 veces por segundo. Debería esperar 50ms antes de la siguiente activación, incluso si el SRF05 detecta un objeto cerca y el pulso del eco es más corto. De esta manera se asegura que el "bip" ultrasónico ha desaparecido completamente y no provocará un falso eco en la siguiente medición de distancia.

1.5.3.2.4. Los otros 5 pines del sensor.

Los 5 pines marcados como "programming pins" (pines de programación) se utilizan sólo una vez durante el proceso de fabricación para programar la memoria Flash en el chip del PIC16F630. Los pines de programación de PIC16F630 se utilizan también para realizar otras funciones en el SRF05, por lo que deberá asegurarse de que nada esté conectado a ellos o se interrumpirá el funcionamiento de los módulos.

1.5.3.2.5. Patrón y el ancho del haz.

El patrón del haz del sensor SRF05 es cónico mientras que el ancho del haz es una función del área de la superficie de los transductores y es fijo. El patrón del haz de los transductores utilizados en el SRF05, según la hoja de datos de los fabricantes, es la que se observa en la figura 1.21.

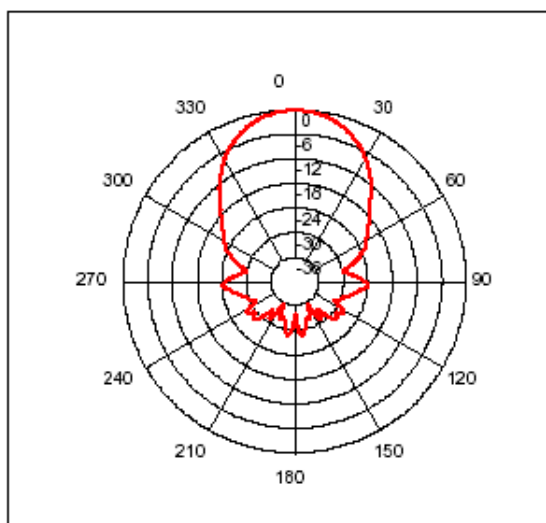


Figura 1.21 Haz de radiación del SRF05.

1.6. MODULO DE VOZ USB-SD MP3 WT9501M03.

En la figura 1.22 se observa al módulo de voz WT9501M03 que maneja archivos de audio en formato MP3, a los cuales se puede acceder mediante 2 modos de uso modo pulsador y modo de control serial.



Figura 1.22 Modulo de voz WT9501M03.

1.6.1. CARACTERISTICAS.

- Reproduce archivos de audio de 8 – 320Kbps.
- Soporta hasta tarjetas de memoria SD de 32G de capacidad.
- Soporta tarjetas SD y memorias flash USB.
- Soporta modo pulsador y modo de control serial.
- Soporta la reproducción directa de audio de cualquiera de las selecciones.
- Integración opcional de un amplificador clase D de salida (3W x 1).
- Dimensiones: 41mm x 39mm.
- Voltaje de operación: DC5V.
- Corriente en reposo: 20mA.
- Corriente máxima de operación: 70mA.

1.6.2. DESCRIPCION DE PINES.

En la figura 1.23 se tiene la distribución de pines del módulo de voz y en la tabla 1.5 la descripción general de cada uno de ellos.

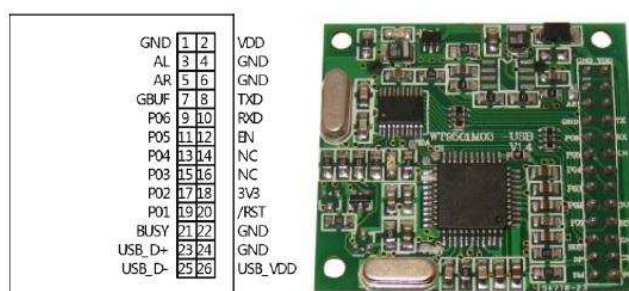


Figura 1.23 Pines del módulo WT9501M03.

No.	Pin Name	Functional Description
1	GND	GND
2	VCC	DC5V input
3	L	Audio left output
4	GND	Power ground
5	R	Audio right output
6	GND	Power ground
7	GBUF	Audio ground
8	TXD	Serial data transmitter
9	P06	I / O port
10	RXD	Serial data receiver
11	P05	I / O port
12	EN	Power Enable
13	P04	I / O port
14	NC	Vacant (Reserved)
15	P03	I / O port

16	NC	Vacant (Reserved)
17	P02	I / O port
18	3V3	DC3.3V Output
19	P01	I / O port
20	/ RST	Reset pin
21	BUSY	Busy signal, the output is low when playing
22	GND	Powerground
23	USB_D +	USB_D +input
24	GND	USB ground
25	USB_D-	USB_D-input
26	USB_VDD	USB Power

Tabla 1.5 Descripción de pines del módulo WT9501M03.

1.6.3. MODO DE FUNCIONAMIENTO.

1.6.3.1. Modo pulsador.

En modo estándar, los puertos de entrada/salida (I/O) P01 a P06 permanecen en un nivel alto mientras esperan un pulso negativo de 10ms que activen los pines. En la tabla 1.6 se da una descripción de la función de los pines que se manejan en este modo.

I/O port	P01	P02	P03	P04	P05	P06
Features	Play/Pause	Last	Next	VOL +	VOL-	Stop

Tabla 1.6 Función de pines en el modo pulsador.

1.6.3.2. Modo de control serial.

La comunicación serial UART es basada en una frecuencias de 9600 baudios. El protocolo de comunicación se define de la forma en que se muestra en la tabla 1.7, donde en la trama se incluye un código de inicio (Start code), el tamaño de la trama

(data length), código de operación (Operation code), bits de datos (data bits) y el código de finalización (End code).

Startcode	Data length	Operationcode	Tenthousandsdigit	Thousandsdigit	Hundredsdigit	Tensdigit	Unitsdigit	Endcode
7E	XX	XX	XX	XX	XX	XX	XX	7E

TABLA 1.7 Protocolo de comunicación usado por el WT9501M03.

1.6.3.2.1. Descripción del código de operación.

En la tabla 1.8 se da una descripción de los códigos de operación (Operation code) que maneja el módulo de voz.

Type	Description	Operation code	Operation data
SD Card	Play (SD card)	A0H	xx xxxxxxxx
	Pause (SD card)	A1H	None
	Play from the pause point (SD card)	A2H	None
	Cease (SD card)	A3H	None
	Volume	A4H	XX
	Last	A5H	None
	Next	A6H	None
	Play onewithoutcycle	A7H	None
	Play all in cycle	A8H	None
	Play one in cycle	A9H	None
USB Flash	Play (USB flash)	B0H	xx xxxxxxxx
	Pause (USB flash)	B1H	None
	Play from the pause point (USB flash)	B2H	None
	Cease (USB flash)	B3H	None
	Volume	B4H	XX
	Last	B5H	None

	Next	B6H	None
	Play onewithoutcycle	B7H	None
	Play all in cycle	B8H	None
	Play one in cycle	B9H	None

TABLA 1.8 Descripción del código de operación.

Los códigos de operación A0 (SD card), B0 (USB flash), the volume A4 (SD card), and B4 (USB flash) necesitan especificarse en el dato de operación, en cambio los otros comandos no necesitan ser especificados.

El WT9501M03 puede reconocer automáticamente los archivos MP3 en la tarjeta SD y Memoria USB. Se debe asignar un número de archivo de acuerdo al tiempo de creación de los archivos. El número del archivo es un número de 5 dígitos. Y este módulo lee los nombres en código ASCII.

Por ejemplo: 00045.mp3.

- El diez milésimo dígito es "0", y en código ASCII es "30H".
- El milésimo dígito es "0", y en código ASCII es "30H".
- El centésimo dígito es "0", y en código ASCII es "30H".
- El décimo dígito es "4", y en código ASCII es "34H".
- El primer dígito es "5", y en código ASCII es "35H".

1.6.3.2.2. Código de inicio (start code).

El código de inicio es 7E, este es un valor asignado para saber cuándo comienza el código q se envía al módulo de voz.

1.6.3.2.3. Tamaño de la trama (data length).

Es el número total de Bytes excluyendo el código de inicio y el código de finalización, pero incluido el propio tamaño de la trama.

1.6.3.2.4. Código de finalización (end code).

Para finalizar la trama se utiliza el mismo código de inicialización el valor 7E.

1.7 LCD¹⁵.

1.7.1. INTRODUCCION.

Una pantalla de cristal líquido o LCD (sigla del inglés liquid crystal display) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. A menudo se utiliza en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.

1.7.2. CARACTERISTICAS.

Cada píxel de un LCD típicamente consiste de una capa de moléculas alineadas entre dos electrodos transparentes, y dos filtros de polarización, los ejes de transmisión de cada uno que están (en la mayoría de los casos) perpendiculares entre sí. Sin cristal líquido entre el filtro polarizante, la luz que pasa por el primer filtro sería bloqueada por el segundo (cruzando) polarizador.

La superficie de los electrodos que están en contacto con los materiales de cristal líquido es tratada a fin de ajustar las moléculas de cristal líquido en una dirección en particular. Este tratamiento suele ser normalmente aplicable en una fina capa de polímero que es unidireccionalmente frotada utilizando, por ejemplo, un paño. La dirección de la alineación de cristal líquido se define por la dirección de frotación.

¹⁵http://es.wikipedia.org/wiki/Pantalla_de_cristal_líquido

Antes de la aplicación de un campo eléctrico, la orientación de las moléculas de cristal líquido está determinada por la adaptación a las superficies. En un dispositivo twistednematic, TN (uno de los dispositivos más comunes entre los de cristal líquido), las direcciones de alineación de la superficie de los dos electrodos son perpendiculares entre sí, y así se organizan las moléculas en una estructura helicoidal, o retorcida. Debido a que el material es de cristal líquido birrefringente, la luz que pasa a través de un filtro polarizante se gira por la hélice de cristal líquido que pasa a través de la capa de cristal líquido, lo que le permite pasar por el segundo filtro polarizado. La mitad de la luz incidente es absorbida por el primer filtro polarizante, pero por lo demás todo el montaje es transparente.

Cuando se aplica un voltaje a través de los electrodos, una fuerza de giro orienta las moléculas de cristal líquido paralelas al campo eléctrico, que distorsiona la estructura helicoidal (esto se puede resistir gracias a las fuerzas elásticas desde que las moléculas están limitadas a las superficies). Esto reduce la rotación de la polarización de la luz incidente, y el dispositivo aparece gris. Si la tensión aplicada es lo suficientemente grande, las moléculas de cristal líquido en el centro de la capa son casi completamente desenrolladas y la polarización de la luz incidente no es rotada ya que pasa a través de la capa de cristal líquido. Esta luz será principalmente polarizada perpendicular al segundo filtro, y por eso será bloqueada y el pixel aparecerá negro. Por el control de la tensión aplicada a través de la capa de cristal líquido en cada píxel, la luz se puede permitir pasar a través de distintas cantidades, constituyéndose los diferentes tonos de gris.

El efecto óptico de un dispositivo twistednematic (TN) en el estado del voltaje es mucho menos dependiente de las variaciones de espesor del dispositivo que en el estado del voltaje de compensación. Debido a esto, estos dispositivos suelen usarse entre polarizadores cruzados de tal manera que parecen brillantes sin tensión (el ojo es mucho más sensible a las variaciones en el estado oscuro que en el brillante). Estos dispositivos también pueden funcionar en paralelo entre polarizadores, en cuyo caso la luz y la oscuridad son estados invertidos. La tensión de compensación en el

estado oscuro de esta configuración aparece enrojecida debido a las pequeñas variaciones de espesor en todo el dispositivo.

Tanto el material del cristal líquido como el de la capa de alineación contienen compuestos iónicos. Si un campo eléctrico de una determinada polaridad se aplica durante un período prolongado, este material iónico es atraído hacia la superficie y se degrada el rendimiento del dispositivo. Esto se intenta evitar, ya sea mediante la aplicación de una corriente alterna o por inversión de la polaridad del campo eléctrico que está dirigida al dispositivo (la respuesta de la capa de cristal líquido es idéntica, independientemente de la polaridad de los campos aplicados)

Cuando un dispositivo requiere un gran número de píxeles, no es viable conducir cada dispositivo directamente, así cada píxel requiere un número de electrodos independiente. En cambio, la pantalla es multiplexada. En una pantalla multiplexada, los electrodos de la parte lateral de la pantalla se agrupan junto con los cables (normalmente en columnas), y cada grupo tiene su propia fuente de voltaje. Por otro lado, los electrodos también se agrupan (normalmente en filas), en donde cada grupo obtiene una tensión de sumidero. Los grupos se han diseñado de manera que cada píxel tiene una combinación única y dedicada de fuentes y sumideros. Los circuitos electrónicos o el software que los controla, activa los sumideros en secuencia y controla las fuentes de los píxeles de cada sumidero.

1.7.3. LCD ALFANUMERICOS¹⁶.

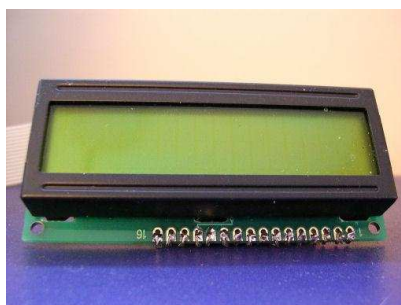


Figura 1.24 LCD alfanumérico.

¹⁶http://www.tolaemon.com/site/lcds_alfanum%C3%A9ricos_compatibles_hd44780

1.7.3.1. Introducción.

Existen gran variedad de LCDs en el mercado, pero quizás los más utilizados en los pequeños proyectos de electrónica son los LCDs alfanuméricos compatibles con el controlador HD44780 de Hitachi como el mostrado en la figura 1.24. Existen otros tipos de LCDs con mejores prestaciones gráficas, pero este controlador ofrece una interfaz muy sencilla y versátil para controlar LCDs alfanuméricos, lo que lo hace ideal para proyectos con microcontrolador en los que no se desea complicar el hardware ni el software.

Los LCDs compatibles con el controlador HD44780 se presentan en el mercado con diferentes configuraciones que se diferencian por el número de filas y de columnas. Existen modelos con 8, 16, 20, 24, 32 o 40 columnas y 1, 2 o 4 filas. La mayoría de los modelos disponen de caracteres de 5x7 pixeles, mientras que algunos tienen caracteres de 5x10 pixeles. La programación de los distintos modelos es prácticamente idéntica, con alguna pequeña diferencia en el proceso de inicialización, configuración y organización de los caracteres en memoria.

1.7.3.2. Conexiones¹⁶.

Normalmente la conexión se realiza mediante un conector de 14 o 16 pines. El conector de 16 pines se usa solo en los modelos que disponen de retroiluminación. En la tabla 1.9 se describe los pines de un LCD de 16 pines donde 8 son para las líneas de datos, 3 para las líneas de control, y el resto son para la alimentación, el control del contraste y el control de la retroiluminación si la tiene. Estos pines no siempre aparecen ordenados en la placa del LCD, por eso, antes de conectar nada es bueno consultar el datasheet para evitar equivocarse y soldar donde no toca. Normalmente estos pines son:

PIN	NOMBRE	FUNCION
1	GND	Masa
2	VCC	Alimentación

3	VEE	Control de contraste
4	RS	Selección de registro de comandos o de datos
5	R/W	Escritura o lectura
6	E	Enable
7	D0	Bit 0 de datos
8	D1	Bit 1 de datos
9	D2	Bit 2 de datos
10	D3	Bit 3 de datos
11	D4	Bit 4 de datos
12	D5	Bit 5 de datos
13	D6	Bit 6 de datos
14	D7	Bit 7 de datos
15	VCC	Alimentación de la retroiluminación
16	VBL	Control de la retroiluminación

Tabla 1.9 Pines del LCD alfanumérico¹⁶.

Estos LCDs pueden operar en modo de 8bits o en modo de 4bits. Este documento únicamente describe el modo de 8 bits, pero con cuatro ideas básicas se puede extender al de 4 bits: la diferencia entre los dos modos es que en el modo de 4bits la transferencia del dato se realiza en dos ciclos de 4 bits mientras que en el modo de 8 bits se realiza en un único ciclo de 8 bits. Si se usa el modo de 4 bits la transferencia de datos tiene lugar por las líneas de datos de D4 a D7, el resto de líneas, es decir D0 a D3 se puede conectar a masa mediante una resistencia (por ejemplo de 10k). Como se puede ver, el primer modo consume menos puertos de E/S pero es más lento y un poco más complejo de programar, en cambio el segundo modo consume más puertos de E/S pero es más rápido y sencillo de programar. Como todas las operaciones son de escritura, en este documento también se supone que en todo momento la línea R/W está a 0, es decir directa ente conectada a masa.

En la figura 1.25 se tiene el esquema de conexión del LCD en modo de 8 bits, donde los 8 bits son usados para transmitir los datos q se visualizaran en la pantalla.

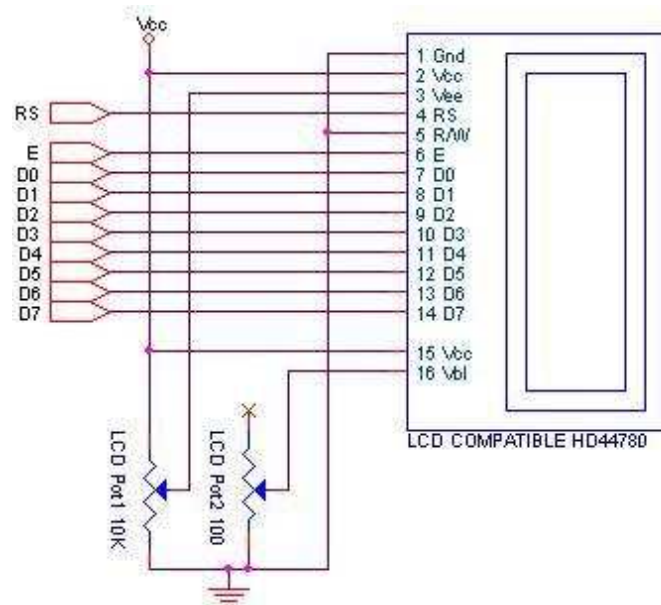


Figura 1.25 Esquema de conexiones en modo 8 bits¹⁶.

En la figura 1.26 se tiene el esquema de conexión del LCD en modo de 4 bits, donde los 4 bits más significativos son usados para transmitir los datos que se visualizaran en la pantalla.

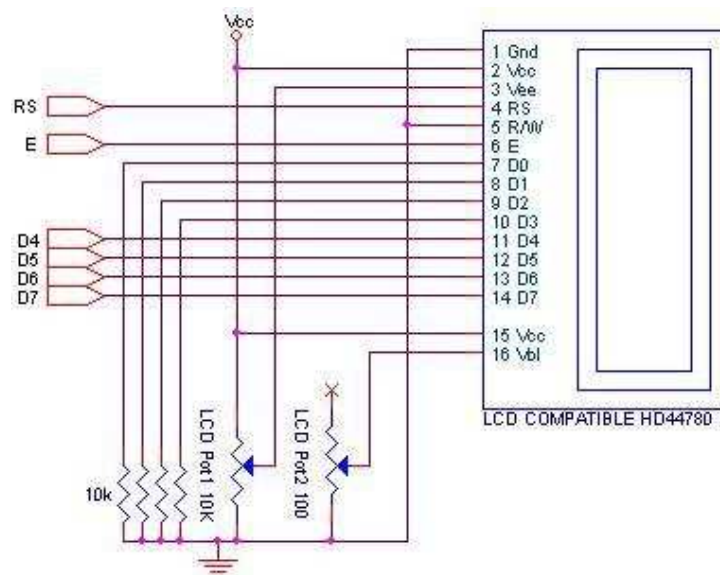


Figura 1.26 Esquema de conexiones en modo 4 bits¹⁶.

1.7.3.3. Funcionamiento.

El funcionamiento general del LCD, en el modo de 8 bits, consiste en ejecutar simultáneamente las siguientes acciones:

- Mantener el dato en las líneas de datos.
- Indicar mediante la línea RS si el dato es un carácter o un comando. Cuando RS está a 0 los datos se interpretan como un comando mientras que cuando está a 1 se interpretan como un carácter.
- Una vez el dato está listo y se ha indicado si es un carácter o comando se da un pulso a la señal E para que el LCD capture los datos. En función de si se trata de un comando o un carácter, se ejecutará una operación, o se escribirá un carácter en pantalla. La duración del pulso es una variable importante ya que en LCDs antiguos, con controladores lentos, puede ser que un pulso demasiado rápido no de tiempo al LCD a procesar los datos.

En la figura 1.27 se observa el envío de un comando ($R/S=0$) en la parte izquierda de la figura mientras que la de la derecha se muestra el envío de un carácter ($R/S=1$). Por tanto a través de la línea de datos del LCD se pueden transmitir dos tipos de datos: los correspondientes a caracteres y los correspondientes a comandos

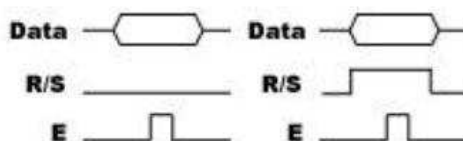


FIGURA 1.27 Timing orientativo de las señales involucradas en el envío de un dato. ¹⁶.

CAPITULO II.

CONSTRUCCIÓN DEL HARDWARE DEL PROYECTO.

2.1. INTRODUCCIÓN.

La construcción del hardware se desarrolló mediante la implementación de un sensor ultrasonido, un microcontrolador y un módulo de voz cuya conexión aplicación y funcionamiento se detallan en este capítulo, con el fin de que se cumplan los objetivos planteados del proyecto y se tenga un correcto funcionamiento entre el sistema de adquisición, el sistema de control y el sistema actuador.

2.2. ARQUITECTURA DE PROYECTO.

Con la investigación obtenida de diferentes fuentes podemos implementar las características de los sistemas electrónicos que permitieron la construcción del sistema electrónico de alerta de obstáculos para personas no videntes mediante el uso de un sensor.

2.2.1. PRINCIPIO FILOSOFICO DEL CONTROL DEL SISTEMA ELECTRÓNICO.

Parte de los objetivos de la implementación del sistema electrónico de alerta de obstáculos para personas no videntes mediante el uso de un sensor, son el de poder detectar obstáculos frontales, gradas y huecos calculando a que distancia se encuentran e informando por medio de audio su valor.

El sistema electrónico está basado en la adquisición de datos, con el uso del sensor ultrasonido el cual al recibir un pulso de disparo de 10 microsegundos por parte del microcontrolador, envía un pulso ultrasónico de 40 MHz que rebotará si se encuentra con un obstáculo menor a los 4 metros y mayor a 1 centímetro, el tiempo que le tome al pulso ultrasónico en ir y regresar se denomina como eco de

la señal. Luego de un tiempo el sensor envía una señal binaria que pasa de un estado alto a un estado bajo con un tiempo de bit proporcional al eco de la señal, a este tiempo lo podemos muestrear para conocer la distancia al obstáculo. El microcontrolador calcula la distancia y envía al módulo de voz el resultado obtenido mediante la comunicación USART (comunicación serial) indicando el mensaje que se debería reproducir para hacerle conocer a la persona no vidente la distancia al obstáculo.

2.3. DIAGRAMA DE BLOQUES DEL PROYECTO.

De manera global el sistema de alerta de obstáculos está formado por un cerebro (el PIC16F877A), un sistema sensorial (sensor SRF05) y un sistema actuador (módulo de voz) tal y como observamos en la figura 2.1.

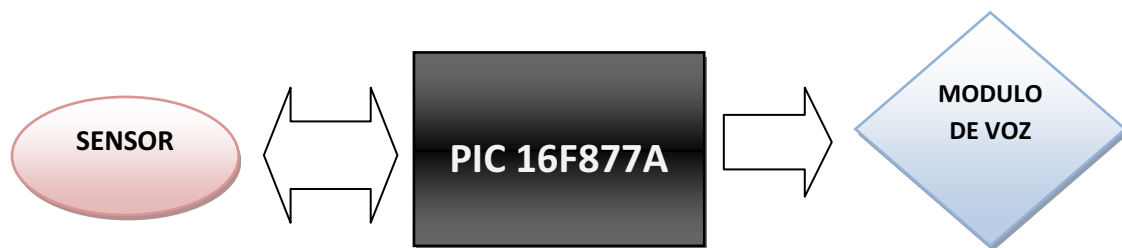


Figura 2.1. Diagrama del Proyecto.

2.4. ANÁLISIS DEL SISTEMA.

El circuito dispone de un PIC16F877A el cual está ubicado en una caja al nivel de la cintura del invidente, este controla al sensor SRF05 el cual está en un mando que es sujetado por la mano y con la cual activa o no la detección tal y como podemos observar en la figura 2.2.



Figura 2.2 Análisis del sistema.

2.5. SISTEMA DE CONTROL.

El sistema de control se hace con el uso del PIC16F887A, el cual se encarga de administrar todos los procesos que se efectúan en el sistema de detección y actuación, es el encargado de dar un funcionamiento sistemático, controlado y lógico con la ayuda del software implementado en el proyecto.

2.6. CONEXIONES A LA TARJETA PRINCIPAL.

2.6.1. CONEXIÓN DE LOS OSCILADORES.

Los modos LP, XT y HS utilizan un oscilador externo como una fuente de reloj cuya frecuencia está determinada por un cristal de cuarzo o por resonadores cerámicos conectados a los pines OSC1 y OSC2. Dependiendo de las características de los componentes utilizados, seleccione uno de los siguientes modos:

- Modo LP - (Baja potencia) se utiliza sólo para cristal de cuarzo de baja frecuencia. Este modo está destinado para trabajar con cristales de 32.768 KHz normalmente embebidos en los relojes de cristal. Es fácil de reconocerlos por sus dimensiones pequeñas y una forma cilíndrica. Al utilizar este modo el consumo de corriente será menor que en los demás modos.

- Modo XT se utiliza para cristales de cuarzo de frecuencias hasta 8 MHz. El consumo de corriente es media en comparación con los demás modos.
- Modo HS - (Alta velocidad) se utiliza para cristales de reloj de frecuencia más alta de 8 MHz. Al utilizar este modo el consumo de corriente será mayor que en los demás modos.

En la figura 2.3 muestra el circuito oscilador tipo XT.

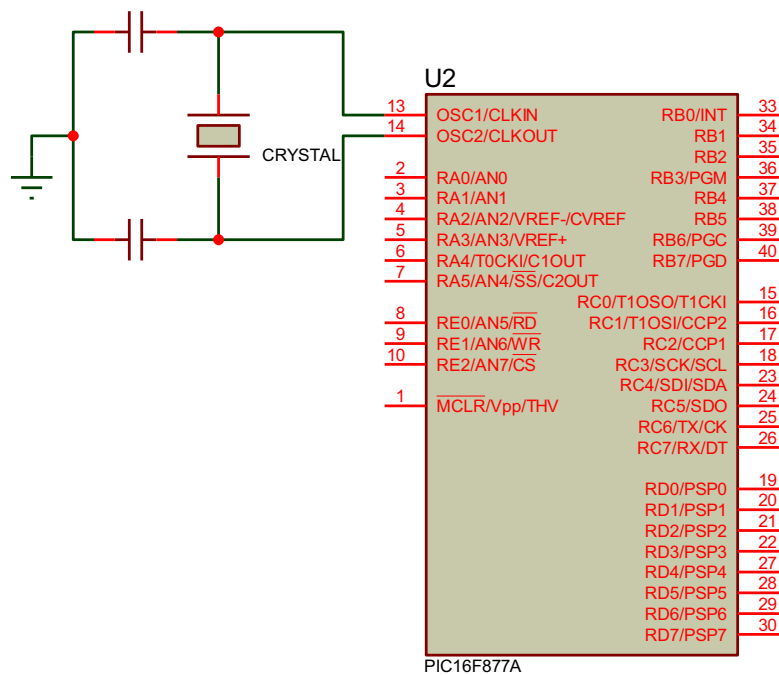


Figura 2.3 Conexión del Oscilador tipo XT.

2.6.2. CONEXIÓN DEL SENSOR SRF 05.

El sistema electrónico usa el sensor ultrasónico SRF05 para la detección de obstáculos dependiendo de a donde se apunte el sistema de adquisición que se sujeta con la mano. En la tabla 2.1 se tiene la descripción de cada uno de sus pines.

Pines	Descripción
+5Vcc	Tensión positiva de alimentación
ECO	Salida del pulso cuya anchura determina el tiempo del recorrido de la señal ultrasónica
Disparo	Entrada de inicio de una nueva medida. Se aplica un pulso con una duración mínima de 10 us.
Modo (N.C.)	Sin conexión se selecciona el modo 1 de compatibilidad con SRF04. Conectado a GND se selecciona el modo 2 de trabajo
GND	Tierra de alimentación.

Tabla 2.1 Conexión del Sensor srf05.

En la figura 2.4 se tiene la conexión del sensor con el PIC para su correcto funcionamiento. Este sensor funciona emitiendo pulsos ultrasónicos que son activados por el PIC16f877A al enviar por el pin RD0 del pòrtico D hacia el pin de Disparo del sensor un pulso de 10uS. Cuando el sonido es reflejado y captado de nuevo por el receptor de ultrasonido este envía por el pin ECO al pin RD1 del pòrtico D del PIC un pulso modulado en ancho de pulso cuyo tamaño de bit es proporcional al tiempo de ida y vuelta de los pulsos ultrasónicos.

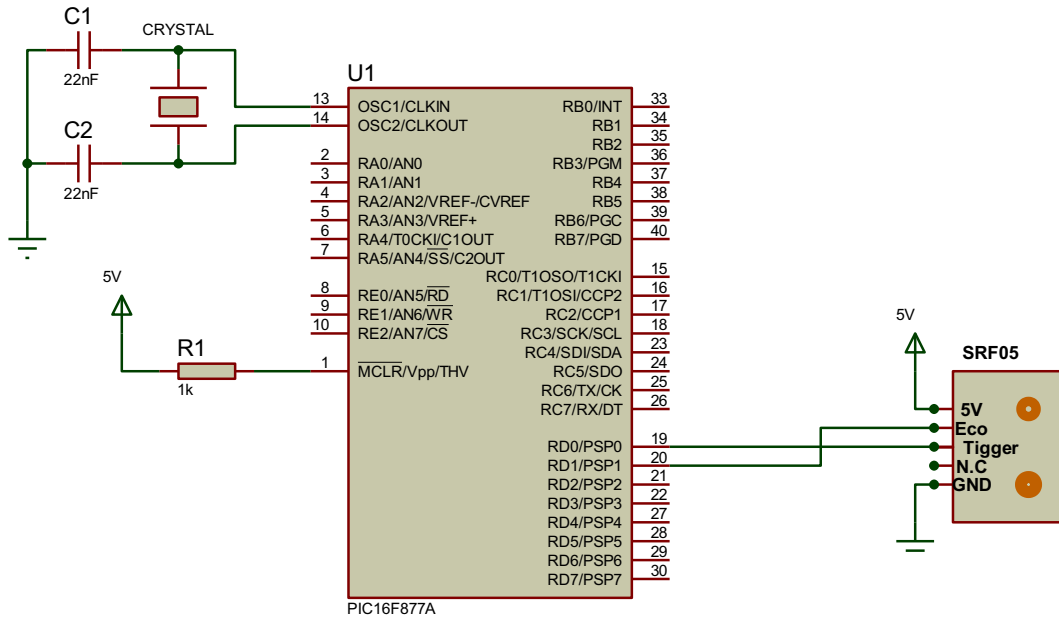


Figura 2.4 Conexión del sensor SRF 05.

2.6.3. CONEXIÓN DEL MÓDULO DE VOZ.

En la figura 2.5 se muestra la conexión para la comunicación serial por medio del protocolo de comunicación USART, el PIC manda una secuencia de bits para establecer un código que el módulo de voz recepte y pueda saber cuál mensaje de voz tiene que reproducir, esto se lo realiza mediante el pin de transmisión (Tx) del pòrtico C, que es salida asincrónica del módulo USART.

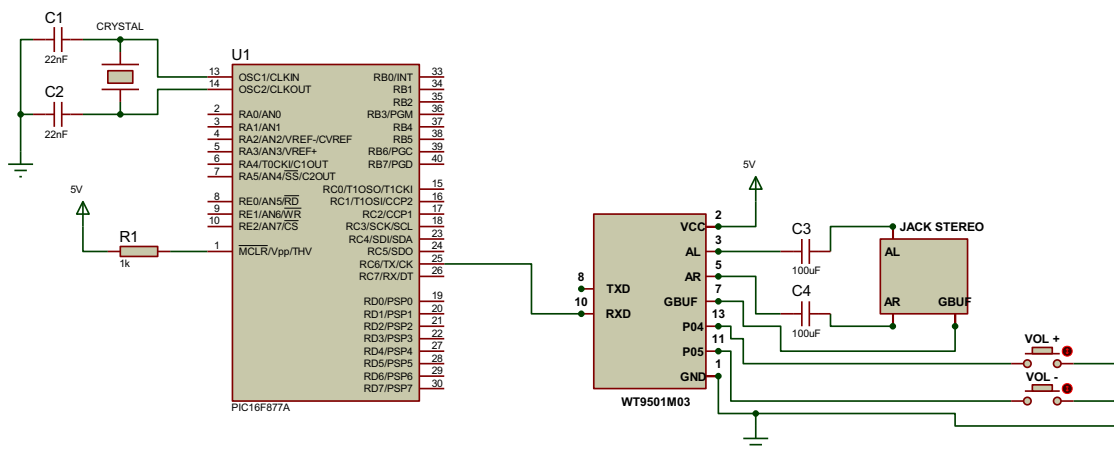


Figura 2.5 Conexión del módulo de voz.

2.6.4. CONEXIÓN DEL LCD.

En la figura 2.6 se muestra la conexión al LCD usando un interfaz de 4 bits.

El LCD se conecta a través del pórtilo B donde los pines de control del LCD RS y E están conectados a los pines RB2 y RB3 respectivamente. Los pines de datos más significativos del LCD D4, D5, D6 y D7 están conectados a los pines RB4, RB5, RB6 y RB7 respectivamente. Por medio del pin VEE del LCD y un potenciómetro se puede controlar el contraste del mismo.

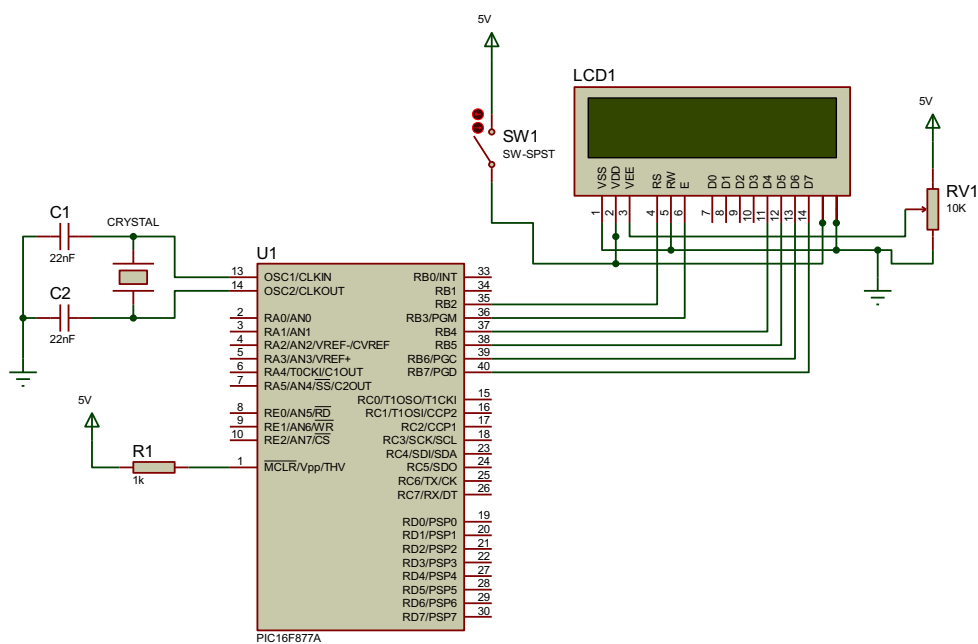


Figura 2.6 (Conexión del LCD)

2.7. CARGADOR DE BATERIAS.

En la figura 2.7 se muestra el circuito cargador de la batería para el sistema electrónico, el cual es para una batería de 6 voltios que sirve para la alimentación del circuito.

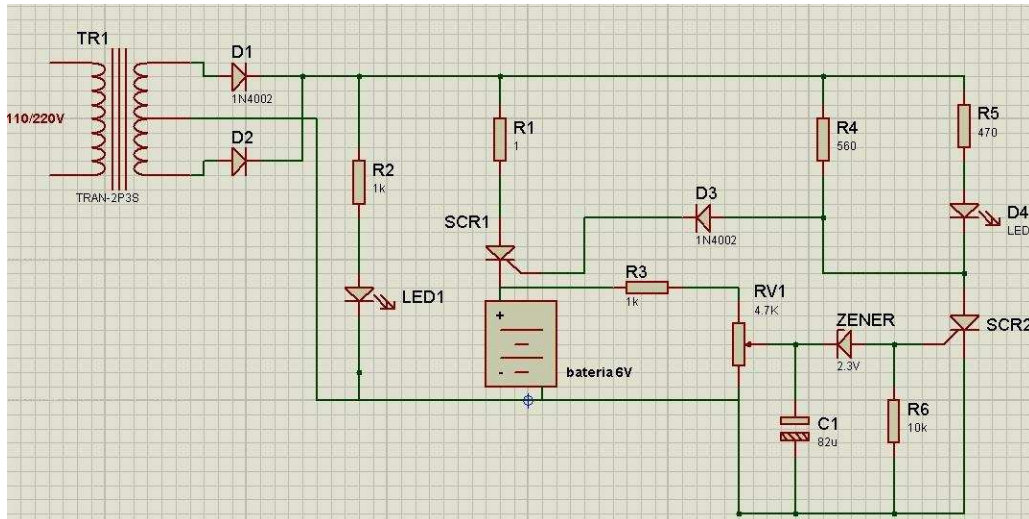


Figura 2.7 Circuito del Cargador de Baterías.

La batería utilizada es de plomo ácido de 6V que permite que el circuito funcione por un largo tiempo hasta que se descargue, para este tipo de baterías fue necesario implementar un circuito cargador con carga controlada de manera que cuando la batería este totalmente cargada se desactive la función de carga y se dé una señal de carga completa.

En la figura 2.7 se observa el diagrama del circuito cargador donde en la salida del secundario se tiene un voltaje de 12V de corriente alterna, la cual es rectificadas por los diodos D1 y D2 (cada uno conduce la mitad del ciclo), lo que proporciona 1.5 amperios, no se filtra el voltaje, en este caso nos sirve para que el SCR pueda ser desconectado al cortarse el voltaje de su compuerta, lo que no ocurre si filtramos la corriente. Si vemos el diagrama del circuito, la batería está conectada en serie con la fuente y el SCR1.

La compuerta del SCR1 es polarizada por el resistor de 560 Ω (R4) y el diodo 1N4002 (D3), en el sentido de conducir la corriente cuando la tensión alcanza aproximadamente 1 V. En la compuerta del SCR2 tenemos el circuito sensor de carga, el cual está formado por un divisor de voltaje (el zener y el SCR2). Ajustamos el trimpot (potenciómetro miniatura) de 4.7K Ω (RV1), para obtener la tensión de disparo del SCR2, que en este caso, corresponde a la tensión del zener cuando la batería está completamente cargada.

Para una batería de 6 voltios, el voltaje del zener debe de ser de 2.1 ó 2.4. Cuando una batería presenta entre sus terminales la tensión que corresponde a la carga completa, el zener conduce y el SCR2 se dispara. Bajo estas condiciones, el mismo prácticamente pone a tierra la compuerta del SCR1, lo que impide el disparo de este, y por lo tanto interrumpe la carga. Al mismo tiempo, la conducción completa del SCR2 hace que el LED2 (verde) se encienda indicando que la batería está completamente cargada.

2.8. CONEXIÓN DEL REGULADOR DE VOLTAJE (7805A).

El regulador de voltaje (7805A) forma parte de la fuente de alimentación para el sistema electrónico como se muestra en la figura 2.8 para su funcionamiento. A la entrada se tendrá un voltaje de 6V provenientes de la batería y a la salida se tiene los 5 V necesarios para que el circuito funcione correctamente, ya que tanto el PIC como el módulo de voz solo admiten 5V estándar.

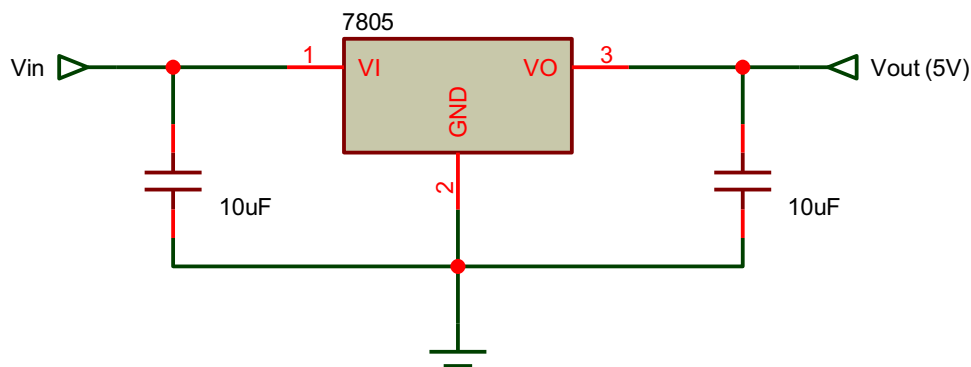


Figura 2.8 Conexión del Regulador de Voltaje.

2.9. SISTEMA DE REPRODUCCIÓN DE MENSAJES DE VOZ.

2.9.1. INTRODUCCIÓN.

El módulo de voz consta de una tarjeta SD con una memoria de 1 GByte en donde se almacena los mensajes en formato Mp3 con una salida estereofónica a los audífonos.

2.9.2. DESCRIPCIÓN DEL MÓDULO DE VOZ (WT9501M03).

2.9.2.1. Direccinamiento de la memoria SD.

La base de UART para la comunicación serial entre el PIC y el módulo de voz, se debe aplicar un modo serial de 9600 baudios, que se establece como protocolo de comunicación. Este incluye:

- Start code
- Data length
- Word size
- End code.

2.9.3. PROCESO DE GRABACIÓN DE MENSAJES.

Para la grabación de los mensajes de voz se utilizó el programa Audacity, en la figura 2.9 se observa parte de los mensajes grabados con este software, luego se redujo los mensajes y se atenuó el ruido con ayuda de las herramientas del programa. Finalmente se los exporto en formato MP3 para que sean reconocidos por el módulo de voz.

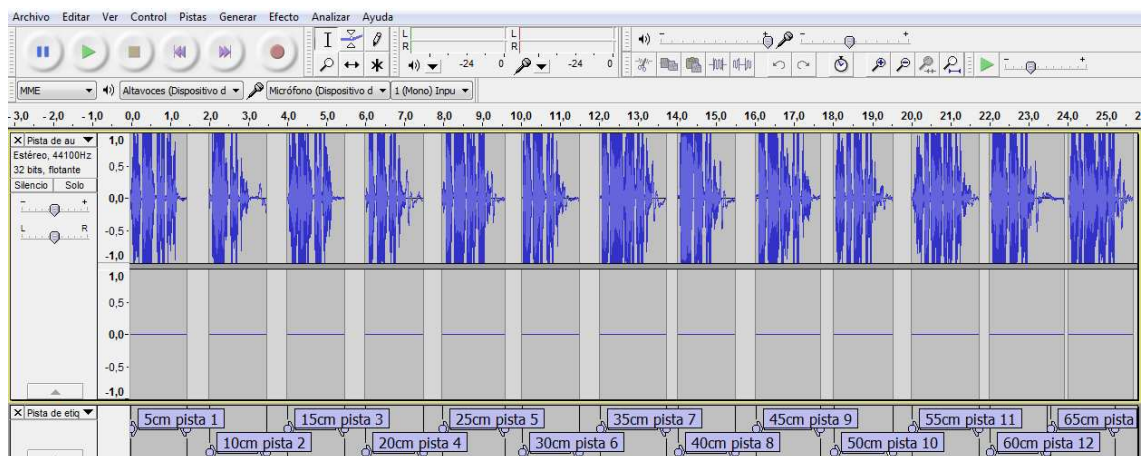


Figura 2.9 Proceso de grabación de los mensajes de voz.

En la programación del microcontrolador se puede dar avisos de la distancia de los objetos detectados hasta los 400 centímetros, para seleccionar cada uno de

los mensajes el microcontrolador tiene que enviar un código determinado, en la tabla 2.2 se describe cada uno de los mensajes grabados con su respectivo código de selección.

Número de pista	Distancia [cm]	Código USART
1	0-5	7E- 07- A0- 30- 30- 30- 30- 31- 7E
2	6-10	7E- 07- A0- 30- 30- 30- 30- 32- 7E
3	11-15	7E- 07- A0- 30- 30- 30- 30- 33- 7E
4	16-20	7E- 07- A0- 30- 30- 30- 30- 34- 7E
5	21-25	7E- 07- A0- 30- 30- 30- 30- 35- 7E
6	26-30	7E- 07- A0- 30- 30- 30- 30- 36- 7E
7	31-35	7E- 07- A0- 30- 30- 30- 30- 37- 7E
8	36-40	7E- 07- A0- 30- 30- 30- 30- 38- 7E
9	41-45	7E- 07- A0- 30- 30- 30- 30- 39- 7E
10	46-50	7E- 07- A0- 30- 30- 30- 31- 30- 7E
11	51-55	7E- 07- A0- 30- 30- 30- 31- 31- 7E
12	56-60	7E- 07- A0- 30- 30- 30- 31- 32- 7E
13	61-65	7E- 07- A0- 30- 30- 30- 31- 33- 7E
14	66-70	7E- 07- A0- 30- 30- 30- 31- 34- 7E
15	71-75	7E- 07- A0- 30- 30- 30- 31- 35- 7E
16	76-80	7E- 07- A0- 30- 30- 30- 31- 36- 7E

17	81-85	7E- 07- A0- 30- 30- 30- 31- 37- 7E
18	86-90	7E- 07- A0- 30- 30- 30- 31- 38- 7E
19	91-95	7E- 07- A0- 30- 30- 30- 31- 39- 7E
20	96-100	7E- 07- A0- 30- 30- 30- 32- 30- 7E
21	101-150	7E- 07- A0- 30- 30- 30- 32- 31- 7E
22	151-200	7E- 07- A0- 30- 30- 30- 32- 32- 7E
23	201-250	7E- 07- A0- 30- 30- 30- 32- 33- 7E
24	251-300	7E- 07- A0- 30- 30- 30- 32- 34- 7E
25	301-350	7E- 07- A0- 30- 30- 30- 32- 35- 7E
26	351-400	7E- 07- A0- 30- 30- 30- 32- 36- 7E
27	< 2 o >400	7E- 07- A0- 30- 30- 30- 32- 37- 7E

Tabla 2.2 Código USART para la reproducción de mensajes en el módulo de voz.

2.10. DISEÑO DEL CIRCUITO ESQUEMATICO.

Para el diseño esquemático se usó el programa de diseño electrónico Proteus 7.7 con el cual se logró hacer el diseño que se muestra en la figura 2.10 donde se observa la conexión de del sensor, el módulo de voz y el LCD al microcontrolador así como también el regulador de voltaje y el cargador para la batería.

2.11. DISEÑO DEL CIRCUITO IMPRESO DE LA PLACA.

La creación del circuito impreso se lo realizo con el software EAGLE 5.11, con este programa podemos obtener la parte esquemática (schematic) y parte circuital (board).

2.11.1. PASOS PARA LA CREACIÓN DEL CIRCUITO ESQUEMÁTICO.

Desde la figura 2.11 a la 2.23 se muestra el proceso a seguir para crear el board para la placa principal, de igual forma se lo hace para las demás placas del sistema electrónico.

1. Abrimos el programa EAGLE en el menú de inicio.

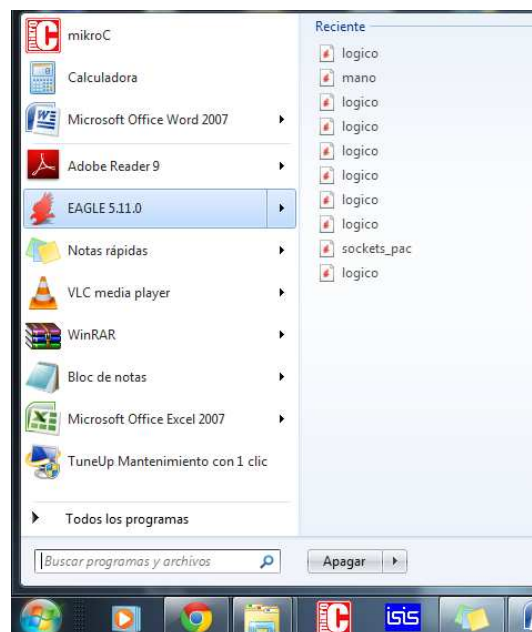


Figura 2.11 Ejecutando Eagle 5.11.0

2. Hacemos click en File, New y Schematic.

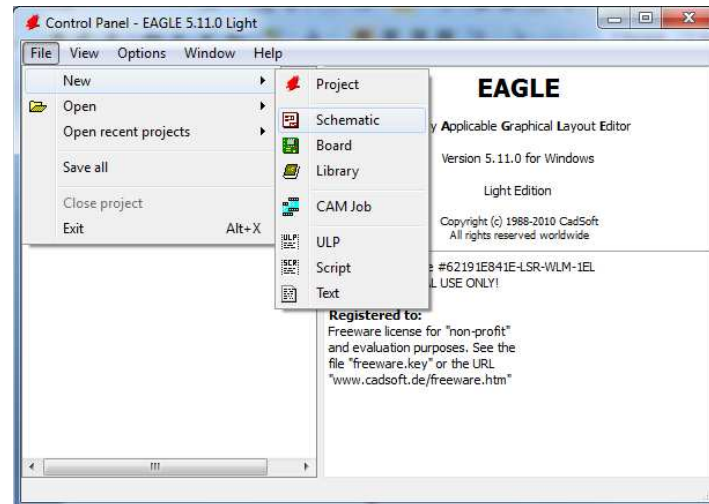


Figura 2.12 Ventana del Eagle.

3. Realizamos el circuito lógico de nuestro proyecto.

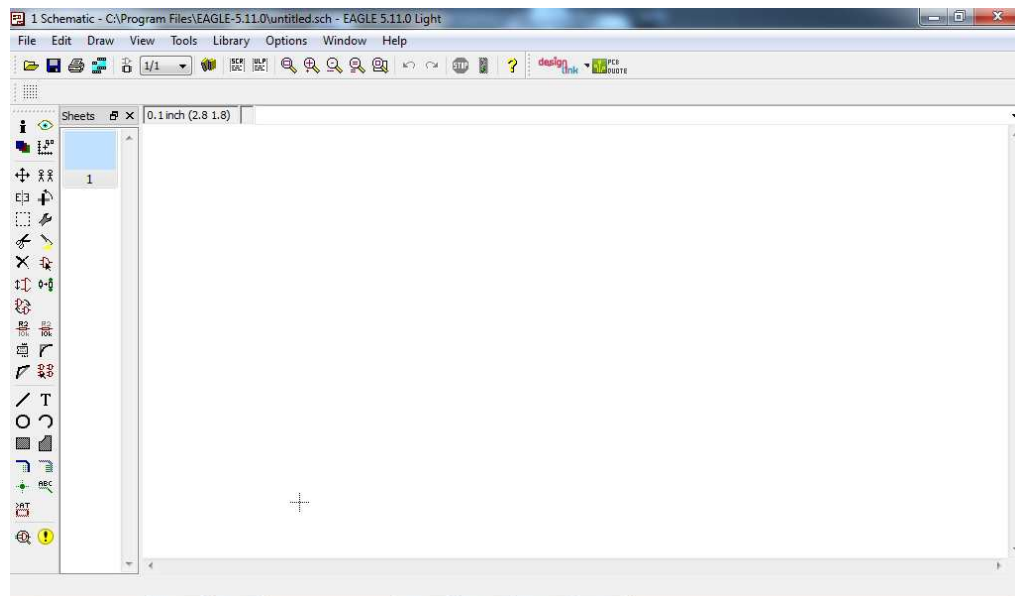


Figura 2.13 Ventana para el diseño de un nuevo esquema.

4. Para comenzar agregar nuestros elementos hacemos clic en Add en la barra de herramientas.

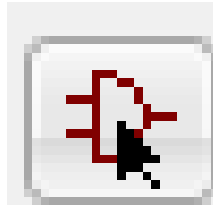


Figura 2.14 Herramienta Add.

5. Una vez ubicados los elementos, hacemos clic en Wire para la conexión entre ellos.

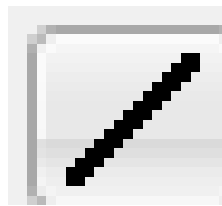


Figura 2.15 Herramienta Wire

6. Con el circuito completo podemos realizar el board que se encuentra en la barra de herramientas.

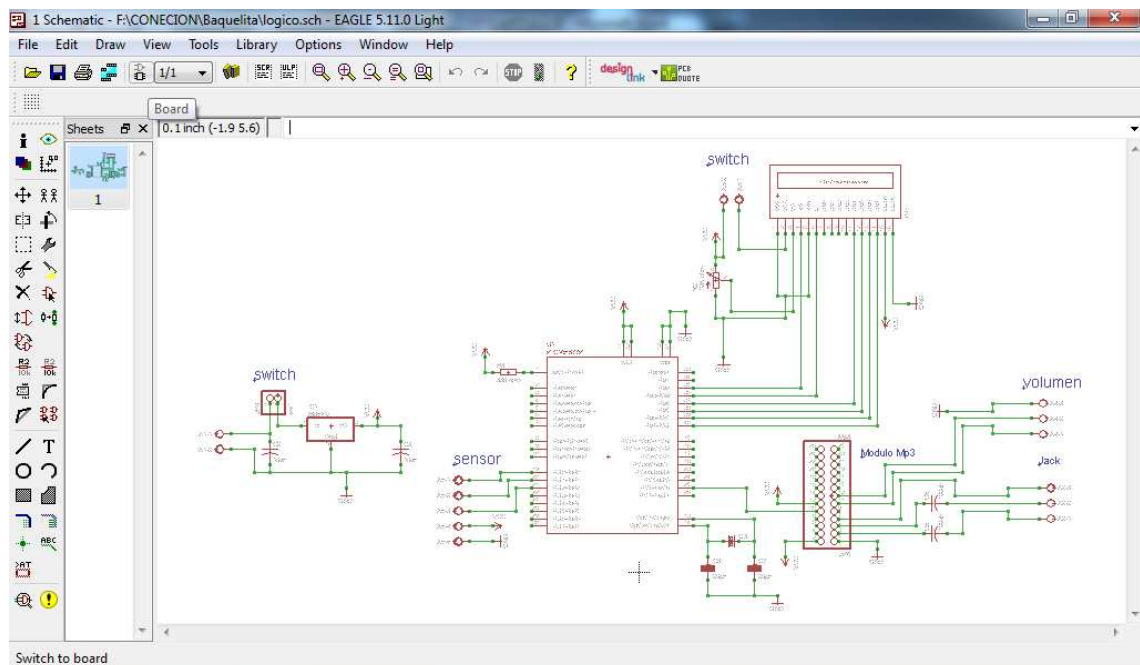


Figura 2.16 Esquema finalizado.

7. Ubicamos en el cuadro de 10x10 cm que va hacer el tamaño de nuestra placa.

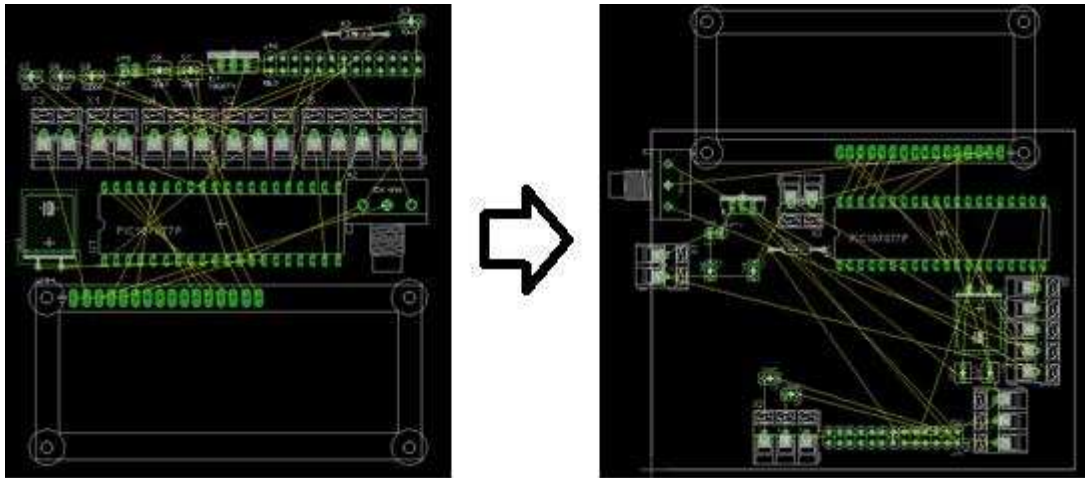


Figura 2.17 Ordenar los elementos.

8. Se realiza el auto ruteado que se encuentra en la barra de herramientas.

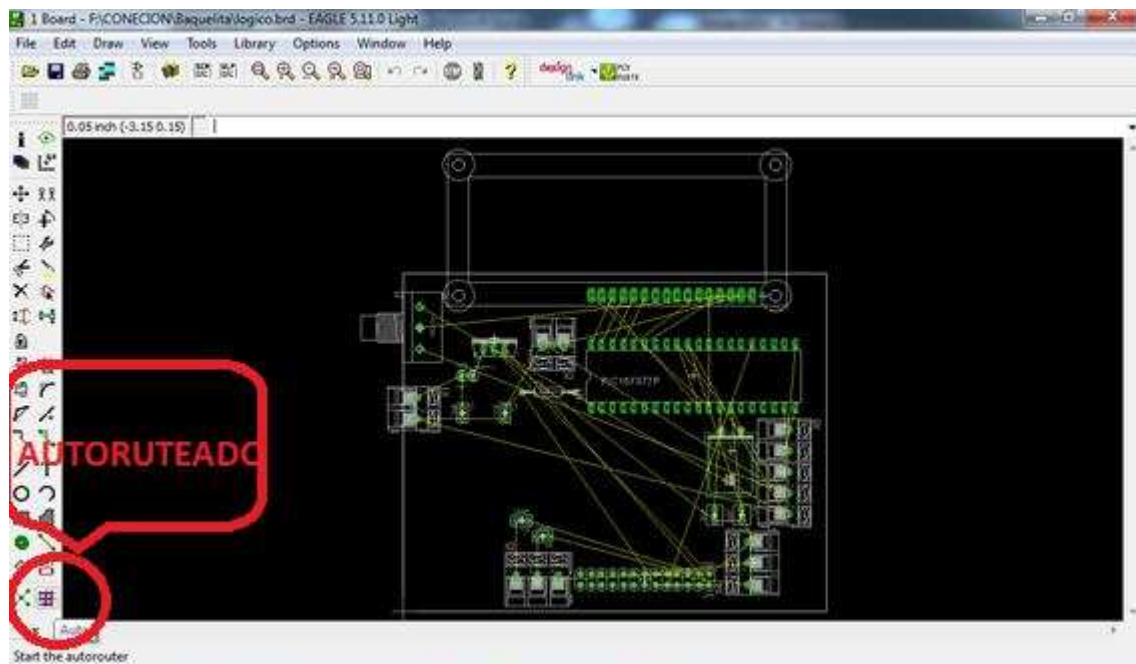


Figura 2.18 Auto ruteado.

9. Se configura el auto ruteado como se muestra en la figura.

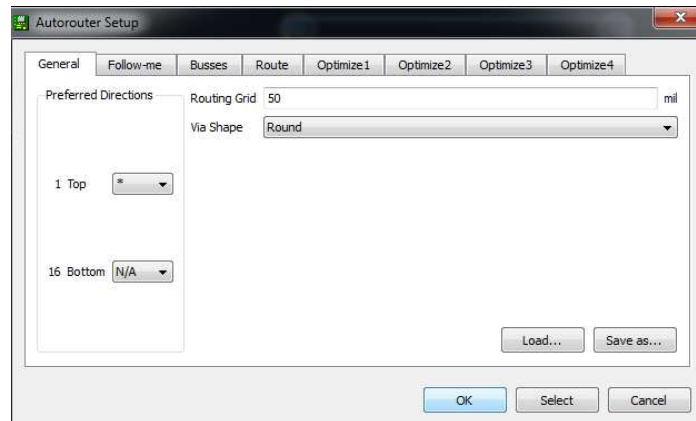


Figura 2.19 Configuración para el auto ruteado.

10. En la figura 2.20 se muestra el circuito board.

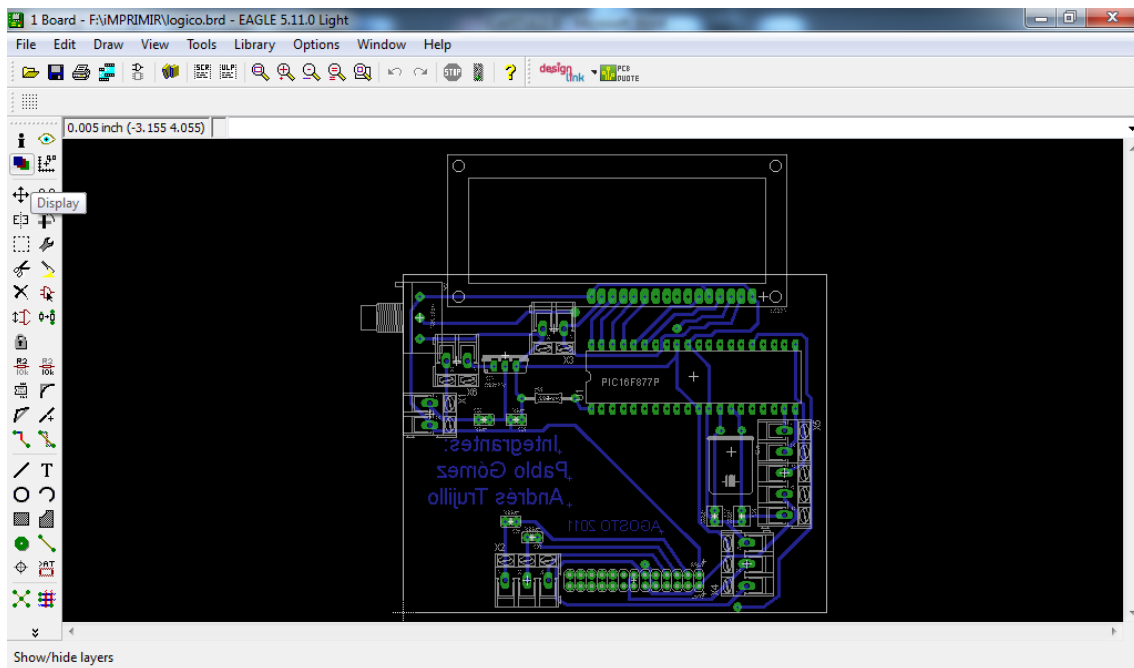


Figura 2.20 Circuito Board.

11. A continuación se debe mostrar solo las pistas y los orificios de la placa, esto podemos realizarlo con la ayuda de la barra de herramientas en Display que apaga las capas no deseadas para el circuito impreso en la placa.

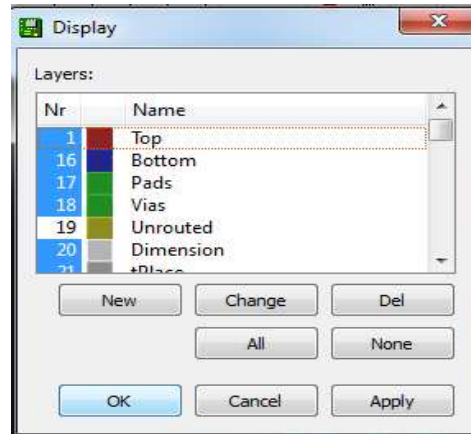


Figura 2.21 Configuración del display.

12. A continuación se muestra el circuito que va a estar impreso en la placa.

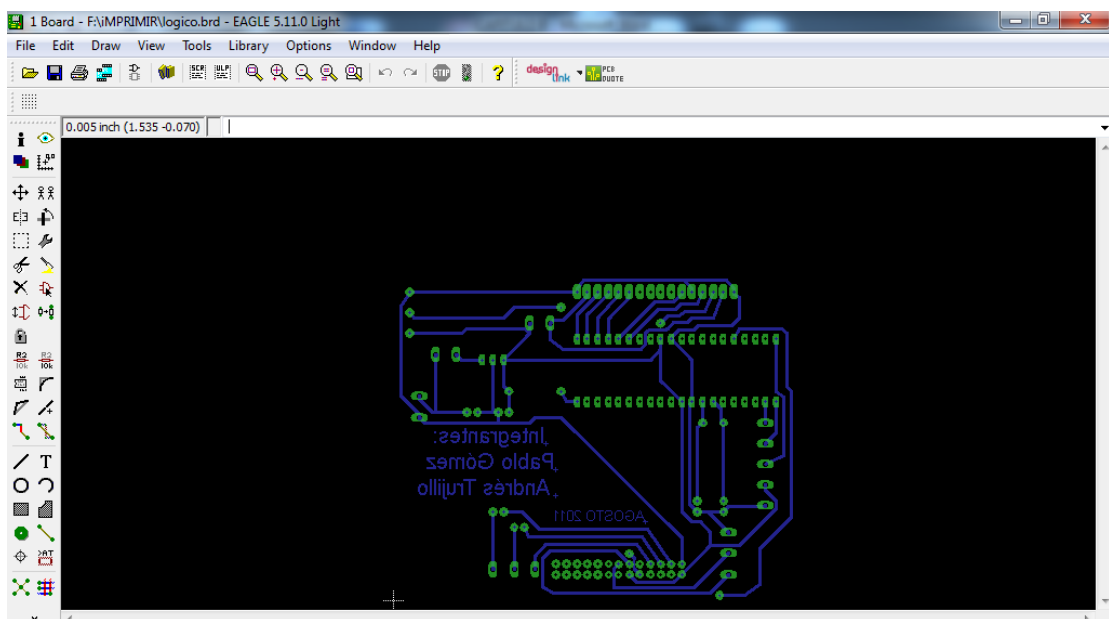


Figura 2.22 Pistas de la placa principal.

13. A continuación se procede a imprimir en la lámina de transferencia térmica como se muestra en la siguiente figura.

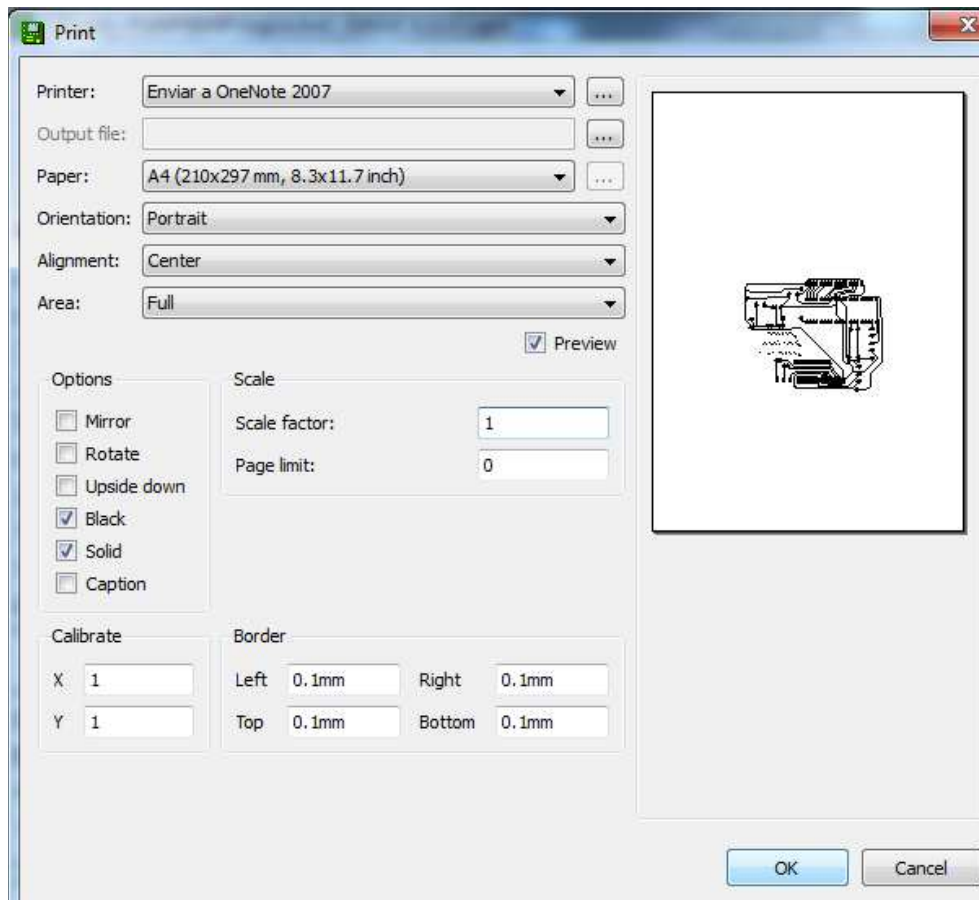


Figura 2.23 Configuración de la impresión del circuito.

2.12. REALIZACION DEL CIRCUITO IMPRESO.

Para la realización del circuito impreso se necesita del gráfico de las pistas para los elementos, esto se consigue con ayuda de un computador, y de un software como Eagle, Ares u Orcad.

- 1) Se imprime el circuito board en una lámina de transferencia térmica las pistas y el screen sobre el lado brillante, es importante utilizar en la impresión una impresora láser o foto copiadora, si no se tendrá problemas para su transferencia en la placa.

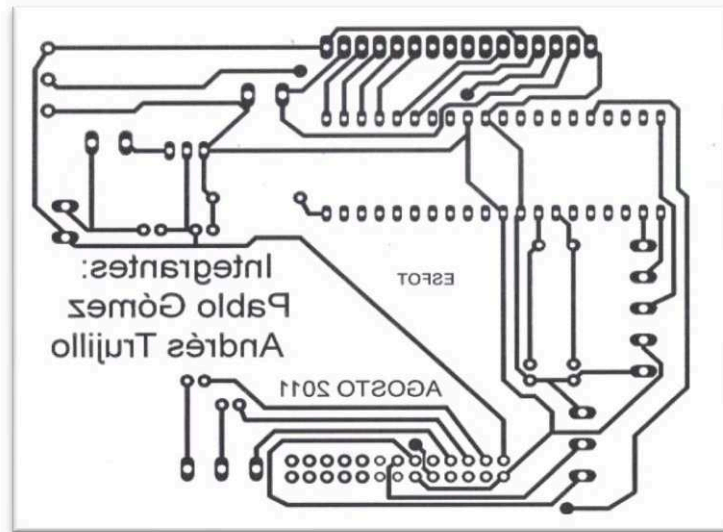


Figura 2.24 Pista impresa de la placa principal

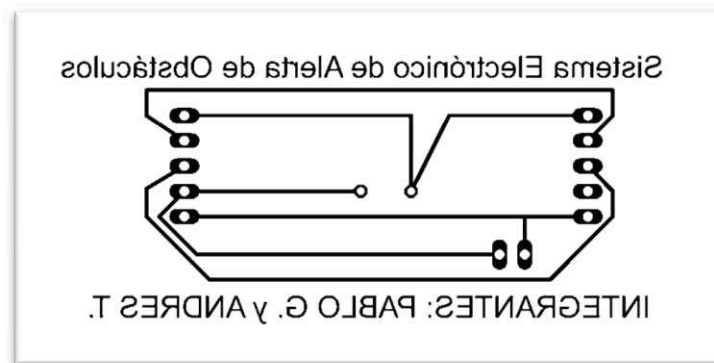


Figura 2.25 Pista impresa de la placa del control manual del sensor.

2) Recortar el área en la que se encuentran las imágenes.



Figura 2.26 Placa a utilizar.

- 3) Limpiar con una esponja de acero (lustre) y jabón de platos el lado de cobre donde se va transferir las pistas, es importante para su transferencia no tocar la superficie con los dedos, ya poseemos grasa.



Figura 2.27 Placa limpia y lista para la transferencia.

- 4) Colocar la lámina en la superficie de cobre de la placa, de modo que se haga contacto la tinta con el cobre.



Figura 2.28 Placa lista para el planchado.

- 5) Con una plancha a su máxima temperatura y con ayuda de un trozo de tela, ejercemos presión fuertemente por unos 30 a 60 segundos, luego de esto lo retiramos para ser enfriado y posteriormente retirar la lámina con cuidado.



Figura 2.29 Proceso de planchado.

- 6) Atacar a la placa con mezcla de cloruro férrico y agua para eliminar el cobre restante e innecesario.



Figura 2.30 Quemado de la placa con acido.

- 7) Limpiar la placa con agua y jabón de platos

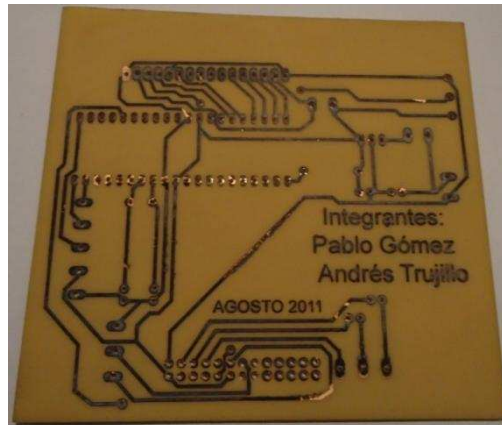


Figura 2.31 Placa limpia.

8) Perforar los orificios con ayuda de una broca U64 y un taladro



Figura 2.32 Taladro para perforar la placa.

9) Realizar el screen; se imprime en la lámina de transferencia térmica los elementos de la placa, para luego planchar en la parte posterior de la placa.

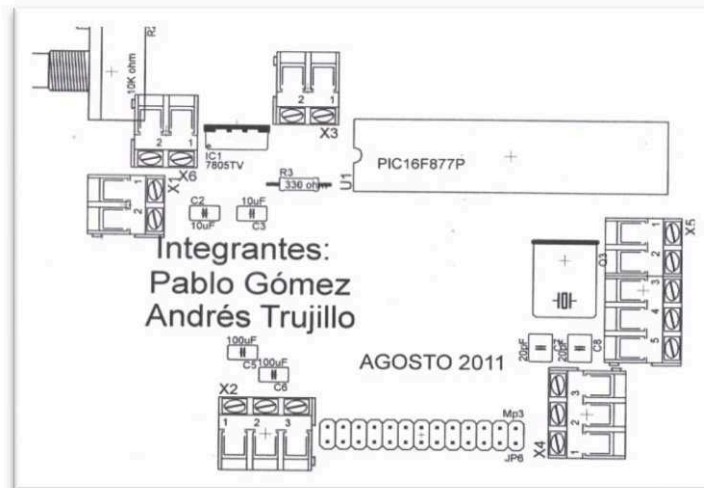


Figura 2.33 Screen impreso.

10) Se desprende la lámina ya planchada con cuidado.



Figura 2.34 Retirar la lámina.

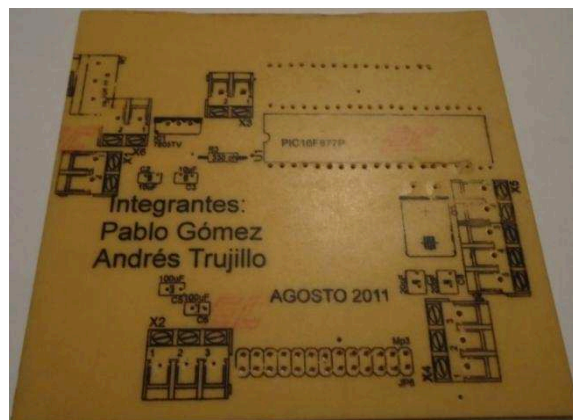


Figura 2.35 Screen transferido a la placa.

11) Se suelda los elementos en el circuito impreso de la placa.



Figura 2.36 Elementos soldados.

12) Finalmente la placa con sus elementos.



Figura 2.37 Placa finalizada y colocados todos sus elementos.

2.14. DESCRIPCION DEL CONTROL ULTRASONIDO.



Figura 2.39 Control de Ultrasonido.

En la figura 2.39 podemos ver que este dispositivo posee un pulsador que permite obtener la información de la distancia del obstáculo en la dirección que se esté apuntando, el sistema dará avisos solo cuando el usuario presione el pulsador caso contrario el sistema permanecerá en reposo.

CAPITULO III.

SOFTWARE PARA EL SISTEMA DE ALERTA DE OBSTÁCULOS PARA NO VIDENTES.

3.1. INTRODUCCIÓN.

El diseño del software para el proyecto es una de las partes más importantes ya que mediante este, el microcontrolador administra y gobierna los dispositivos conectados a él como son el sensor, el LCD y el módulo de voz.

Existen varios lenguajes de programación pero el que se escogió para el proyecto es el lenguaje C ya que es uno de los lenguajes más fáciles de implementar y de usar en la actualidad.

3.2. PROCESOS PARA PROGRAMAR UN MICROCONTROLADOR.

La programación de un microcontrolador se la debe hacer por varios procesos, tal como se muestra en la figura 3.1 luego de escribir el código del programa en lenguaje C lo hacemos pasar por un compilador para que este lo convierta en lenguaje assembler, el compilador que se uso es el MikroC propio de Microchip. Luego de todo esto se convierte finalmente a lenguaje hexadecimal que es el que entiende el PIC, la programación se la hace a través de un programador el cual se consigue fácilmente en cualquier tienda de electrónica.

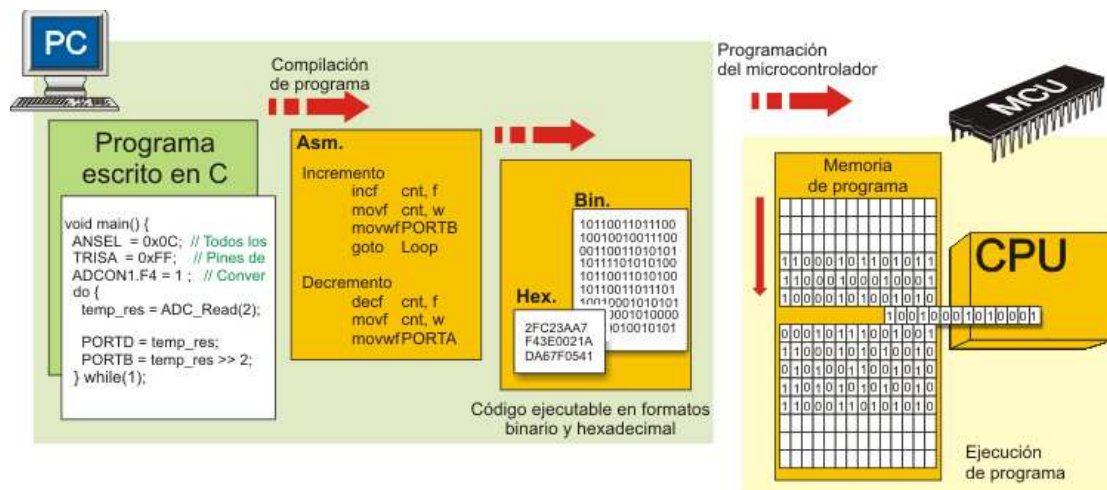


Figura 3.1 Procesos para programar un microcontrolador.

3.3. LENGUAJE DE PROGRAMACION.

3.3.1 MIKROC™.

MikroC™ es una poderosa herramienta de desarrollo para microcontroladores PIC. Está diseñado para proporcionar al usuario la solución más simple en el desarrollo de aplicaciones para sistemas embebidos, sin comprometer el rendimiento o el control. Los PICs y el lenguaje C se acoplan perfectamente: los PICs son los microcontroladores de 8 bits más populares actualmente, empleados en una amplia variedad de aplicaciones, y el lenguaje C, muy apreciado por su eficiencia, es la elección natural para el desarrollo de sistemas.

MikroC™ se caracteriza por su IDE avanzado, compilador conforme a las normas ANSI, amplio conjunto de librerías para hardware, documentación muy fácil de comprender, y muchos ejemplos listos para su ejecución.

3.3.1.1. Características de MikroC™.

Le permite al usuario desarrollar rápidamente aplicaciones complejas:

—

- Escribir el código fuente en lenguaje C usando el Editor de Código incorporado (Asistentes de Parámetros y Código, SyntaxHighlighting, Auto Corrección, Plantillas de Código, y más).
- Usar las librerías incluidas para agilizar drásticamente el proceso de desarrollo: adquisición de datos, memoria, displays, conversiones, comunicaciones, etc. Monitorear la estructura del programa, variables, y funciones en el Explorador de código.
- Generar lenguaje ensamblador y archivos HEX estándar, compatibles con todos los programadores.
- Inspeccionar el flujo del programa y depurarlo con el Simulador (en Software).
- Obtener reportes detallados: mapas RAM y ROM, estadísticas de código, archivos de listado, árbol de llamadas, y más.

Para crear un proyecto en MikroC se escoge primeramente el PIC que se va a programar ya que es importante definirlo desde un principio porque el programa es de acuerdo al dispositivo que se va a usar, además se configura la frecuencia de reloj y los registros que se va a utilizar.

En la figura 3.2 se observa la creación de un nuevo proyecto, se usara el PIC16F877A, una frecuencia de reloj de 12MHz y los registros por default.

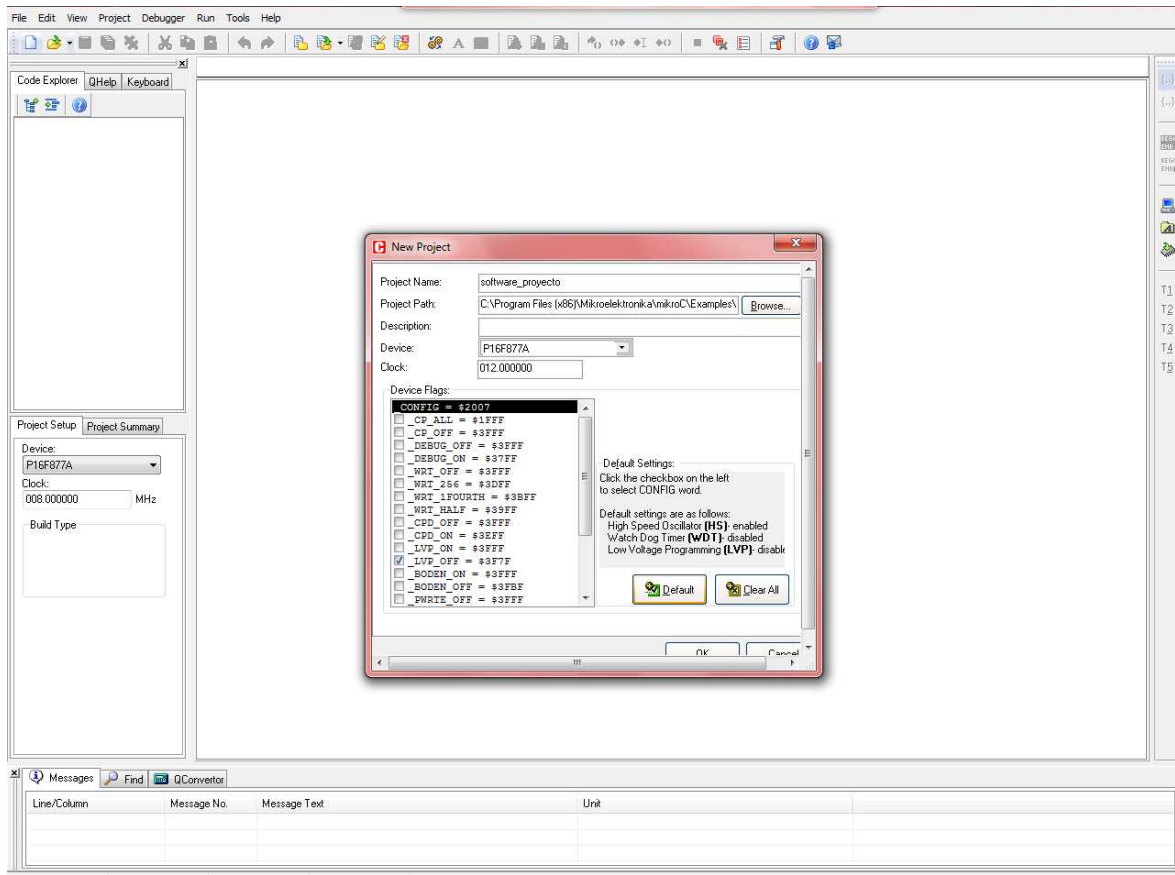


Figura 3.2 Ventana de MikroC.

3.3.2. LENGUAJE DE PROGRAMACION.

MikroC usa un lenguaje C con ciertas características y funciones propias de este compilador que difieren de otros compiladores, a pesar de que se basan también en lenguaje C. A continuación se da una breve explicación de los operadores, tipo de datos, sentencias y funciones que se usa en mikroC, cada una con el formato respectivo.

3.3.2.1. Estructura de programa.

La idea principal de escribir un programa en C es de “romper” un problema mayor en varios trozos más pequeños, la idea general es cada uno se puede escribir como una sola función. Todos los programas escritos en mikroC contienen por lo menos una función llamada main() que encierra entre llaves {} las sentencias a ser ejecutadas. Esto es la primera función a ser ejecutada al iniciarse la ejecución de programa. Las otras funciones se pueden llamar

dentro de la función main. En otras palabras, podemos decir que la función main() es obligatoria, mientras que las demás son opcionales.

La figura muestra la estructura de programa, señalando las partes en las que consiste.

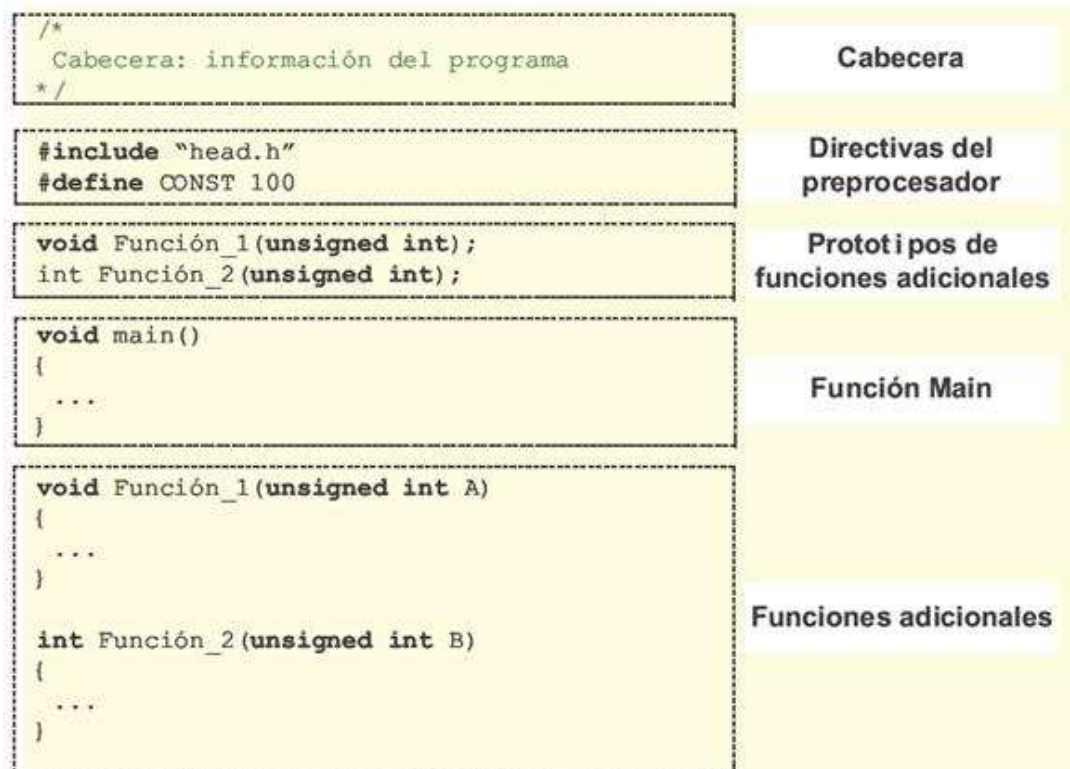


Figura 3.3 Estructura de un programa en MikroC.

La manera de escribir el código en C es muy importante. Por ejemplo, C difiere entre minúsculas y mayúsculas, así que la función main() no se puede escribir MAIN() o Main(). Además, note que dos líneas del código dentro de la función terminan con un punto y coma. En C todas las sentencias deben terminar con un punto y coma ';', así el compilador puede aislarlas y traducirlas a código máquina.

3.3.2.2. Comentarios.

Los comentarios son las partes del programa utilizados para aclarar las instrucciones de programa o para proporcionar más información al respecto. El compilador no hace caso a los comentarios y no los compila al código ejecutable.

Se tiene 2 formatos:

Formato C

```
/* aquí escriba su comentario
```

```
hasta ... */
```

Formato C++

```
// línea de comentario
```

```
// otra línea de comentario
```

MikroC acepta los dos formatos.

3.3.2.3. Operadores aritméticos.

En la tabla 3.1 se tienen los principales operadores aritméticos usados en mikroC cada uno con su respectiva descripción. Los operadores aritméticos se utilizan en las operaciones aritméticas y siempre devuelven resultados numéricos.

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Retorna el residuo de la división entre

	enteros
++	Incrementa en uno al valor del operando
--	Decrementa en uno al valor del operando

Tabla 3.1 Operadores aritméticos.

3.3.2.4. Operadores de relación.

Los operadores relacionales se utilizan en comparaciones con el propósito de comparar dos valores. En mikroC, si una expresión es evaluada como falsa (false), el operador devuelve 0, mientras que si una oración es evaluada como verdadera (true), devuelve 1. En la tabla 3.2 se tienen los operadores lógicos usados en mikroC.

Operador	Operación
==	Igual
!=	No es igual
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

Tabla 3.2 Operadores de relación.

3.3.2.5. Operadores de bits.

A diferencia de las operaciones lógicas que se realizan sobre los valores o expresiones, las operaciones de manejo de bits se realizan sobre los bits de un operando. Se enumeran en la tabla 3.3.

Operador	Operación
&	AND entre bits
	OR entre bits
^	XOR entre bits
~	Complementa bits
<<	Rota bits a la izquierda
>>	Rota bits a la derecha

Tabla 3.3 Operadores de bits.

3.3.2.6. Operadores lógicos.

Hay tres tipos de operaciones lógicas en el lenguaje C: Y (AND) lógico, O (OR) lógico y negación - NO (NOT) lógico. Los operadores lógicos devuelven verdadero (1 lógico) si la expresión evaluada es distinta de cero. En caso contrario, devuelve falso (0 lógico) si la expresión evaluada equivale a cero. En la tabla 3.4 se observan los operadores lógicos usados en MickyC.

Operador	Operación
&&	AND lógico
	OR lógico
!	Negación lógica

Tabla 3.4 Operadores lógicos.

3.3.2.7. Tipos de variables.

En el lenguaje C, los datos tienen un tipo, o sea, cada dato utilizado en el programa debe tener su tipo especificado. Esto permite al compilador conocer el tamaño de dato (número de bytes requerido en la memoria) y su representación. Hay varios tipos de datos que se pueden utilizar en el lenguaje de programación mikroC dependiendo del tamaño de dato y del rango de valores. La tabla 3.5 muestra el rango de valores que los datos pueden tener cuando se utilizan en su forma básica.

Type	Size	Range
(unsigned) char	8-bit	0 .. 255
signed char	8-bit	- 128 .. 127
(signed) short (int)	8-bit	- 128 .. 127
unsigned short (int)	8-bit	0 .. 255
(signed) int	16-bit	-32768 .. 32767
unsigned (int)	16-bit	0 .. 65535
(signed) long (int)	32-bit	-2147483648 .. 2147483647
unsigned long (int)	32-bit	0 .. 4294967295
float	32-bit	±1.17549435082E-38 .. ±6.80564774407E38
double	32-bit	±1.17549435082E-38 .. ±6.80564774407E38
long double	32-bit	±1.17549435082E-38 .. ±6.80564774407E38

Tabla 3.5 Tipo de variables.

Lo que está en paréntesis puede suprimirse.

3.3.2.8. Sentencias.

Las sentencias básicamente se dividen en:

- Sentencias de etiqueta
- Sentencias de selección
- Sentencias de iteración
- Sentencias de salto

- Sentencias compuestas o bloques

3.3.2.8.1. Sentencias de etiqueta.

Sintaxis:

etiqueta_identificadora: sentencia;

Ejemplo:

repita:  etiqueta

a = a + 3;

3.3.2.8.2. Sentencias de selección.

Son dos: if ,switch

- **Sentencia if.**

Sintaxis:

if (expresión) sentencia1

[else sentencia2]

Ejemplo:

if (var == 1) x = x + 5 ;

else y = y - 3 ;

- **Sentencia switch.**

Sintaxis:

switch (expresión) {

case constante_1: sentencia_1 ;

caseconstante_n: sentencia_n ;


```
[default :sentencia ;]  
  
}
```

Ejemplo:

```
switch (input) {  
  
case 1 : LED1 = 1;  
  
case 2 : LED2 = 1;  
  
default : LED7 = 1;  
  
}
```

3.3.2.8.3. Sentencias de iteración.

– Sentencia while.

Sintaxis:

```
while (expresión) sentencia
```

Ejemplo:

```
int s, i ;  
  
s = i = 0  
  
while ( i < 6 ) {  
  
s = s + 2;  
  
i = i + 1;  
  
}
```

– Sentencia do.

Sintaxis:

do sentencia while (expresión)

Ejemplo:

```
int s, i ;

s = i = 0 ;

do {

s = s + 2 ;

i = i + 1 ;

} while ( i < 7 ) ;
```

– **Sentencia for.**

Sintaxis:

for ([exp_inicial] ; [exp_condicional] ; [exp_incremento]) sentencia

Ejemplo:

```
for (sum = 0 , cont = 0 ; cont < 5 ; cont++) {

sum += 2 ;

}
```

3.3.2.8.4. Sentencias de salto.

– **Sentencia break.**

Ejemplo:

```
int i = 0 , s = 1 ; // declaración e inicialización de variables

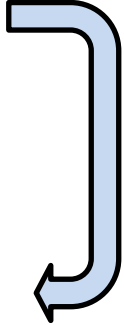
while (1) { // lazo infinito

    if ( i == 4 ) break ;
```

```

    s = s * 2 ;
    i++ ;
}

```



Cuando i = 4

– **Sentencia continue.**

Ejemplo:

```
int i = 0 , s = 1 ; // declaración e inicialización de variables
```

```
while (1) { // lazo infinito
```

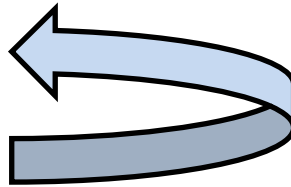
```
s = s * 2 ;
```

```
i++ ;
```

```
if ( i != 4 ) continue ;
```

```
break ;
```

```
}
```



Cuando i ≠ 4

– **Sentencia goto.**

Sintaxis:

Goto etiqueta ;

Ejemplo:

```
    siga : // etiqueta
```

```
    ..... ;
```

```
    goto siga ;
```

3.4 DIAGRAMAS DE FLUJO DEL PROGRAMA.

En las figuras 3.4, 3.5, 3.6 y 3.7 se observan los diagramas de flujo que permitieron la realización del código de programa.

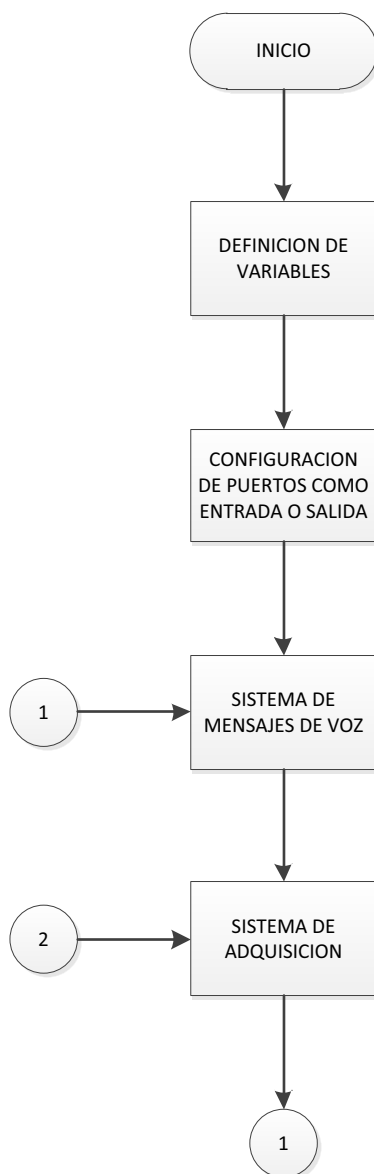


FIGURA 3.4 Diagrama de flujo del sistema de control.

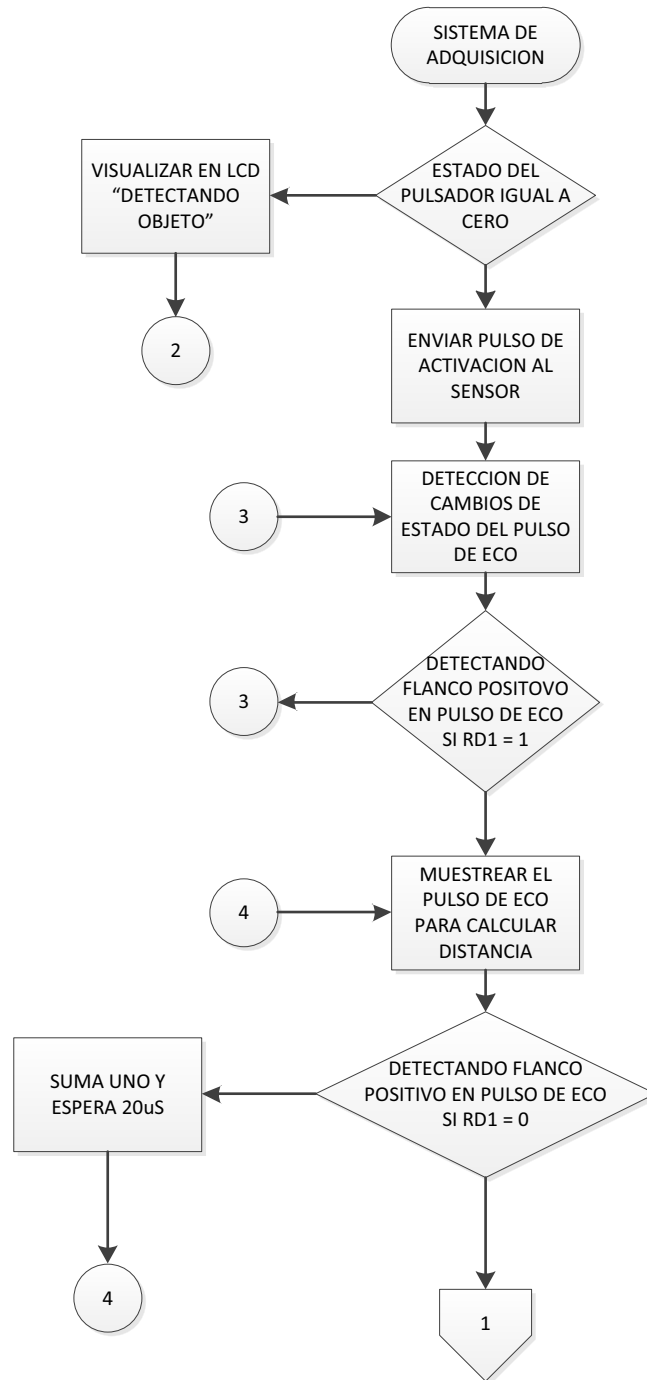


FIGURA 3.5 Diagrama de flujo del sistema de adquisición.

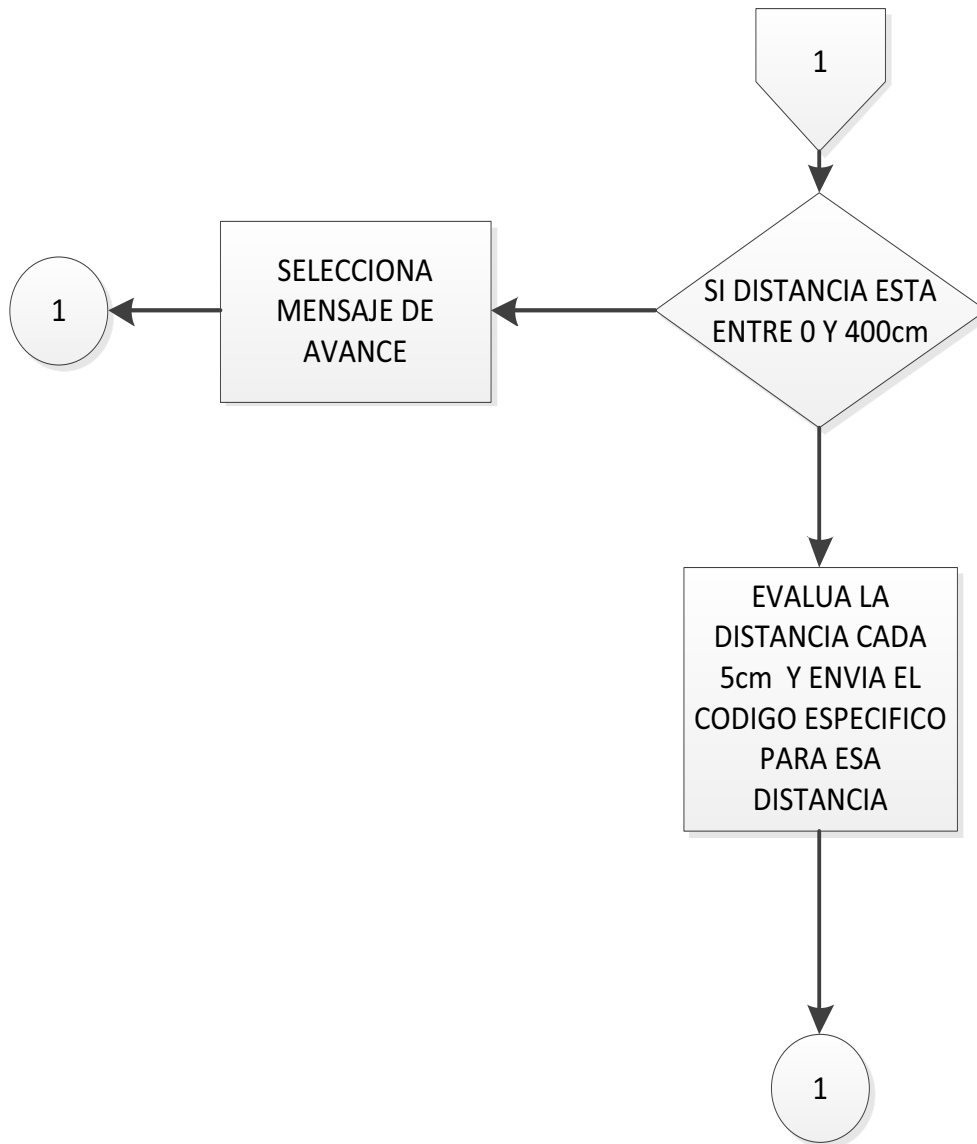


FIGURA 3.6 Diagrama de flujo del sistema de adquisición.

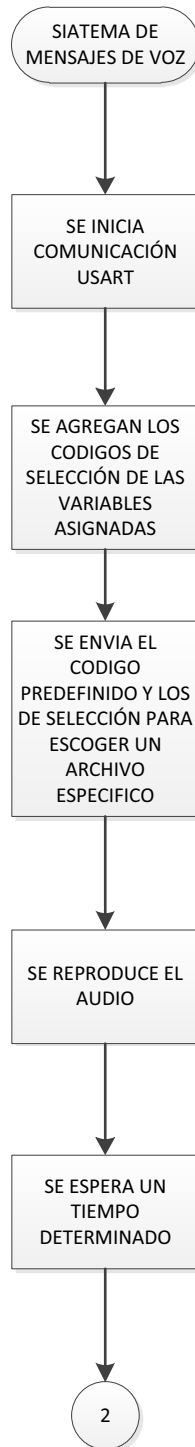


FIGURA 3.7 Diagrama de flujo del sistema de mensajes de voz.

3.5. CODIGO DE PROGRAMA PARA EL PIC16F887A.

/*ESCUELA POLITECNICA NACIONAL

ESCUELA DE FORMACION TECNOLOGICA "ES.FO.T"

TECNOLOGIA EN ELECTRONICA Y TELECOMUNUCACIONES

TEMA: Implementación de un sistema electrónico de alerta de obstáculos para personas no videntes mediante el uso de un sensor.

INTEGRANTES:

PABLO GOMEZ

ANDRES TRUJILLO

TUTOR: ING. PABLO LOPEZ

AÑO: 2011

*/

```
void main() {
```

```
int cont2,num,u,d;
```

```
char text2[6],txt[6];
```

```
TRISB = 0; // PORTB salida
```

```
TRISC =1;
```

```
TRISD.F1=1; // Pin RD1 entrada
```

```
TRISD.F2=1;
```

```
TRISD.F0=0; // Pin RD0 entrada
```

```
PORTD.F0=0;
```



```

Lcd_Init(&PORTB);           // Inicializa el LCD conectado a PORTB

Lcd_Cmd(Lcd_CLEAR);        // Borra display

Lcd_Cmd(Lcd_CURSOR_OFF);  // Cursor off

{

Usart_Init(9600);          //iniciar comunicación USART

Usart_Write(0x7E);        //enviar código por TX para poner play en
                           // el módulo de voz

Usart_Write(0X02);

Usart_Write(0XA0);

Usart_Write(0x7E);

delay_ms(1650);

d=0x32;                    //se selecciona el mensaje 28
"bienvenida"

u=0x38;

goto play;

}

pulso:

    if(PORTD.F2==0)        // verifica el estado del pulsador de
                           // activación de lectura de distancia

    {

        cont2=0;          // envía el pulso de activación al SRF05

        PORTD.F0=1;

```

```

delay_us(10);                //duración de 10uS

PORTD.F0=0;

rep1:

if(PORTD.F1==1)

{

con:

if(PORTD.F1 == 0)           //detectar el flanco positivo en RD1

{

    cont2 = cont2/3;

    cont2=cont2*30;

    cont2=cont2/24;

if(0<cont2 && cont2<401) {           //pregunta si la lectura es mayor a 0
                                     y menor a 401 cm

Lcd_Cmd(Lcd_CLEAR);                // Borra display

byteToStr(cont2, text2);           // convierte el contenido de cont2 al ASCII

Lcd_Out(2,1,text2);                // correspondiente para verlo en el LCD

Lcd_Out(2,6,"CM");                // correspondiente para verlo en el LCD

                                     if(0<cont2 && cont2<5) {           //pregunta si la lectura es mayor a 0
                                     y menor a 5 cm

```

```
d=0x30; //selecciona el código para mensaje 1

u=0x31;

goto play;

    }

Else if(5<=cont2 && cont2<10) { //pregunta si la lectura es
                                mayor a 5 y menor a 10 cm

d=0x30; //selecciona el código para mensaje 2

u=0x32;

goto play;

    }

Else if(10<=cont2 && cont2<15) { //pregunta si la lectura es
                                mayor a 10 y menor a 15 cm

d=0x30; //selecciona el código para mensaje 3

u=0x33;

goto play;

    }

Else if(15<=cont2 && cont2<20) { //pregunta si la lectura es
                                mayor a 15 y menor a 20 cm

d=0x30; //selecciona el código para mensaje 4

u=0x34;
```

```
goto play;

        }

Else if(20<=cont2 && cont2<25) {           //pregunta si la lectura es
                                           mayor a 20 y menor a 25
                                           cm

d=0x30;                                     //selecciona el código para mensaje 5

u=0x35;

goto play;

        }

Else if(25<=cont2 && cont2<30) {           //pregunta si la lectura es
                                           mayor a 25 y menor a 30 cm

d=0x30;                                     //selecciona el código para mensaje 6

u=0x36;

goto play;

        }

Else if(30<=cont2 && cont2<35) {           //pregunta si la lectura es
                                           mayor a 30 y menor a 35 cm

d=0x30;                                     //selecciona el código para mensaje 7

u=0x37;

goto play;
```

```
    }  
  
    Else if(35<=cont2 && cont2<40) {                               //pregunta si la lectura es  
                                                                    mayor a 35 y menor a 40 cm  
  
    d=0x30;                                                         //selecciona el código para mensaje 8  
  
    u=0x38;  
  
    goto play;  
  
    }  
  
    Else if(40<=cont2 && cont2<45) {                               //pregunta si la lectura es  
                                                                    mayor a 40 y menor a 45 cm  
  
    d=0x30;                                                         //selecciona el código para mensaje 9  
  
    u=0x39;  
  
    goto play;  
  
    }  
  
    Else if(45<=cont2 && cont2<50) {                               //pregunta si la lectura es  
                                                                    mayor a 45 y menor a 50 cm  
  
    d=0x31;                                                         //selecciona el código para mensaje 10  
  
    u=0x30;  
  
    goto play;  
  
    }  
  
    }
```



```

Else if (300<=cont2 && cont2<350){                                     //pregunta si la lectura
                                                                    es mayor a 300 y
                                                                    menor a 350 cm

Lcd_Cmd(Lcd_CLEAR);          // Borra display

num=cont2%100;

Lcd_Out(2,1,"3,");          // correspondiente para verlo en el LCD

ByteToStr(num, txt);

Lcd_Out(2,3,txt);          // correspondiente para verlo en el LCD

Lcd_Out(2,10,"Metros");

d=0x32;                    //selecciona el código para mensaje 25

u=0x35;

goto play;

        }

Else if (350<=cont2 && cont2<400){                                     //pregunta si la lectura
                                                                    es mayor a 350 y
                                                                    menor a 400 cm

Lcd_Cmd(Lcd_CLEAR);          // Borra display

num=cont2%100;

Lcd_Out(2,1,"3,");          // correspondiente para verlo en el LCD

ByteToStr(num, txt);

Lcd_Out(2,3,txt);          // correspondiente para verlo en el LCD

Lcd_Out(2,10,"Metros");

d=0x32;                    //selecciona el código para mensaje 26

```

```
u=0x36;

goto play;

    }

else {          //selecciona el código para mensaje 27

    d=0x32;

    u=0x37;

goto play;

    }

}

else {          //selecciona el código para mensaje 27

    d=0x32;

    u=0x37;

goto play;

    }

}

else {

cont2++;      //se incrementa en 1 cada 20us el valor de cont2 con el fin de
              ircontando el tiempo que el pulso de eco está en un valor alto

delay_us(20);

goto con;

    }
```


cuando no se ha
activado el sistema de
adquisición

```
goto pulso;
```

```
}
```

```
}
```

CAPITULO IV

PRUEBAS DE FUNCIONAMIENTO DEL PROYECTO.

4.1. INTRODUCCION.

Luego de la implementación de hardware y software del proyecto se realizó las pruebas ya que con estas se comprobó si el sistema funcionaba y cumplía los requerimientos planteados. Lo principal que se evaluó es el sistema que calculo correctamente la distancia y que esta coincido con la distancia visualizada en el LCD. También se evaluó el sistema de voz reproduzca el mensaje correcto dentro del rango de tolerancia establecido por nosotros.

4.2PRUEBAS DE FUNCIONAMIENTO.

Las pruebas que se realizaron fueron 3:

- Funcionamiento del LCD
- Funcionamiento del sensor
- Funcionamiento del módulo de voz

4.2.1. FUNCIONAMIENTO DEL MÓDULO LCD

Para el funcionamiento correcto del LCD se calibró el contraste para una buena visualización. Como ya se mencionó el LCD es solo con fin de mantenimiento por lo que una correcta visualización es muy importante para el técnico encargado del mantenimiento. Este debe permitir visualizar de forma clara y además tiene que mostrar las mediciones sin equivocaciones sino el técnico no podrá llevar un correcto mantenimiento. En la figura 4.1 se puede observar claramente lo que nos muestra el LCD.



Figura 4.1 Prueba del LCD.

4.2.2. FUNCIONAMIENTO DEL SENSOR ULTRASONIDO SRF 05.

El funcionamiento correcto del sensor SRF05 se lo comprobó mediante la medición de la distancia a un objeto cercano, se usó el flexómetro para poder comprobar que las lecturas de las distancias sean las correctas comparadas con las visualizadas en el LCD. En la figura 4.2 se hace la prueba con el flexómetro.



Figura 4.2 Prueba con el flexómetro a 54cm.

Presionando el pulsador, el sensor entra en funcionamiento, para lo cual se realizó pruebas del funcionamiento para la comunicación con el microcontrolador.

	PRUEBA		Detección de obstáculo del sensor	
	DISTANCIA		54 cm	
Número	Prueba realizada	Resultado	Observaciones	Objetivo alcanzado
1	Se comparó con la medición del flexómetro y el LCD, a una distancia de 54 cm.	Como resultado se obtuvo que la distancia visualizada en el LCD es igual a la medida con el flexómetro.	Se obtuvo el resultado deseado con un mínimo de error.	Si

Tabla 4.1 Detección de obstáculo del sensor.

4.2.3. FUNCIONAMIENTO DEL MÓDULO DE VOZ.



Figura 4.3 Controlador Ultrasonido.

Esta prueba es infaltable y muy necesaria para su funcionamiento del módulo de voz. El objetivo principal es obtener el mensaje de acuerdo con la distancia a la que se encuentre con la configuración establecida en el código programa. En la figura 4.3 se tiene el controlador ultrasónico con el cual se va a hacer la prueba.

	PRUEBA		Reproducción del mensaje	
	DISTANCIA		73 cm	
Numero	Prueba realizada	Resultados	Observaciones	Objetivo alcanzado
1	Se colocó el controlador ultrasónico frente a un objeto a 73 cm y la medida se la comparo con el flexómetro	El módulo de voz reprodujo el mensaje "Setenta y cinco centímetros"	El modulo reprodujo correctamente ya que de 71 a 75 centímetros se programó reproducir el mensaje de 75 centímetros	Si

Tabla 4.2 Reproducción del mensaje.

Las pruebas realizadas nos llevaron a las siguientes conclusiones:

- El sensor ultrasonido mostro buena respuesta al detectar objetos perpendiculares a su lóbulo de radiación pero mostro ciertos problemas con objetos con una inclinación mayor a 30 grados que tiene como tolerancia según las especificaciones técnicas, ya que el pulso ultrasónico no rebotaba de regreso a este o rebotaba en varias superficies, provocando falsas lecturas.
- Además de que el sensor actuó de varias formas dependiendo del material en el que rebotaba, es así que en superficies totalmente planas actuó muy bien pero no lo hizo sobre superficies rugosa.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES.

5.1. CONCLUSIONES.

- Mediante la comunicación USART se logró controlar el módulo de voz para la reproducción de mensajes. Esto resulto muy útil a la hora de implementar todos los dispositivos al microcontrolador ya que se usó un solo pin del microcontrolador (el de transmisión) ya que solo se necesitó que este enviara códigos al módulo de voz para la reproducción de los mensajes.
- Al momento de pasar los archivos mp3 al módulo de voz se los tuvo que hacer de uno en uno debido a que este reconoce el orden en el que se copian a la memoria SD, es decir si el mensaje tres por ejemplo se lo copia primero al momento de que el microcontrolador enviara el código de selección del archivo uno se reproducía el mensaje tres erróneamente, provocando que la distancia real sea diferente a la escuchada por los audífonos. Se tuvo este inconveniente pero se lo soluciono de la manera indicada.
- El desarrollo de este sistema en varias etapas y las pruebas iniciales que se realizó probando cada dispositivo por separado nos ayudó mucho a comprender el funcionamiento de todos los dispositivos, de esta manera se simplifico un poco la implementación.
- Como conclusión final de este proyecto es que es un sistema bastante útil para una persona no vidente ya que le permite movilizarse evitando obstáculos que le pueda hacer daño, detectando los obstáculos en la dirección que él no invidente desee dándole así una oportunidad de tener un estilo de vida normal ya que ellos son personas que se merecen las mismas posibilidades y oportunidades que todos.

5.2. RECOMENDACIONES.

- Se recomienda que la persona no vidente lleve su bastón como segundo recurso ya que si llegara a agotarse la batería en medio de una caminata por ejemplo, la persona tenga otro medio para regresar a casa y cargar la batería.
- Para bajar o subir el volumen en el módulo de voz se debe pulsar al momento que se está reproduciendo cualquier mensaje, porque de otra forma no funcionara esta opción del módulo de voz.
- Se recomienda cargar la batería de 6V de forma separada al circuito en un lugar fresco y con buena ventilación ya que por especificaciones técnicas de los fabricantes de esta es sensible a espacios cerrados como cajas.
- Una recomendación que se hace para posibles mejoras del proyecto es la implementación de otro tipos de sensores que sean capaces de detectar otras variables físicas con el fin de que se pueda saber el tipo de obstáculo que se tiene en frente y además si fuera posible detectar personas y seres vivos en general.
- En cuanto al diseño del sistema electrónico se recomienda el uso del software correcto como el Proteus para la simulación de circuitos, ya que tiene buenos recursos para esto, pero para el diseño de circuito impreso carece de varios elementos. Por lo que se recomienda el software Eagle para obtener el circuito impreso en la placa, ya que este cuenta con librerías extensas para el diseño de un circuito impreso con conectores y borneras fáciles de encontrar.

BIBLOGRAFIA

Libros:

1. BOYLESTAD NASHESKY
Electrónica: teoría de circuitos y dispositivos electrónicos
Octava edición, editorial Person Educación, México, pág.: 930.
2. GARCIA BREIJO, Eduardo
Compilador C CCS y simulador PROTEUS para microcontroladores
PIC
Primera edición, Alfaomega Grupo Editor, S.A. de CV., México, pág.: 65
– 66
3. COSTALES, Alcívar
Apuntes de la materia Control con Microprocesadores

Sitios web:

1. Bastón blanco.
http://www.juntadeandalucia.es/averroes/caidv/interedvisual/ftp/fm_personas_c_y_uso_baston.rtf
2. Eye Stick, bastón con sensor para ayudar a los invidentes.
<http://espegizmo.com/2009/09/08/general/eye-stick-baston-con-sensor-para-ayudar-a-los-invidentes/>
3. Kinect ayuda para ciegos.
<http://beatgeek.net/kinect-ayuda-para-ciegos/>

4. Una vara para ciegos.
<http://www.impresionante.net/10-03-2008/general/una-vara-para-ciegos>

5. Visión de murciélago para ciegos.
<http://www.neoteo.com/vision-de-murcielago-para-ciegos-14686>

6. Comunicación serial.
<http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F0058881>

7. Microcontroladores.
http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf
<http://ingenio-upp.blogdiario.com/>
<http://www.olimex.cl/tutorial/tutorial1.pdf>
<http://www.monografias.com/trabajos12/microco/microco.shtml>
<http://es.wikipedia.org/wiki/Microcontrolador>
http://es.wikipedia.org/wiki/Microcontrolador_PIC

8. USART.
<http://www.ucontrol.com.ar/wiki/index.php?title=USART>

9. Sensores.
<http://www.slideshare.net/diego5wh/sensores-de-distancia>

10. LCD.
http://es.wikipedia.org/wiki/Pantalla_de_cristal_líquido
http://www.tolaemon.com/site/lcds_alfanum%C3%A9ricos_compatibles_hd4478

ANEXOS

**ANEXO 1:
DESCRIPCION DEL
MICROCONTROLADOR PIC16F877A.**



PIC16F87XA
Data Sheet

28/40/44-Pin Enhanced Flash
Microcontrollers



PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F876A
- PIC16F874A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
DC – 200 ns instruction cycle
- Up to 8K × 14 words of Flash Program Memory,
Up to 368 × 8 bytes of Data Memory (RAM),
Up to 256 × 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin
PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during Sleep via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™
(Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) – 8 bits wide with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital
Converter (AD)
- Brown-out Reset (BOR)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference
(VREF) module
 - Programmable input multiplexing from device
inputs and internal voltage reference
 - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash
program memory typical
- 1,000,000 erase/write cycle Data EEPROM
memory typical
- Data EEPROM Retention > 40 years
- Self reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™)
via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

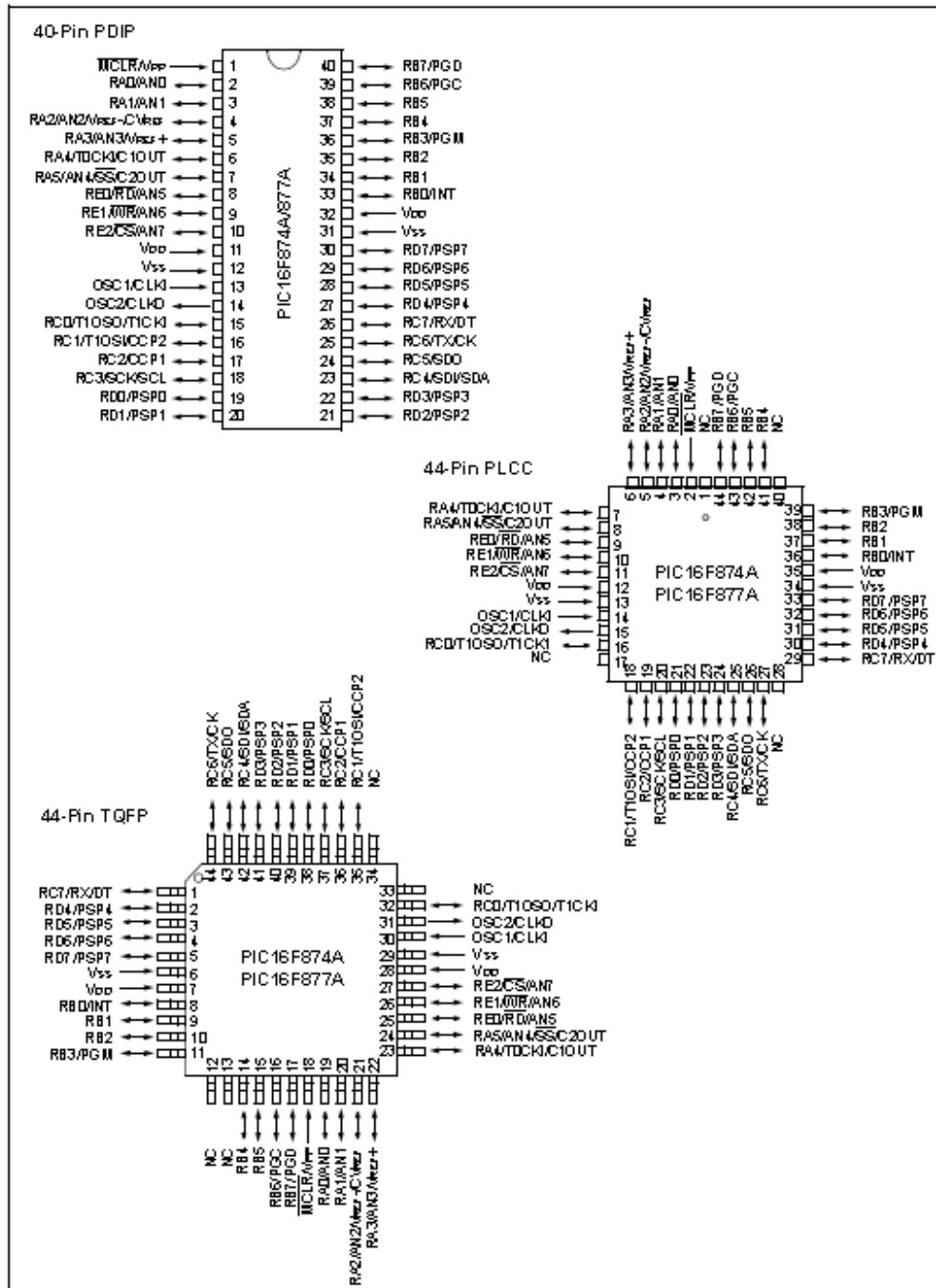
CMOS Technology:

- Low-power, high-speed Flash/EEPROM
technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	WD	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

PIC16F87XA

Pin Diagrams (Continued)



PIC16F87XA

1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC 16F873A
- PIC 16F874A
- PIC 16F876A
- PIC 16F877A

PIC 16F873A/876A devices are available only in 28-pin packages, while PIC 16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC 16F87XA family share common architecture with the following differences:

- The PIC 16F873A and PIC 16F874A have one-half of the total on-chip memory of the PIC 16F876A and PIC 16F877A.
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five.
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen.
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight.
- The Parallel Slave Port is implemented only on the 40/44-pin devices.

The available features are summarized in Table 1-1. Block diagrams of the PIC 16F873A/876A and PIC 16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

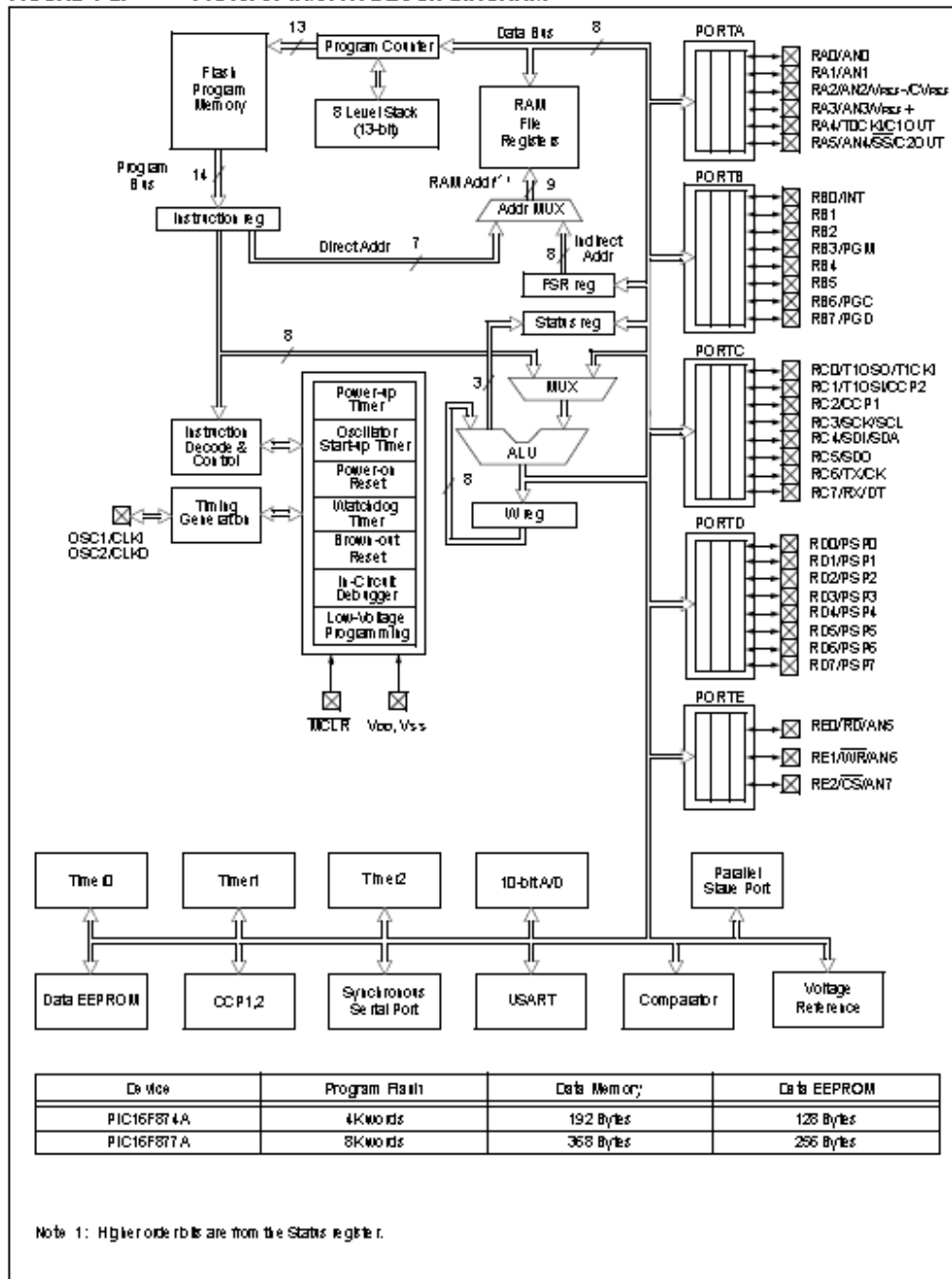
Additional information may be found in the PICmicro[®] Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC 16F873A	PIC 16F874A	PIC 16F876A	PIC 16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

PIC16F87XA

FIGURE 1-2: PIC16F874A/877A BLOCK DIAGRAM



PIC16F87XA

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	33	36	8	9	I/O I	TTL/ST ⁽¹⁾	PORT B is a bidirectional I/O port. PORT B can be software programmed for internal weak pull-up on all inputs. Digital I/O. External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	36	39	11	12	I/O I	TTL	Digital I/O. Low-voltage ICSP programming enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	39	43	16	16	I/O I	TTL/ST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming clock.
RB7/PGD RB7 PGD	40	44	17	17	I/O I/O	TTL/ST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming data.

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F87XA

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	VO/P Type	Buffer Type	Description
RC0/T1OSQ/T1CKI RC0 T1OSQ T1CKI	15	16	32	34	VO O I	ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSVCCP2 RC1 T1OSI CCP2	16	18	35	35	VO I VO	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	17	19	36	36	VO VO	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	18	20	37	37	VO VO VO	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA RC4 SDI SDA	23	25	42	42	VO I VO	ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	24	26	43	43	VO O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	27	44	44	VO O VO	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	26	29	1	1	VO I VO	ST	Digital I/O. USART asynchronous receive. USART synchronous data.

Legend: I = input O = output VO = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

PIC16F87XA

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0 RD0 PSP0	19	21	38	38	I/O I/O	ST/TT L ⁽²⁾	PORT D is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus. Digital I/O. Parallel Slave Port data.
RD1/PSP1 RD1 PSP1	20	22	39	39	I/O I/O	ST/TT L ⁽²⁾	Digital I/O. Parallel Slave Port data.
RD2/PSP2 RD2 PSP2	21	23	40	40	I/O I/O	ST/TT L ⁽²⁾	Digital I/O. Parallel Slave Port data.
RD3/PSP3 RD3 PSP3	22	24	41	41	I/O I/O	ST/TT L ⁽²⁾	Digital I/O. Parallel Slave Port data.
RD4/PSP4 RD4 PSP4	27	30	2	2	I/O I/O	ST/TT L ⁽²⁾	Digital I/O. Parallel Slave Port data.
RD5/PSP5 RD5 PSP5	28	31	3	3	I/O I/O	ST/TT L ⁽²⁾	Digital I/O. Parallel Slave Port data.
RD6/PSP6 RD6 PSP6	29	32	4	4	I/O I/O	ST/TT L ⁽²⁾	Digital I/O. Parallel Slave Port data.
RD7/PSP7 RD7 PSP7	30	33	5	5	I/O I/O	ST/TT L ⁽²⁾	Digital I/O. Parallel Slave Port data.
RE0/RD/AN5 RE0 RD AN5	8	9	25	25	I/O I I	ST/TT L ⁽²⁾	PORT E is a bidirectional I/O port. Digital I/O. Read control for Parallel Slave Port. Analog input 5.
RE1/WR/AN6 RE1 WR AN6	9	10	26	26	I/O I I	ST/TT L ⁽²⁾	Digital I/O. Write control for Parallel Slave Port. Analog input 6.
RE2/CS/AN7 RE2 CS AN7	10	11	27	27	I/O I I	ST/TT L ⁽²⁾	Digital I/O. Chip select control for Parallel Slave Port. Analog input 7.
V _{SS}	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
V _{DD}	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12, 13, 33, 34	13	—	—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

ANEXO 2:
DESCRIPCION DEL MODULO DE VOZ
WT9501M03.

USB-SD MP3 Module Manual

WT9501M03

www.elechoUSE.com

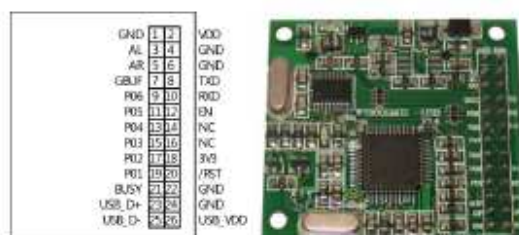
Copyright reserved by elechoUSE



Features

- Can play 8 ~ 320Kbps MP3 audio files;
- Support maximum capacity of 32G Byte SD card;
- Support USB flash disk and SD;
- Support key mode and serial control mode;
- Support direct audio playback of any section;
- Power memory function can be customized (mass order);
- Optional built-in Class D amplifier (3W × 1) output;
- Size: 41mm × 39mm
- Operating voltage: DC5V
- Quiescent Current: 20mA
- Maximum operating current: 70mA

Pin Function



No.	Pin Name	Functional Description
1	GND	GND
2	VCC	DC5V input
3	L	Audio left output
4	GND	Power ground
5	R	Audio right output
6	GND	Power ground
7	GBUF	Audio ground
8	TXD	Serial data transmitter
9	P06	I / O port
10	RXD	Serial data receiver
11	P05	I / O port
12	EN	Power Enable
13	P04	I / O port
14	NC	Vacant (Reserved)
15	P03	I / O port
16	NC	Vacant (Reserved)
17	P02	I / O port
18	3V3	DC3.3V Output

www.elechose.com

19	P01	I/O port
20	/RST	Reset pin
21	BUSY	Busy signal, the output is low when playing
22	GND	Power ground
23	USB_D+	USB_D+input
24	GND	USB ground
25	USB_D-	USB_D-input
26	USB_VDD	USB Power

Difference between GND and GND will be explained later

23 to 26 pin can be used as USB flash disk data pins, also SD card data pins.

SD card format: FAT or FAT32

SD card and USB flash file

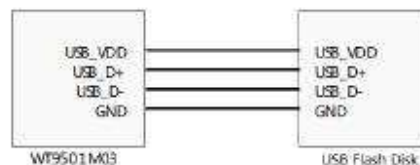
SD card connection

WT9501M03 has SD card slot.



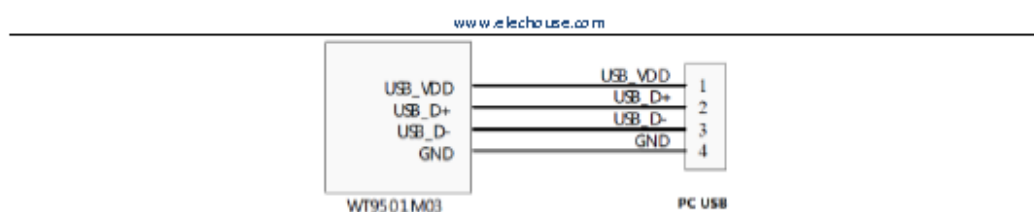
USB flash disk connection (Not support by this version)

Connection diagram is shown below.



SD serves as USB storage media (Not support by this version)

When the SD card is inserted in WT9501M03 module, it can be connected to a computer via USB cable, downloading or upload files.



SD card and USB flash file storage

Mp3 files are stored in root directory of SD card or USB flash, and the file name should begin with 5 digits, such as 00001.mp3, 00002.mp3 and so on. Support maximum 10,000 segments of audio in the SD card and USB flash. However, the more the number of audio files, the longer the time from the trigger to play.

Working Mode

Key mode

In standard mode, I/O P01 ~ P06 is high while standby, negative pulse of 10ms will trigger the pins.

I/O port	P01	P02	P03	P04	P05	P06
Features	Play/Pause	Last	Next	VOL +	VOL-	Stop

Note: After power on or reset, the first time triggering the play/pause button will make play/stop action, and the subsequent triggering will be play/pause.

Serial mode

UART serial communication is based on 9600 baud rate. The following communication protocol is defined, including start code, data length, operating code, data bits and stop code.

Start code	Data length	Operation code	T en thousands digit	Thousands digit	Hundreds digit	Tens digit	Units digit	End code
7E	07	XX	XX	XX	XX	XX	XX	7E

Operation code description

Type	Description	Operation code	Operation data
SD Card	Play (SD card)	A0H	xx xx xx xx xx
	Pause (SD card)	A1H	None
	Play from the pause point (SD card)	A2H	None
	Cease (SD card)	A3H	None
	Volume	A4H	XX
	Last	A5H	None

www.elechouse.com

	Next	A6H	None
	Play one without cycle	A7H	None
	Play all in cycle	A8H	None
	Play one in cycle	A9H	None
USB Flash	Play (USB flash)	B0H	xx xx xx xx xx
	Pause (USB flash)	B1H	None
	Play from the pause point (USB flash)	B2H	None
	Cease (USB flash)	B3H	None
	Volume	B4H	XX
	Last	B5H	None
	Next	B6H	None
	Play one without cycle	B7H	None
	Play all in cycle	B8H	None
	Play one in cycle	B9H	None

Operation code A0 (SD card), B0 (USB flash), the volume A4 (SD card), and B4 (USB flash) need operation data, and the other commands do not need.

WT9501M03 can automatically recognize the MP3 files in SD card and USB flash. It assigns the file number according to creating time of files. File number is a 5-digit number. And this module reads file name in **ASCII code**. For example: 00045.mp3

- ◆ Ten thousands digit is "0", and the ASCII code is "30H"
- ◆ Thousands digit is "0", and the ASCII code is "30H"
- ◆ Hundreds digit is "0", and the ASCII code is "30H"
- ◆ Tens digit is "4", and ASCII code is "34H"
- ◆ Units digit is "5", and ASCII code is "35H"

Start code: 7E

Data Length: the total number of bytes excluding the start code and end code, but including Data Length itself.

End code: 7E

Play Specific Track

If need to play the 45th track in SD card, send the data as follows:

Start code	Data length	Operation code	Ten thousands digit	T thousands digit	Hundreds digit	Tens digit	Units digit	End code
7E	07	A0	30	30	30	34	35	7E

If need to play the 45th track USB flash, send the data as follows

Start code	Data length	Operation code	Ten thousands digit	T thousands digit	Hundreds digit	Tens digit	Units digit	End code
7E	07	B0	30	30	30	34	35	7E

If WT9501M03 contains files both in SD card and USB flash, it can switch playing file from SD card to USB or the reverse. Delay might happen after the demand is sent, depending on the file numbers in the media.

Pause

Pause to play files in SD card:

Start code	Data length	Operation code	End code
7E	02	A1	7E

Pause to play files in USB flash:

Start code	Data length	Operation code	End code
7E	02	B1	7E

Resume playing from the pause point

Resume playing files in SD card

Start code	Data length	Operation code	End code
7E	02	A2	7E

Resume playing files in USB flash

Start code	Data length	Operation code	End code
7E	02	B2	7E

Cease

Cease playing files in the SD card:

Start code	Data length	Operation code	End code
7E	02	A3	7E

Cease playing files in USB flash:

Start code	Data length	Operation code	End code
7E	02	B3	7E

Volume Adjustment

In the volume control command, there are 26 grades of volume from 00H to 19H. 00H is mute, and 19H is the highest.

Play SD card and adjust the volume, send the following data:

Start code	Data length	Operation code	Volume value	End code
7E	07	A4	XX	7E

Play USB flash and adjust the volume, send the following data:

Start code	Data length	Operation code	Volume value	End code
7E	07	B4	XX	7E

Note:

1. After power on or reset, this command is invalid. You need to play the voice first and then send this command.
2. Sending the value greater than 19H, it will adjust the volume to maximum.

Last

Play SD card content and switch to previous one, send the following data:

Start code	Data length	Operation code	End code
7E	02	A5	7E

Play USB flash content and switch to previous one, send the following data:

Start code	Data length	Operation code	End code
7E	02	B5	7E

Next

Play SD card files and switch to next one, send the following data:

Start code	Data length	Operation code	End code
7E	02	A6	7E

Play USB flash content and switch to next one, send the following data:

Start code	Data length	Operation code	End code
7E	02	B6	7E

Play one without cycle

Play SD card files, and stop playing after finishing the voice file. Send the following data:

Start code	Data length	Operation code	End code
7E	02	A7	7E

Play USB flash files, and stop playing after finishing the voice file. Send the following data:

Start code	Data length	Operation code	End code
7E	02	B7	7E

Play one in cycle

Play SD card files, and loop one. Send the following data:

Start code	Data length	Operation code	End code
7E	02	A8	7E

Play USB flash files, and loop one. Send the following data:

Start code	Data length	Operation code	End code
7E	02	B8	7E

Play all in loop

Play SD card files, and loop all. Send the following data:

www.elechouse.com

Start code	Data length	Operation code	End code
7E	02	A9	7E

Play USB flash files, and loop all. Send the following data:

Start code	Data length	Operation code	End code
7E	02	B9	7E

Return Code Description

Return Code is the reply after sending the command. It begins with 7E 7E. Data following the 7E 7E have meaning as follows:

Data Address	Value and function
0X10	High bit of current play address
0X11	Low bit of current play address
0X12	0x01: Play 0x02: Pause 0x03: Cease 0x04: Last 0x05: Next 0x06: Volume adjustment command (read volume value at address 0X15) 0x07: Reserved 0x08: LED display volume (V0-V25) 0x09: LED display song number 0x0a: LED display loop mode (invalid) 0x0b: Standard display (display the current song) 0x0c: Play one without loop 0x0d: Play all in circle 0x0e: Play one in circle 0x0f: LED numeric display off 0x10: LED numeric display on
0X13	0X XX LED display value (reserved)
0X14	0x01: play USB flash songs 0x02: play SD card song
0X15	Volume, 0~25
0X16-0X1F	Reserved
0X20 (return from here)	High bit of current play address in USB flash
0X21	Low bit of current play address in USB flash
0X22	High bit of current play address in SD card
0X23	Low bit of current play address in SD card
0X24	0x00: No play 0x01: USB flash is playing 0x02: SD card is playing
0X25	High byte of MP3 file total numbers
0X26	Low byte of MP3 file total numbers
0X27	Reserved

www.elechouse.com

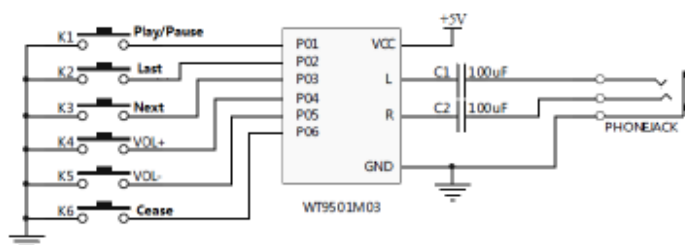
0X28	Volume, 0~25
0X29	0x02: Play all in circle 0x03: Play one without loop 0x04: Play one in circle (forced to 03 after powered on)
0X2A	0x08: LED display the volume 0x09: LED display song number 0x0a: LED display cycle model 0x0b: Standard display (display the current song)
0X2B	The number LED displays, 0~99(invalid, always 0x10)
0X2C	0x01: USB flash connected 0x02: No USB flash connected
0X2D	0x01: SD card connected 0x02: SD card flash connected
0X2E	0x01: Now Playing 0x02: Now Paused 0x03: Now Ceased
0X2F	Reserved
0X30-0X4F	The name of the currently playing music file

Note: Return code in orange-marked row might have error

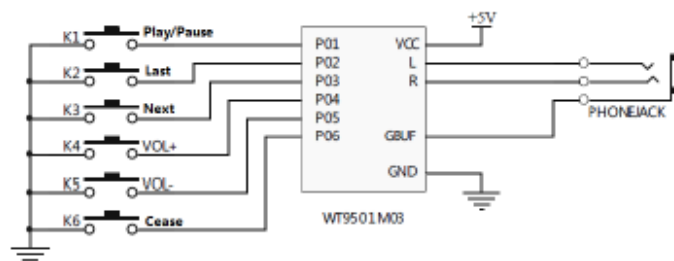
Application Circuit

key mode application circuit

L, R and GND connect to headphones, and audio line output requires series with 100uF capacitor.

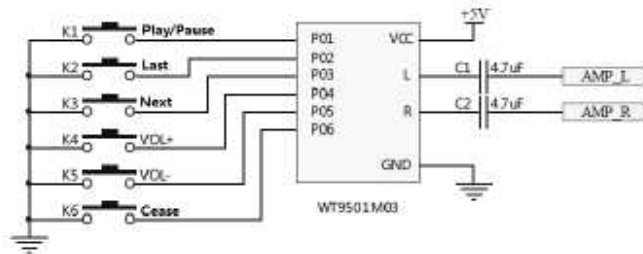


L, R and GBUF connect to headphone:



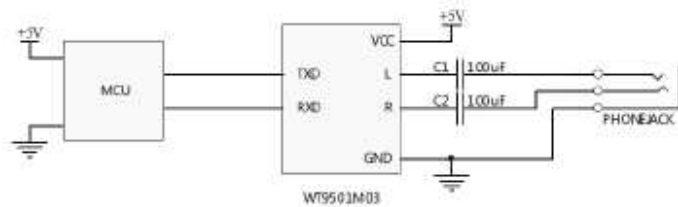
www.lechouse.com

L, R GND connect to external amplifier (GBUF is not recommended):

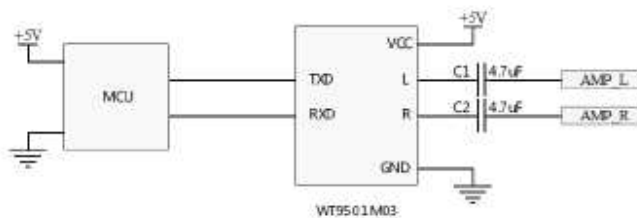


MCU control mode application circuit

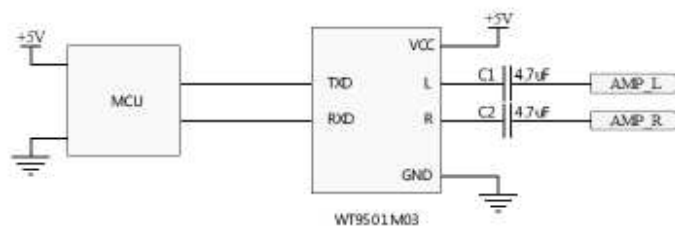
L, R and GND connect to headphones, and audio line output requires series with 100uF capacitor.

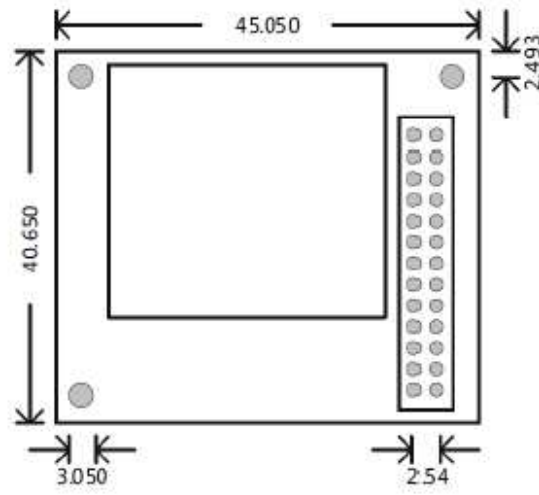


L, R and GBUF connect to headphone:



L, R GND connect to external amplifier (GBUF is not recommended):

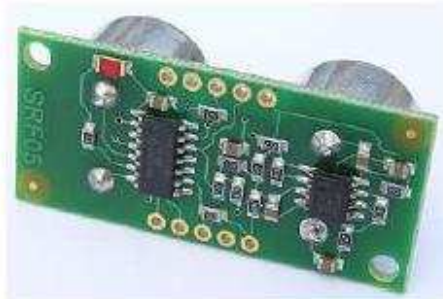


Package dimensions

ANEXO 3:
DESCRIPCION DEL SENSOR SRF05.

SRF05 - Ultra-Sonic Ranger

Technical Specification

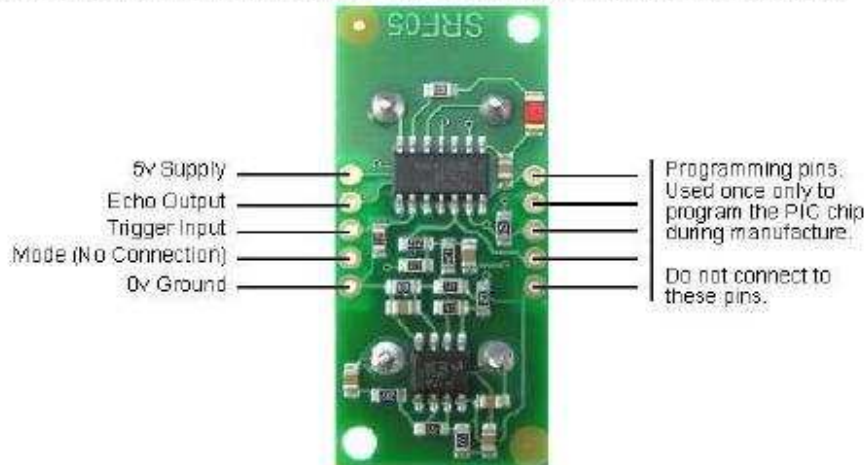


Introduction

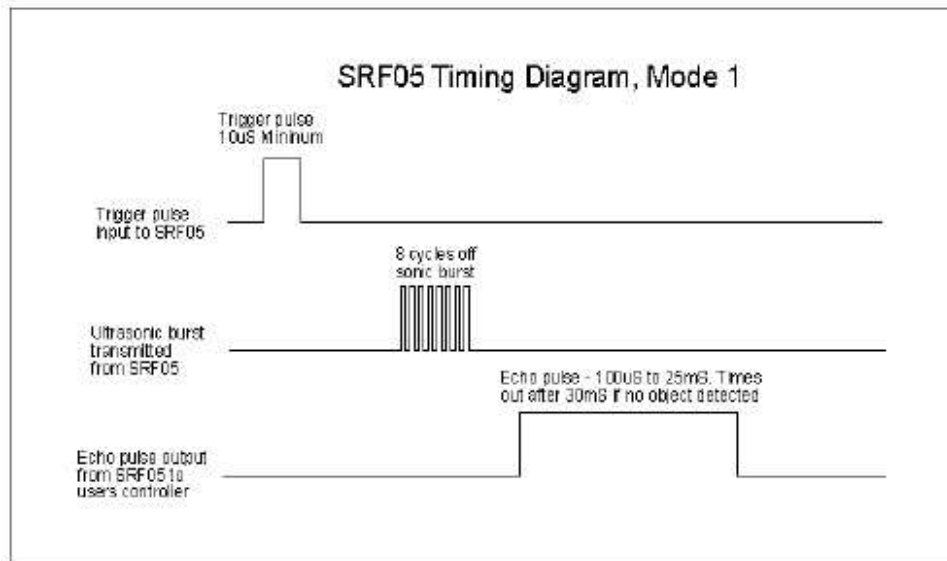
The SRF05 is an evolutionary step from the SRF04, and has been designed to increase flexibility, increase range, and to reduce costs still further. As such, the SRF05 is fully compatible with the SRF04. Range is increased from 3 meters to 4 meters. A new operating mode (tying the mode pin to ground) allows the SRF05 to use a single pin for both trigger and echo, thereby saving valuable pins on your controller. When the mode pin is left unconnected, the SRF05 operates with separate trigger and echo pins, like the SRF04. The SRF05 includes a small delay before the echo pulse to give slower controllers such as the Basic Stamp and Picaxe time to execute their pulse in commands.

Mode 1 - SRF04 compatible - Separate Trigger and Echo

This mode uses separate trigger and echo pins, and is the simplest mode to use. All code examples for the SRF04 will work for the SRF05 in this mode. To use this mode, just leave the mode pin unconnected - the SRF05 has an internal pull up resistor on this pin.

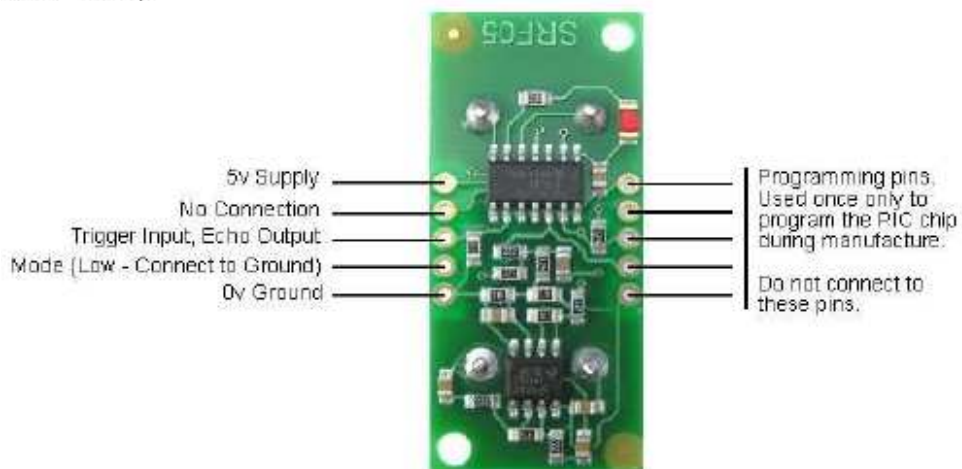


Connections for 2-pin Trigger/Echo Mode (SRF04 compatible)

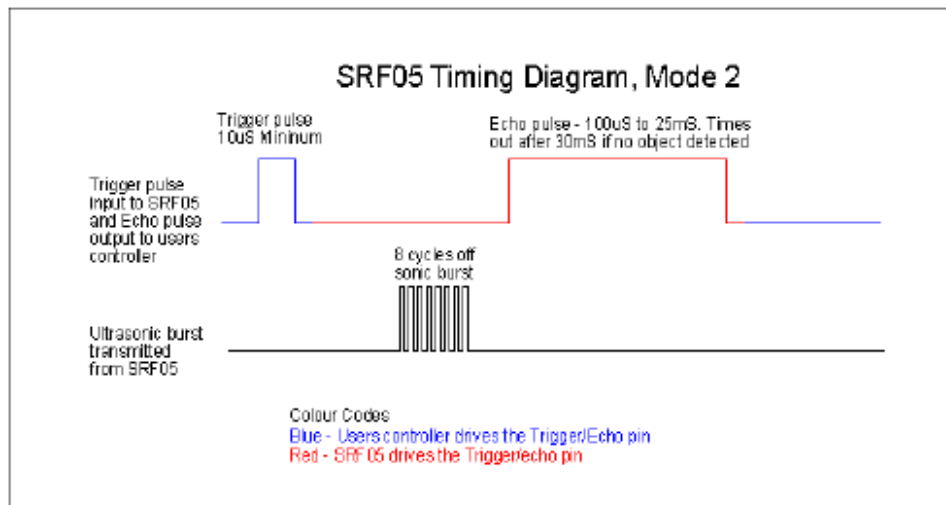


Mode 2 - Single pin for both Trigger and Echo

This mode uses a single pin for both Trigger and Echo signals, and is designed to save valuable pins on embedded controllers. To use this mode, connect the mode pin to the 0v Ground pin. The echo signal will appear on the same pin as the trigger signal. The SRF05 will not raise the echo line until 700µs after the end of the trigger signal. You have that long to turn the trigger pin around and make it an input and to have your pulse measuring code ready. The PULSIN command found on many popular controllers does this automatically.



Connections for single pin Trigger/Echo Mode



To use mode 2 with the Basic Stamp BS2, you simply use PULSOUT and PULSIN on the same pin, like this:

```
SRF05 PIN 15          ' use any pin for both trigger and echo
Range VAR Word       ' define the 16 bit range variable

SRF05 = 0            ' start with pin low
PULSOUT SRF05, 5     ' issue 10uS trigger pulse (5 x 2uS)
PULSIN SRF05, 1, Range ' measure echo time
Range = Range/29     ' convert to cm (divide by 74 for inches)
```

Calculating the Distance

The SRF05 Timing diagrams are shown above for each mode. You only need to supply a short 10µs pulse to the trigger input to start the ranging. The SRF05 will send out an 8 cycle burst of ultrasound at 40kHz and raise its echo line high (or trigger line in mode 2). It then listens for an echo, and as soon as it detects one it lowers the echo line again. The echo line is therefore a pulse whose width is proportional to the distance to the object. By timing the pulse it is possible to calculate the range in inches/centimeters or anything else. If nothing is detected then the SRF05 will lower its echo line anyway after about 30ms.

The SRF04 provides an echo pulse proportional to distance. If the width of the pulse is measured in µs, then dividing by 58 will give you the distance in cm, or dividing by 148 will give the distance in inches. $\mu\text{s}/58=\text{cm}$ or $\mu\text{s}/148=\text{inches}$.

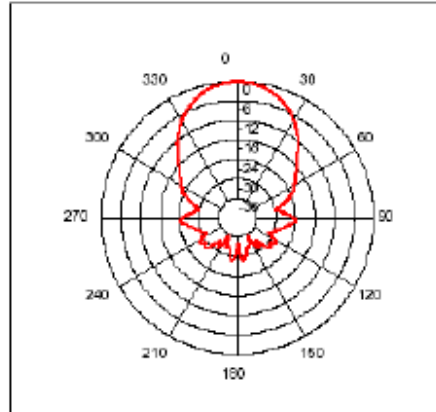
The SRF05 can be triggered as fast as every 50ms, or 20 times each second. You should wait 50ms before the next trigger, even if the SRF05 detects a close object and the echo pulse is shorter. This is to ensure the ultrasonic "beep" has faded away and will not cause a false echo on the next ranging.

The other set of 5 pins

The 5 pins marked "programming pins" are used once only during manufacture to program the Flash memory on the PIC16F630 chip. The PIC16F630's programming pins are also used for other functions on the SRF05, so make sure you don't connect anything to these pins, or you will disrupt the module's operation.

Changing beam pattern and beam width

You can't! This is a question which crops up regularly, however there is no easy way to reduce or change the beam width that I'm aware of. The beam pattern of the SRF05 is conical with the width of the beam being a function of the surface area of the transducers and is fixed. The beam pattern of the transducers used on the SRF05, taken from the manufacturers data sheet, is shown below.



**ANEXO 4:
MANUAL DE USUARIO.**

MANUAL DE USUARIO

CARACTERÍSTICAS DE LA TARJETA PRINCIPAL.

- Consumo aproximado de corriente 100 mA.
- Voltaje de operación 5 voltios.

CARACTERÍSTICAS ELECTRICAS NOMINALES DE LA BATERIA.

- Voltaje 6 V.
- Capacidad 4 Ah.

CARACTERISTICAS DE LAS PARTES QUE CONFORMAN LA TARJETA PRINCIPAL.

1. CARACTERÍSTICAS MICROCONTROLADOR PIC 16F887A.

- Consumo de corriente 16mA.
- Voltaje de operación 5 voltios.

2. CARACTERÍSTICAS DEL SENSOR SRF 05.

- Consumo de corriente 2 mA.
- Voltaje de operación 5 voltios.

3. CARACTERÍSTICAS DEL LCD.

- Consumo de corriente 15 mA.
- Voltaje de operación 5 voltios.

4. CARACTERÍSTICAS MODULO DE VOZ WT9501M03.

- Consumo de corriente 38mA y 60 mA cuando se reproduce un mensaje de audio.
- Voltaje de operación 5 voltios.

ALIMENTACION Y BATERIAS.

- Duración de la batería hasta 40 horas

- 1.- Control de contraste para el LCD.
- 2.- Bornera para el switch de encendido y apagado del circuito.
- 3.- Bornera para la conexión con la entrada de voltaje de la batería.
- 4.- Conector de chasis de la entrada DC para la batería.
- 5.- Switch de encendido y apagado del circuito.
- 6.- Jack estéreo de audio.
- 7.- Bornera para la conexión del Jack de salida de audio.
- 8.- Conector para el módulo de voz.
- 9.- Botones de control de nivel de audio del sistema de voz.
- 10.- Bornera para la conexión de los botones de control de audio.
- 11.- Bornera para la conexión del control del módulo de adquisición.
- 12.- Puerto de conexión del módulo de adquisición y el modulo principal.
- 13.- PIC16F877A.
- 14.- LCD.
- 15.- Switch de encendido y apagado del LCD.

PARTES QUE INTEGRAN EL MODULO PRINCIPAL DEL PROYECTO.

En la figura A4.2 se tiene al módulo de adquisición con cada uno de sus partes.

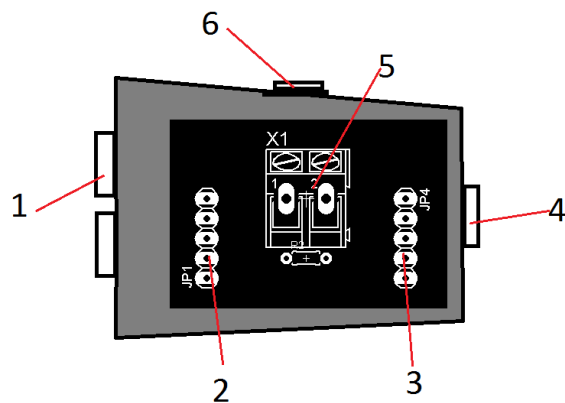


Figura A4.2 Partes del módulo de adquisición.

- 1.- Sensor ultrasónico SRF05.
- 2.- Conector para las líneas del sensor.
- 3.- Conector para la salida al módulo principal.
- 4.- Puerto de conexión del módulo de adquisición y el modulo principal.
- 5.- Bornera para conexión del pulsador activador de la lectura de distancia.
- 6.- Pulsador de activación de lectura de distancias.

MENSAJES DE VOZ.

Los mensajes de voz que se escucharán son de acuerdo a la tabla A4.1.

Número de pista	Distancia [cm]	Comentario
1	0-5	Cinco centímetros
2	6-10	Diez centímetros
3	11-15	Quince centímetros
4	16-20	Veinte centímetros

5	21-25	Veinticinco centímetros
6	26-30	Treinta centímetros
7	31-35	Treinta y cinco centímetros
8	36-40	Cuarenta centímetros
9	41-45	Cuarenta y cinco centímetros
10	46-50	Cincuenta centímetros
11	51-55	Cincuenta y cinco centímetros
12	56-60	Sesenta centímetros
13	61-65	Sesenta y cinco centímetros
14	66-70	Setenta centímetros
15	71-75	Setenta y cinco centímetros
16	76-80	Ochenta centímetros
17	81-85	Ochenta y cinco centímetros
18	86-90	Noventa centímetros
19	91-95	Noventa y cinco centímetros
20	96-100	Cien centímetros
21	101-150	Ciento cincuenta centímetros
22	151-200	Doscientos centímetros
23	201-250	Doscientos cincuenta centímetros
24	251-300	Trescientos centímetros
25	301-350	Trescientos cincuenta centímetros
26	351-400	Cuatrocientos centímetros
27	< 2 o >400	Usted puede avanzar

Tabla A4.1 Mensajes de audio.

El sistema está configurado para que se reproduzca un solo audio para un rango de distancias. Este rango varía dependiendo del tipo de detección que se va a realizar por ejemplo si se quiere saber si hay una grada o un hueco el rango es de 5cm ya que una grada tiene en alto de aproximadamente 15cm y si se usara un rango mayor a este no se pudiera saber si la grada está o no esta, y para detectar obstáculos frontales se usó un rango de 50cm ya que una persona da un paso

aproximadamente de esta distancia y si se hubiera usado un rango mayor , entonces la persona hubiera avanzado más rápido que las lecturas echas por tanto pudiera tener un accidente antes de saber que hay un obstáculo ante él.

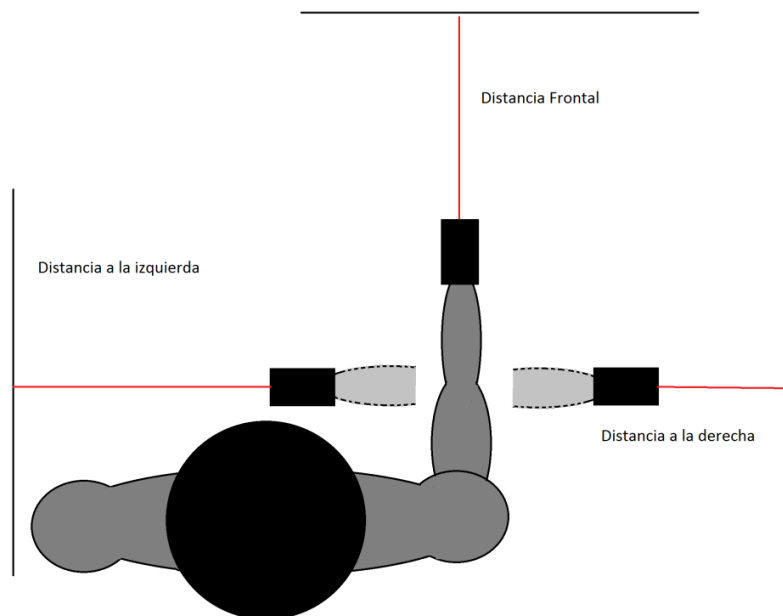
FUNCIONAMIENTO.

El sistema electrónico de alerta de obstáculos para personas no videntes con el uso de un sensor tiene 2 modos de funcionamientos. El primero que sirve para detectar obstáculos frontales, y el segundo para detectar gradas y huecos.

Detección de obstáculos frontales.

Para hacer la detección de obstáculos frontales apuntamos hacia el frente paralelamente al piso, luego pulsamos activando la detección y escuchamos el mensaje de voz que se reproduce. Si la distancia calculada está en un rango de 0 a 400cm escuchamos la distancia correspondiente. Para distancias mayores a 400cm se reproduce un mensaje que dice "USTED PUEDE AVANZAR".

El usuario debería hacer por lo menos 3 lecturas una al frente, una a la derecha y otra a la izquierda (ver figura A4.3), al saber las distancias de estas tres direcciones el usuario deberá tomar la mejor decisión para avanzar que lógicamente debería ser la dirección que tenga mayor distancia calculada o que le dé el mensaje "USTED PUEDE AVANZAR".



Se analiza que distancia es menor y se toma esa dirección

Figura A4.3 Detección de obstáculos frontales.

Detección de huecos y gradas.

Para detectar huecos o gradas la lectura se la hace con el módulo de adquisición apuntando perpendicularmente al piso a una altura aproximada de la cintura. También se puede hacer la lectura con el módulo de adquisición apuntando con una inclinación de -90° a -60° con respecto al piso en cuyo ángulo los lóbulos de radiación del sonido ultrasónico también tienen efecto, pero se debe tener cuidado de usarlo de este modo ya que se podrían producir rebotes del sonido produciendo lecturas erróneas.

Para que el usuario sepa de la presencia de un hueco o una grada tiene que hacer 2 lecturas (ver figura A4.4), la primera a una distancia de unos 5 cm de él y la otra unos 5 o 10cm más adelante, así tendrá 2 distancias que si son iguales pues no hay ningún obstáculo y se puede avanzar pero si se tiene una variación de más de 5cm es señal de que hay un obstáculo de estos. Para un hueco se tendría que la primera

lectura es menor que la segunda y si se tiene lo contrario entonces se tiene una grada adelante.

Se realiza 2 lecturas y se compara las distancias obtenidas, si hay variación se tiene un hueco o una grada dependiendo de si la segunda lectura es mayor o menor

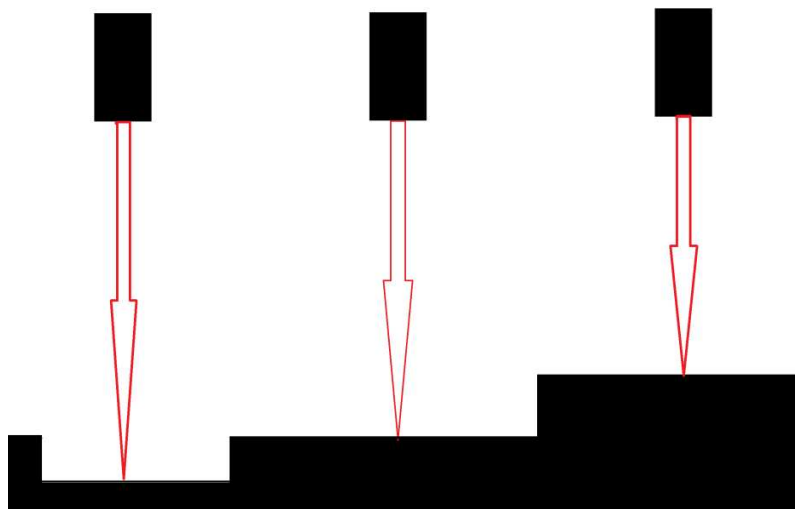


Figura A4.4 Detección de huecos y gradas.

MANTENIMIENTO PREVENTIVO Y CORRECTIVO.

Para el mantenimiento por parte del técnico encargado se debe considerar lo siguiente:

- 1.- Revisar que todos los elementos estén funcionando de forma correcta.
- 2.- Comprobar que el sensor esté funcionando, esto se puede saber observando el led que tiene este, si esta con un color verde y titila de vez en cuando quiere decir que si lo está, pero si esta de color rojo, entonces algo debe de estar mal.
- 3.- Observar los resultados que se visualizan en el LCD, este está dentro de la caja del módulo principal y tiene que estar apagado para que no consuma energía en vano ya que este LCD esta solo con fin de mantenimiento. El técnico lo que debe hacer es realizar una prueba calculando con el sensor la distancia a un obstáculo y

comparar que el audio escuchado coincida con el visualizado en el LCD y con la medida hecha con un flexómetro, si coinciden entonces todo está bien pero si no se debe hacer las correcciones pertinentes y volver a probar hasta que coincidan.