

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA



FACULTAD DE CIENCIAS DE LA COMPUTACIÓN



KIP
KIT INALÁMBRICO DE PERIFÉRICOS

Tesis que para obtener el grado de
Licenciado en Ciencias de la Computación
Presenta:

Rita Ayón Rodríguez

Asesor
M.C. José Esteban Torres León

Puebla Pue. Agosto 2015

ÍNDICE

CAPÍTULO 1 INTRODUCCIÓN	
1.1. Motivación	1
1.2. Objetivos	2
1.3. Organización de la tesis	3
CAPÍTULO 2 MICROCONTROLADORES	
2.1 Microcontrolador	4
2.1.1 Gamas de la familia de los MCUs de microchip	5
2.1.2 Empaquetamientos de los MCUs	6
2.1.3 Definición de las partes de la nomenclatura de un MCU PIC	7
2.2 Bosquejo Histórico	8
2.3 Características generales de los MicrosPICs	8
2.4 Arquitecturas utilizadas en los microcontroladores	10
2.5 Memoria	11
CAPÍTULO 3 TECNOLOGÍA BLUETOOTH	
3.1 Introducción	17
3.2 Bosquejo de la tecnología Bluetooth	18
3.3 Funcionamiento del Bluetooth	19
CAPÍTULO 4 ANÁLISIS Y DISEÑO DEL KIP	
4.1 Análisis	24
4.2 Diseño del KIP	25
4.2.1 Alimentación	25
4.2.2 Módulos de Entrada Salida (E/S)	25
4.2.3 Microcontrolador	25
4.2.4 Herramientas de programación	26
4.2.5 Componentes del diseño	27
4.2.6 Diseño	27
4.3 Descripción de cada módulo	28
4.3.1 Módulos de Salida	28
4.3.2 Módulos de Entrada	30
4.3.3 Módulos Micros	31
4.3.4 Módulos Bluetooth	31
CAPÍTULO 5 IMPLEMENTACIÓN DEL KIP	
5.1 Módulo Leds	32
5.2 Módulo Display con decodificador	37
5.3 Módulo 4 Displays	39
5.4 Módulo LCD	41
5.5 Módulo Teclado Lineal	45
5.6 Módulo Teclado 4x4	46
5.7 Módulo Teclado LDR	48
5.8 Módulo Bluetooth	51
5.9 Módulo Micro	56

CAPÍTULO 6 APLICACIONES	
6.1 Uso de los módulos	59
6.2 Aplicaciones con sistemas digitales (SD)	60
6.3 Aplicaciones con microcontroladores (MCU)	80
6.4 Aplicaciones inalámbricas	94
CONCLUSIONES Y PERSPECTIVAS	
Conclusiones	99
Perspectivas	99
BIBLIOGRAFÍA	100

CAPÍTULO 1

INTRODUCCIÓN

En este capítulo se describe la razón que ha llevado al desarrollo de este proyecto, los objetivos, el alcance y organización del mismo.

1.1. Motivación.

El presente proyecto es parte de un proyecto mayor que quería automatizar todo un laboratorio, Redes de Computadoras, con tecnología alamburada e inalámbrica. La idea del gran proyecto era soportar aplicaciones para el control y monitoreo de aplicaciones digitales y de microprocesadores por internet o por una red local, pero lo principal era manejar entrada y salida básica para luego conectarlo a las redes de computadoras.

Este proyecto, denominado originalmente Kit entrenador de periféricos (KEP), fue planteado a finales de los 90's, con la idea de tener módulos periféricos de entrada y salida, módulos independientes, para usos académicos y que se pudieran utilizar en varias materias. Esto permitiría que profesores y alumnos los utilizaran para realizar prácticas del área de Hardware y la nueva área de Redes de Computadoras, que es donde se quería dar un fuerte impulso a algo que hoy se conoce como internet de las cosas.

Como estudiante la realización de prácticas en las áreas señaladas, hace ya algunos años, eran alamburadas totalmente, sin embargo, al regresar a la escuela, ver y revisar cómo en el laboratorio se realizan hoy las prácticas, sorpresa, se siguen manejando igual. Es decir, se arman las prácticas de Hardware y Redes de computadoras de manera alamburada.

Antes se utilizaban protoboards, material para armar las prácticas y una fuente de poder básica. Hoy el protoboard está incluido en una fuente o módulo de experimentos, que presenta un arreglo de switches y leds para de entrada y salida. Y tal vez se conecten a una computadora, y se puedan manejar por software, pero no se sabe cómo están armados y porque funcionan así y, cuando se dañan, sorpresa, no saben cómo repararlas, lo que antes si se hacía en el laboratorio.

Parece ser que la tecnología abandonó al área de Hardware ya que hoy todo se maneja de forma modular y con conexión inalámbrica. La modularidad permite resolver problemas de hardware de manera rápida y no se debe perder tiempo en armar los periféricos de entrada y salida, sólo se deben conectar. Y lo inalámbrico y ante tantas tecnologías se pueden comunicar dispositivos diversos para formar redes de dispositivos, redes de sensores, Redes had-hoc, o se utilizan en cómputo móvil o cómputo ubicuo. También se pueden armar tarjetas inteligentes para la industria, la Domótica, etc.

Ante la inquietud de que los alumnos y maestros dejen de alambrear los periféricos de entrada-salida (E/S) para sus prácticas de Hardware y Redes de Computadoras y empiecen a utilizar módulos de periféricos E/S, y además empiecen a utilizar alguna tecnología inalámbrica, se propone desarrollar módulos periféricos de E/S para atender la parte experimental de las materias del área de hardware como Diseño Digital y Microcontroladores así como incluir el desarrollo de módulos inalámbricos (maestro y esclavo) para comunicar dos periféricos.

Las materias de Digitales y Microcontroladores se han elegido, dado su peso en la industria, en las oficinas, empresas, etc. Y su uso está en áreas tan diversas como aplicaciones embebidas, internet de la cosas, periféricos alternos E/S por mencionar algunas. Y en estas áreas, todos los periféricos E/S que se manejan son módulos sencillos o módulos inteligentes.

Por otro lado, en el mercado ya existen tecnologías inalámbricas como IrDA, pero aspectos como su ancho de banda o su operatividad (línea visual entre los dispositivos que se intercomunican) no han permitido el asentamiento esperado en el mercado. Por ello, se propone utilizar un sistema de radio, de bajo coste que permita la conectividad entre dispositivos, sin utilizar cables y sin la necesidad de que haya una línea visual entre emisor y receptor. Actualmente el Bluetooth es una tecnología en auge que ofrece muchas ventajas respecto al resto de tecnologías y en la que parece que todas las empresas del sector de las telecomunicaciones y la informática están muy interesadas.

1.2 Objetivos

De lo anterior se desprenden los siguientes objetivos:

Objetivo General

El objetivo general de este proyecto es el diseño e implementación de Módulos periféricos de E/S para soportar prácticas de Digitales y Microcontroladores. Así como dejar las bases para comunicar de manera inalámbrica un módulo maestro y un módulo esclavo basado en microcontroladores.

Aparte de este objetivo, se deben cubrir los objetivos específicos:

- Investigar y desarrollar prácticas del área de Digitales y Microcontroladores que incluya periféricos de entrada salida.
- Investigar y probar la tecnología Bluetooth que permita a nivel de microcontroladores comunicar un módulo maestro con un módulo esclavo.
- Investigar y aplicar las herramientas como MPLAB, PIC C y Proteus para programar y simular las prácticas.
- Investigar y aplicar la herramienta PCB Wizard para realizar los circuitos impresos.
- Desarrollar prácticas tipos
- Documentar el trabajo

1.3 Organización de la tesis.

Este trabajo se compone de varios capítulos, todos importantes para el trabajo. Inicialmente, en este primer capítulo se está presentando al proyecto, atendiendo la motivación, objetivos que se persiguen y finalmente como está organizada la documentación del mismo.

En el segundo capítulo se describe a los microcontroladores PIC, sus principales características, en especial los microcontroladores PIC 16F877 y 18F4550.

En la tercera unidad se describen las tecnologías inalámbricas, en particular a la tecnología Bluetooth, sus principales características y funcionamiento.

El análisis de requerimientos y diseño se lleva a cabo en el capítulo cuarto.

La implementación del sistema KIP se describe en la unidad cinco.

En la sexta unidad se describen las pruebas realizadas y resultados obtenidos.

Finalmente se presentan las conclusiones y perspectivas, así como la bibliografía.

CAPÍTULO 2

MICROCONTROLADORES PIC

En este capítulo se describe un panorama general de los microcontroladores PIC de Microchip.

2.1 MICROCONTROLADOR

Los avances en la microelectrónica han permitido que en un solo chip (dado) de silicio, que originalmente contenían compuertas, hoy en día contengan computadoras completas, arreglos vectoriales y matriciales de procesadores, entre otros. En particular los microcontroladores están aplicados en muchos equipos electrónicos como hornos de microondas, televisores, lavadoras, control de luces, etc.

Un microcontrolador es una computadora encapsulada en un chip, capaz de ejecutar un programa. El microcontrolador integra en un solo circuito integrado al microprocesador, memoria de programa, memoria de datos, puertos de entrada y puertos de salida, y algunas características especiales como puertos serie, comparadores, convertidores de analógico-digital, protección de código, controlador de interrupciones, etc.

El mercado está inundado de microcontroladores de diferentes empresas como Microchip, Motorola, Intel y otras. Dependiendo de la aplicación, prestaciones, costos y otros parámetros que se requieran para una aplicación, es como se elige al microcontrolador.

El microcontrolador ejecuta instrucciones, con las herramientas adecuadas y una guía de programación se genera el programa. Las instrucciones son leídas de la memoria de programa para ejecutarlas una detrás de otra. La memoria de programa contiene las instrucciones que se quieren que el microcontrolador ejecute.

Programar un microcontrolador consiste en introducir un programa con extensión hex (hexadecimal) en la memoria del microcontrolador. Las instrucciones son operaciones simples como sumar, restar, escribir en un puerto, activar un bit de un dato, etc. Con las instrucciones básicas se pueden realizar operaciones más complejas y así llegar al objetivo de la aplicación.

En esta tesis se trabajará con microcontroladores de Microchip Technology o microcontroladores PIC, debido a que están muy extendidos en el mercado gracias a su gran variedad y bajo coste. Otra razón del éxito de los PICs es su utilización, ya que una vez se ha aprendido a utilizar uno, conociendo su arquitectura y juego de instrucciones, es muy fácil emplear otro modelo diferente.

El nombre de PIC no es un acrónimo, en realidad el nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller o controlador de interfaz periférico.

De las diferentes familias de microcontroladores (MCU), se elige a la familia de Microchip por las siguientes razones:

- Sencillez de manejo ya que su arquitectura RISC tiene un juego reducido de instrucciones de 35 en la gama media.
- Buena información
- Precio bajo comparativamente inferior al de otras familias.
- Buen promedio de parámetros: velocidad, consumo, tamaño, alimentación, código compacto, etc.
- Herramientas de desarrollo fáciles y baratas. El software se puede obtener libremente a través de internet desde Microchip y herramientas hardware que permiten grabar, depurar, borrar y comprobar el comportamiento de los PIC.
- Diseño rápido.
- Variedad de modelos, permitiendo elegir el que mejor responda a los requerimientos de la aplicación.

2.1.1 Gamas de la familia de los MCUs de microchip

Los MCUs de Microchip se dividen en gamas.

Gama enana. Son MCUs PIC12CXXX y PIC12FXXX de 8 pines (terminales), conjunto reducido de instrucciones de 12 ó 14 bits de ancho, voltaje de operación bajo de 2.5V, manejo de interrupciones, una pila de varios niveles, múltiples canales A/D, memoria de programa FLASH, OTP o ROM y una memoria de datos EEPROM. Todas estas características proporcionan un nivel de inteligencia no antes disponible tomando en cuenta el costo y tamaño.

Gama Básica. Son MCUs PIC16C5X con instrucciones de 12 bits de longitud, disponible en circuitos de 14, 18, 20 y 28 pines, operan con bajo voltaje a menos de 2.0V para OTP MCUs lo que hace a esta familia ideal para las aplicaciones que requieren de batería. Además, el PIC16HV5XX puede operar con más de 15V para el uso directo con una batería.

Gama media. Microchip ofrece en esta gama a los MCUs PIC16CXXX y PIC16FXXX con instrucciones de 14 bits de longitud. Proporciona un ADC (Convertidor Analógico/Digital) de 12 bits. Estos MCUs ofrecen un amplio rango de opciones en número de terminales de 18 a 68, van del nivel bajo al más alto en cuanto a la integración de periféricos, manejan interrupciones y una pila de 8 niveles. Esta gama proporciona la versatilidad para cubrir los requerimientos de más demanda hoy en día.

Gama alta. Son MCUs PIC17CXXX con instrucciones de 16 bits de ancho, mejora el set de instrucciones y la capacidad de manejo de interrupciones. Cuenta con las características de periféricos de mayor demanda en las aplicaciones.

Gama superior. Familia de MCUs de alto nivel con PIC18CXXX y PIC18FXXX, instrucciones de 16 bits de longitud, de tecnología CMOS, tiene un convertidor A/D, una pila de 32 niveles, múltiples interrupciones internas y externas, un set de 77 instrucciones. Además tiene un set de largos registros que permiten algunas de las innovaciones usadas para alcanzar un alto rendimiento de 10 MIPS (Millones de Instrucciones por Segundo). Esta gama tiene características especiales para reducir Componentes externos, por ello se reduce el costo, aumenta la confiabilidad y reduce el consumo de energía.

2.1.2 Empaquetamientos de los MCUs

Los tipos de empaquetamiento que hay para los MCUs de microchip se muestran en las figuras 2.1 a, b, c, y d.



Figura 2.1.a Empaquetamiento de plástico PDIP (Plastic Dual in line Package) On Time Programmable (OTP), Pin Through Hole (PTH)

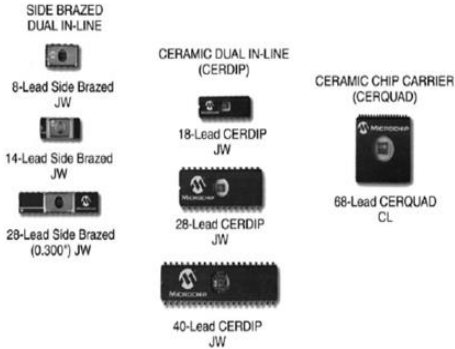


Figura 2.1.b Empaquetamiento de ceramic (Ceramic DIP- CERDIP), Quaduple In-Line Package – QIP

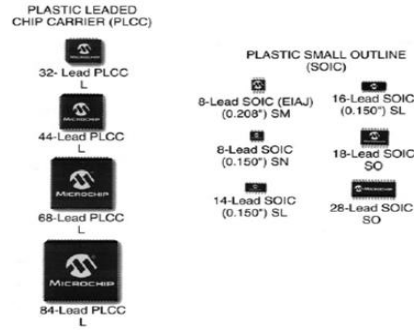


Figura 2.1.c Empaquetamiento de montaje superficial estandar.

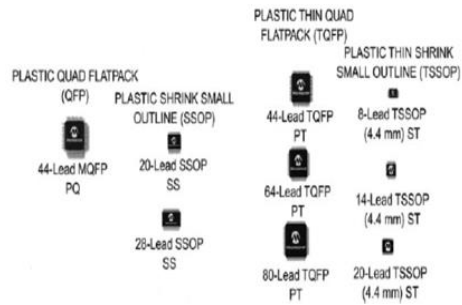


Figura 2.1.d Empaquetamiento de montaje superficial de plástico fino.

Figura 2.1 Diferentes tipos de empaquetamiento para MCUs PICs.

2.1.3 Definición de las partes de la nomenclatura de un MCU PIC

La nomenclatura de un MCU PIC tiene varias partes como el número del MCU PIC, rangos de velocidad y temperatura, tipo de empaquetamiento entre otros. En la figura 2.2 se muestra la definición de las partes de la nomenclatura de los MCUs PIC.

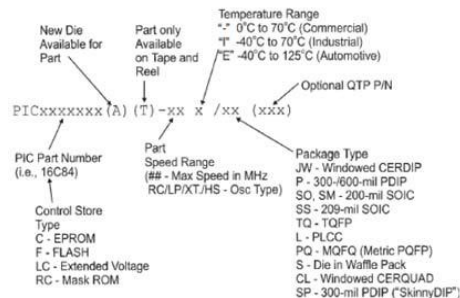


Figura 2.2 Definición de las partes del número de los MCUs PIC

2.2 BOSQUEJO HISTÓRICO

En 1965 General Instrument (GI) formó una división de microelectrónica destinada a generar las primeras arquitecturas viables de memoria EPROM y EEPROM.

En 1970, en sus inicios, se creó el CP16000 que era un microprocesador bastante bueno pero no manejaba muy bien los puertos de E/S.

En el año 1971 apareció el primer microprocesador 4004 de Intel, el cual originó un cambio decisivo en las técnicas de diseño de los equipos. A partir de ese momento, se construyó equipo electrónico con un microprocesador, unos pocos circuitos accesorios y un software. Los diseños electrónicos se hicieron más pequeños y simplificados. Los diseñadores de equipos electrónicos realizaban mayor cantidad de tareas en menos tiempo y el tamaño de los equipos se redujo considerablemente.

En 1975 se desarrolló el PIC de 8 bits para mejorar el rendimiento del sistema quitando peso de E/S a la CPU.

Para 1980, GI reestructuró la empresa y creó la GI Microelectronics.

En 1985 la división de microelectrónica de General Instrument se separa como compañía independiente que es incorporada como filial y se creó la actual Arizona Microchip Technology y centraron su negocio en la fabricación de PICs, memorias EEPROM y EPROM. 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.

Actualmente Microchip ha realizado un gran número de mejoras a la arquitectura original de los PICs y ha entrado a trabajar con DSPICs (Microcontroladores para el Procesamiento de señales) y ha definido familias adaptándola a las actuales tecnologías y al bajo costo de los semiconductores.

2.3 CARACTERÍSTICAS GENERALES DE LOS MicroPICs

Las principales características de los PICs son las siguientes.

- La arquitectura del microcontrolador PIC es una arquitectura Harvard, es decir, el microprocesador se conecta por buses separados a la memoria de instrucciones y a la memoria de datos, esto permite al CPU acceder simultáneamente a las dos memorias. Esto proporciona mayor velocidad además de numerosas ventajas al funcionamiento del sistema, ver figura 2.3.

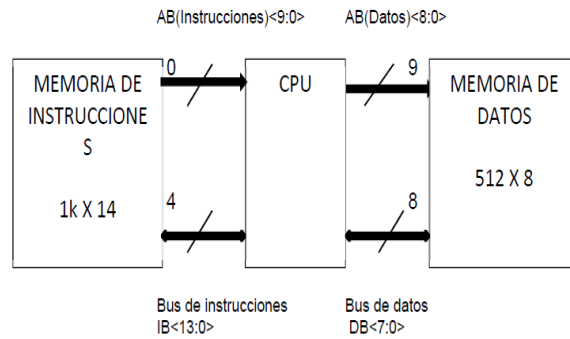


Figura 2.3 Arquitectura Harvard y memorias de datos e instrucciones

- Se aplica la técnica de segmentación ("pipeline") en la ejecución de las instrucciones. La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj). La segmentación permite al procesador ejecutar cada instrucción en un ciclo de instrucción equivalente a cuatro ciclos de reloj. En cada ciclo se realiza la búsqueda de una instrucción y la ejecución de la anterior, ver figura 2.4. Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

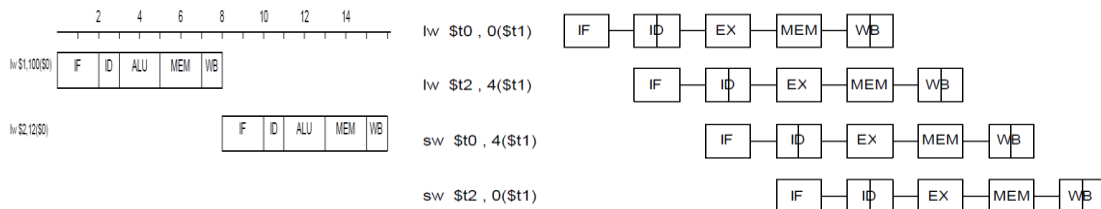


Figura 2.4 Ejecución sin segmentación y con segmentación

- El formato de todas las instrucciones tiene la misma longitud. Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.
- Procesador RISC (Computador de Juego de Instrucciones Reducido).

Dependiendo de la gama del procesador (baja, media o alta) tienen más o menos número de instrucciones. Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y unas 76 los de la alta.

- Todas las instrucciones son ortogonales. Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.
- Arquitectura basada en un banco de registros. Esto significa que todos los objetos del sistema (puertos de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.
- Diversidad de modelos de microcontroladores con prestaciones y recursos diferentes.
- La gran variedad de modelos de microcontroladores PIC permiten que el usuario pueda seleccionar el más conveniente para su aplicación.
- Herramientas de soporte potentes y económicas.
En nuestro caso se utilizarán las herramientas de software que facilita Microchip. Dichas herramientas se pueden descargar de la página Web de microchip (www.microchip.com), son gratuitos.

2.4 ARQUITECTURAS UTILIZADAS EN LOS MICROCONTROLADORES

Hay dos arquitecturas que se utilizan para fabricar microcontroladores:

1. Microcontroladores de arquitectura cerrada.
Este tipo de arquitectura en el microcontrolador impide la modificación de recursos específicos, es decir, no admiten ningún tipo de variaciones ni de ampliaciones.
Esto obliga a que una aplicación sólo utilice lo que proporciona el microcontrolador.
Microchip ha elegido este modelo de arquitectura para sus aplicaciones.
2. Microcontroladores de arquitectura abierta
La arquitectura de este tipo de microcontroladores, aparte de tener una estructura interna determinada, permite ampliar el uso de sus líneas de E/S al brindar al exterior los buses de datos, direcciones y control, con lo que se posibilita la ampliación de la memoria y las E/S con circuitos integrados externos.
Microchip dispone de modelos PIC con arquitectura abierta, sin embargo, esta alternativa se escapa de la idea de un microcontrolador incrustado y se asemeja a la solución que emplean los clásicos microprocesadores.

2.5 MEMORIA

Los PICs debido a su arquitectura Harvard poseen dos bloques de memoria distintos: una para la memoria de programa y otra para la memoria de datos.

Las dos memorias son independientes entre ellas, cada una tiene su tamaño y longitudes de palabra distintas.

Cada bloque posee su propio bus, lo que permite que el acceso a cada bloque puede producirse durante el mismo ciclo del reloj.

Memoria de datos

La memoria de datos se divide en memoria RAM para propósitos generales y los Registros de Funciones Especiales (SFR).

La memoria de datos también contiene los datos de la memoria EEPROM. Esta memoria no está directamente introducida en la memoria de datos, si no que se accesa de forma indirecta. Esto significa que un puntero indirecto de direcciones especifica la dirección de la memoria de datos EEPROM para escribir y leer.

La memoria de datos se organiza en bancos o páginas, la cantidad de los bancos depende de la gama de PIC que se utilice. Para poder acceder a un banco se hace por medio del registro FSR.

La memoria de datos funciona de forma similar al banco de registros de un procesador, por lo cual, sus posiciones implementan registros de propósito especial y propósito general. Las primeras posiciones de la memoria se destinan a registros específicos.

Memoria de programa.

La memoria de programa o memoria de instrucciones, es donde se aloja al programa de aplicación.

Existen diferentes tipos de memorias de programa, dependiendo de las necesidades, en nuestro caso se trabajará con memoria tipo flash. Este tipo de memoria se puede programar y borrar eléctricamente alrededor de 1000 veces, lo cual, la hace muy útil para el aprendizaje ya que con una pequeña inversión en un chip lo podemos programar tantas veces como se quiera.

La cantidad de memoria también depende de la gama de PIC que se esté utilizando ya que en gamas bajas y programas simples se tienen PICs con menos memoria y menos costos y para proyectos de mayor necesidad se tienen PICs de más memoria.

Registros.

Los PICs utilizan una arquitectura basada en registros. Esto significa que todos los objetos del sistema (puertos de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.

Existen diferentes tipos de registros:

Los Registros de propósito general utilizados para Puertos E/S, Contadores, etc...

Los registros especiales, que son registros de funcionamiento y configuración.

Todos los registros están ubicados en una posición específica de la memoria.

Contador de programa.

Este registro denominado PC (program counter), es equivalente al de todos los microprocesadores y contiene la dirección de la próxima instrucción a ejecutar. Se incrementa automáticamente al ejecutar cada instrucción, de manera que la secuencia natural de ejecución del programa es lineal, una instrucción después de la otra.

Algunas instrucciones de control, cambian el contenido del PC alterando la secuencia lineal de ejecución. Dentro de estas instrucciones se encuentran el GOTO y el CALL, que permiten cargar en forma directa un valor constante en el PC haciendo que el programa salte a cualquier posición de la memoria. Otras instrucciones de control son los SKIP o “saltos” condicionales, que producen un incremento adicional del PC si se cumple una condición específica, haciendo que el programa salte, sin ejecutar, la instrucción siguiente.

Al resetearse el microprocesador, todos los bits del PC toman valor 1, de manera que la dirección de arranque del programa es siempre la última posición de memoria de programa. En esta posición se deberá poner una instrucción de salto al punto donde verdaderamente se inicia el programa.

A diferencia de la mayoría de los microprocesadores convencionales, el PC es también accesible al programador como registro de memoria interna de datos, en la posición de 02h. Es decir que cualquier instrucción común que opere sobre registros puede ser utilizada para alterar el PC y desviar la ejecución del programa.

El uso indiscriminado de este tipo de instrucciones complica el programa y puede ser muy peligroso, ya que puede producir comportamientos difíciles de predecir. Sin embargo, algunas de estas instrucciones utilizadas con cierto método, pueden ser muy útiles para implementar poderosas estructuras.

Stack.

En los microcontroladores PIC, el stack o pila, es una memoria interna dedicada, de

tamaño limitado, separada de las memorias de datos y de programa, inaccesible al programador, y organizada en forma de pila, que es utilizada sólo, y en forma automática, para guardar las direcciones de retorno de subrutinas e interrupciones.

Permite guardar una copia completa del PC. Como en toda memoria tipo pila, estructura LIFO (Last Input First Output), los datos son accedidos de manera tal que el primero que entra es el último que sale.

Las posiciones de la pila en un PIC está limitada, lo que no permite hacer uso intensivo del anidamiento de subrutinas. Sólo se pueden anidar dos niveles de subrutinas, es decir que una subrutina que es llamada desde el programa principal, puede a su vez llamar a otra subrutina, pero esta última no puede llamar a una tercera, porque se desborda la capacidad del stack, que solo puede almacenar dos direcciones de retorno, esto dependiendo de la gama del PIC.

Aunque esto a priori sea un problema, es una buena solución de compromiso ya que estos microcontroladores están diseñados para aplicaciones de alta velocidad en tiempo real, en las que el overhead (demoras adicionales) que ocasiona un excesivo anidamiento de subrutinas es inaceptable.

El stack y el puntero interno que lo direcciona, son invisibles para el programador, solo se les accede automáticamente para guardar o rescatar las direcciones de programa cuando se ejecutan las instrucciones de llamada o retorno de subrutinas, o cuando se produce una interrupción o se ejecuta una instrucción de retorno de ella.

Puertos de Entrada / Salida.

Un recurso imprescindible para los microcontroladores son los puertos de Entradas y Salidas con los cuales se comunica con los periféricos del mundo exterior.

Dependiendo de la gama de PIC que se esté utilizando, estos tendrán más o menos puestos de E/S y también dependiendo de la gama utilizada pueden ser digitales, analógicas, multiplexadas, etc. También algunos PICs permiten utilizarlas como comparadores y conversores, pero esas características ya dependen de cada PIC en particular.

Los puertos de entrada y salida utilizan la denominación de Puerto A (RA), Puerto B (RB), etc, dependiendo de la cantidad de puertos que tengan.

Temporizador / Contador.

Una exigencia en las aplicaciones de control es la regulación estricta de los tiempos que duran las diversas acciones que realiza el sistema. El dispositivo típico destinado a gobernar los tiempos recibe el nombre de temporizador o timer y, básicamente, consiste en un contador ascendente o descendente que determina un tiempo determinado entre el valor que se le carga y el momento en que se produce su desbordamiento o paso por 0.

La cantidad de temporizadores depende de la gama de PIC que se esté utilizando, en la gama baja, los microcontroladores PIC sólo disponen de dos temporizadores. Uno de ellos actúa como Principal y sobre él recae el control de tiempos de las operaciones del sistema. El otro recibe el nombre de Perro guardián o "Watchdog". Las gamas más altas de microcontroladores PIC disponen de más temporizadores.

El Perro guardián vigila que el programa no se "cuelgue" y dejen de ejecutarse las instrucciones secuenciales del mismo tal como lo ha previsto el diseñador. Para realizar esta labor de vigilancia, el Perro guardián da un paseo por la CPU cada cierto tiempo y comprueba si el programa se ejecuta normalmente; en caso contrario, por ejemplo si el control está detenido en un bucle infinito o a la espera de algún acontecimiento que no se produce, el perro ladra y provoca un reset, reiniciando todo el sistema.

Tanto el Temporizador principal TMR0 como el Perro guardián WDT, a veces precisan controlar tiempos largos y aumentar la duración de los impulsos de reloj que les incrementan o decrementan. Para cubrir esta necesidad, se dispone de un circuito programable llamado Divisor de frecuencia que divide la frecuencia utilizada por diversos rangos para poder realizar temporizaciones más largas.

Interrupciones.

Una interrupción consiste en una detención del programa en curso para realizar una determinada rutina que atienda la causa que ha provocado la interrupción. Es como una llamada a subrutina, que se origina por otra causa, más no por una instrucción del tipo CALL. Tras la terminación de la rutina de interrupción, se retorna al programa principal en el punto en que se abandonó.

Las causas que originan una interrupción pueden ser externas, como la activación de una terminal con el nivel lógico apropiado, e internas, como las que pueden producirse al desbordarse un temporizador como el TMR0.

En las aplicaciones industriales, las interrupciones son un producto muy potente para atender los acontecimientos físicos en tiempo real.

Existen diferentes tipos de interrupciones que dependen del tipo de microcontrolador que se utiliza, y para saber de ellas, se puede consultar el data sheet del PIC utilizado.

Instrucciones.

Dependiendo de la gama del microcontrolador que se esté utilizando (baja, media o alta,...) se tienen más o menos número de instrucciones.

Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y unas 76 los de la gama alta.

Modos de direccionamiento.

Cuando se programa un PIC, a la hora de especificar los datos y operandos, se puede hacer de 3 formas distintas que se llaman modos de direccionamiento. Estos modos de direccionamiento son: el Inmediato, el Directo y el Indirecto.

El modo de direccionamiento inmediato es cuando se utiliza el valor de una literal. El modo de direccionamiento directo es cuando se utiliza un valor que apunta a una determinada posición de memoria y el modo de direccionamiento indirecto es

cuando se utiliza como operando el registro INDF, el cual, accede a la posición que apunta el contenido del registro FSR ubicado en el área de datos.

Oscilador externo.

Un microcontrolador utiliza un circuito externo que le indica la velocidad a la que debe trabajar. Este circuito es el oscilador o reloj, es muy simple pero de vital importancia para el buen funcionamiento del sistema.

Los PIC pueden utilizar cuatro tipos de osciladores diferentes: RC (Oscilador con resistencia y condensador), XT (Cristal de cuarzo), HS (Cristal de alta velocidad) y LP (Cristal para baja frecuencia y bajo consumo de potencia).

En el momento de programar o "quemar" el microcontrolador se debe especificar el tipo de oscilador que se usa. Esto se hace a través de unos fusibles llamados "fusibles de configuración".

Generalmente se suele utilizar un cristal de 4Mhz y de esta manera, internamente esta frecuencia, queda dividida por cuatro, lo que hace que la frecuencia efectiva de trabajo sea de 1 MHz, por lo que cada instrucción se realiza en un microsegundo (1 μ s). El cristal debe ir acompañado de dos condensadores y se conecta como se muestra en la figura 2.5.

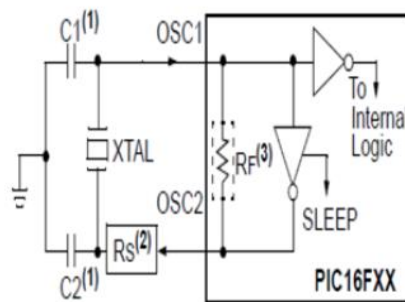


Figura 2.5 Oscilador de cristal.

Herramientas de desarrollo.

Microchip ofrece herramientas de desarrollo totalmente gratuitas, las cuales se pueden descargar de la página de www.microchip.com. Con estas herramientas se pueden programar los microcontroladores PIC.

La herramienta que ofrece Microchip Technology es "MPLAB", MPLAB es un entorno de desarrollo integrado (Integrated Development Environment, IDE) que corre en entornos "Windows", mediante el cual se pueden desarrollar aplicaciones para los microcontroladores PIC.

MPLAB permite escribir, depurar y optimizar los programas (firmware) de nuestros diseños con PICs. MPLAB incluye un editor de texto, un simulador y un

organizador de proyectos. Además, el MPLAB soporta el emulador PICMASTER y a otras herramientas de desarrollo de Microchip como el PICSTART-Plus.

CAPÍTULO 3

TECNOLOGÍA BLUETOOTH

Este capítulo describe la tecnología de comunicación inalámbrica Bluetooth, considerando una introducción, funcionamiento y conceptos.

3.1. INTRODUCCIÓN

Bluetooth es una tecnología orientada a la conectividad inalámbrica entre dispositivos tan dispares como PCs (Computadoras Personales), PDAs (Asistentes Digitales Personalizados), teléfonos móviles, electrodomésticos, entre otros.

Bluetooth, aparte de ser una tecnología es también una especificación abierta para comunicaciones inalámbricas de voz y datos. Se basa en un enlace de radio de bajo coste y corto alcance, el cual proporciona conexiones instantáneas (ad hoc) tanto para entornos de comunicaciones móviles como estáticos.

Esta tecnología ha revolucionado el mercado de la conectividad ya que es capaz de comunicar cualquier dispositivo que cumpla con las especificaciones inalámbricas del Bluetooth, ver figura 3.1.

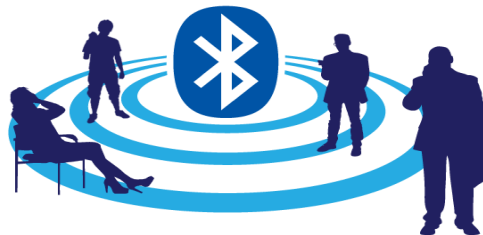


Figura 3.1 Bluetooth

La principal ventaja que ofrece esta tecnología es la conectividad sin cables de todos los dispositivos, pero más que reemplazar los incómodos cables, esta tecnología ofrece un puente entre las redes de datos hoy existentes y el exterior.

Bluetooth, al ser un estándar abierto, pretende conectar una amplia gama de dispositivos sin importar su marca.

Las principales características de Bluetooth son su Robustez, Bajo coste, Consumo de poca potencia, Baja complejidad y es un estándar global.

La tecnología Bluetooth comprende hardware, software y requerimientos de interoperabilidad, por lo que, para su desarrollo ha sido necesaria la participación

de los principales fabricantes de los sectores de las telecomunicaciones y la informática.

En la figura 3.2 se puede observar la imagen global de lo que pretende esta especificación.

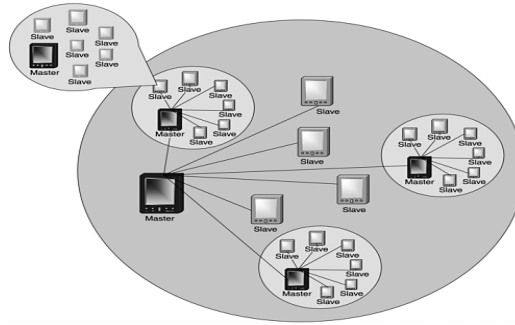


Figura 3.2 Comunicación Bluetooth

3.2. BOSQUEJO DE LA TECNOLOGÍA BLUETOOTH.

El nombre Bluetooth es en honor al rey Danés Harald Blaaland del siglo X. Se dice que tenía grandes habilidades comunicativas que lo hicieron famoso e inició el proceso de cristianización de la sociedad vikinga y logro unificar Dinamarca que estaba dividida.

La historia de Bluetooth es relativamente corta. Las primeras investigaciones fueron realizadas por Ericsson Mobile Communications en 1994. Esta compañía promovió una iniciativa para estudiar la viabilidad de una interfaz por radio entre los teléfonos móviles y sus accesorios, con la característica y condición de que tuviese un bajo coste y pequeño consumo. Hasta entonces, las tecnologías de comunicaciones basadas en el cable funcionaban con eficiencia, sin embargo, su instalación y configuración resultaba bastante dificultosa.

Conforme avanzaba el proyecto, Ericsson fue despertando el interés de otros fabricantes y enseguida se vio el gran abanico de posibilidades que ofrecía esta tecnología. A principios de 1998 se creó la SIG (grupo de interés especial), el cual, estaba formado por Ericsson, Nokia, IBM, Toshiba e Intel.

Actualmente se han integrado otras empresas al grupo como 3Com, Corporation, Lucent Technologies, Microsoft Corporation, Motorola Inc y otras más, llegando hoy a estar formado por más de 2000 empresas del sector informático y de las telecomunicaciones.

El propósito principal del SIG es establecer un standard para la interfaz aérea junto con su software de control, su fin es asegurar la interoperabilidad de los diversos equipos de diferentes fabricantes.

3.3. FUNCIONAMIENTO DE BLUETOOTH

Bluetooth funciona en la banda ISM (médico-científica-industria), con rangos que van de los 2,4 a los 2,5 GHz, excepto en algunos países como Francia, España y Japón en los cuales hay ciertas restricciones. La banda ISM, es una banda abierta en todo el mundo que no necesita licencia.

La potencia de transmisión llega hasta 100mW. La distancia nominal en el enlace va de 10cm hasta 10m, pudiéndose alcanzar 100m si se aumenta suficientemente la potencia.

Cuando un equipo Bluetooth está dentro del radio de cobertura de otro, estos pueden crear un enlace entre ellos. Hasta ocho unidades Bluetooth pueden comunicarse entre ellas y forman lo que se denomina una Piconet. La unión de varias piconets se denomina Scatternet.

En todas las Piconets sólo puede haber una unidad maestra que normalmente es quien inicia la conexión, el resto de unidades Bluetooth se denominan esclavas.

Cada unidad de la Piconet utiliza su identidad maestra y reloj nativo para seguir en el canal de salto. Cuando se establece la conexión, se añade un ajuste de reloj a la propia frecuencia de reloj nativa de la unidad esclava para poder sincronizarse con el reloj nativo del maestro. El reloj nativo mantiene siempre constante su frecuencia, sin embargo, los ajustes producidos por las unidades esclavas para sincronizarse con el maestro, sólo son válidos mientras dura la conexión.

Dentro de la misma área pueden coexistir diversas piconets ya que cada piconet tiene una unidad maestra distinta con su propia secuencia de saltos de canales y de fase. A medida que se tienen más piconets en la misma área de cobertura, la probabilidad de colisión aumenta produciendo una degradación del espectro y reduciendo el rendimiento del sistema.

Una unidad Bluetooth puede participar secuencialmente en varias Piconets gracias al sistema TDM (división de tiempo multiplexada). Esto es posible siempre y cuando la unidad solo esté activa en una Piconet a la vez. Para realizar este proceso, la unidad cuando se incorpora a la nueva Piconet debe ajustar el offset de su reloj nativo y realizar los ajustes de configuración correspondientes a la nueva Piconet.

Cuando una unidad abandona una piconet, la esclava informa al maestro actual que ésta no estará disponible por un determinado periodo, que será en el que estará activa en otra piconet. Durante su ausencia, el tráfico en la Piconet entre el maestro y otros esclavos continúan igualmente.

Una unidad maestra también puede cambiar de Piconet, pero en este caso el tráfico de la Piconet, en la cual está activa, deja de tener tráfico hasta la vuelta de la unidad maestra. La maestra que entra en una nueva piconet, en principio, lo hace como esclava, a no ser que posteriormente solicite actuar como maestra.

Canal

La banda ISM tiene muchas interferencias, para solucionar este problema el Bluetooth utiliza un método de salto de frecuencia pseudo-aleatoria llamado FH/TDD (salto de frecuencia/división de tiempo duplex), en el que el canal queda dividido en intervalos de $625 \mu\text{s}$, llamados slots (ranuras), donde cada salto de frecuencia es ocupado por un slot. Esto da lugar a una frecuencia de salto de 1600 veces por segundo, en la que un paquete de datos ocupa un slot para la emisión y otro para la recepción y que pueden ser usados alternativamente, dando lugar a un esquema de tipo TDD. De esta manera se pueden conseguir transceptores de banda estrecha con una gran inmunidad a las interferencias, ver figura 3.3.

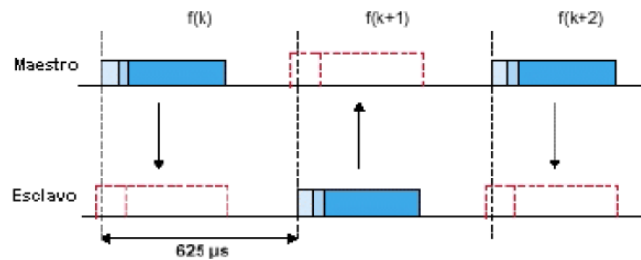


Figura 3.3 Canal de comunicación Bluetooth

En países donde la banda está abierta a 80 canales o más, espaciados todos ellos a 1 Mhz., se han definido 79 saltos de portadora, y en aquellos donde la banda es más estrecha se han definido 23 saltos.

Paquete.

Las unidades Bluetooth intercambian paquetes de datos entre ellas. Cada paquete está formado por un conjunto de slots. En la figura 3.4 se observa la estructura de cada paquete.



Figura 3.4 Estructura de un paquete

Todos los paquetes tienen el mismo formato. Constan de un código de acceso de 72 bits en el cual se puede encontrar la identidad de la unidad maestra, después tienen una cabecera de 54 bits, en los cuales, se puede encontrar información de control, tipo de paquete, bits de control de flujo, chequeo de errores, etc..., y finalmente se puede encontrar el paquete con información, el cual, puede variar su longitud entre 0 y 2754 bits.

Los paquetes de datos están protegidos por un esquema ARQ (repetición automática de consulta), en el cual, los paquetes perdidos son automáticamente

retransmitidos, aun así, con este sistema, si un paquete de datos no llega a su destino, sólo una pequeña parte de la información se perdería.

La voz no se retransmite nunca, sin embargo, se utiliza un esquema de codificación muy robusto basado en una modulación variable de declive delta (CSVD), que sigue la forma de la onda de audio y es muy resistente a los errores de bits. Estos errores son percibidos como ruido de fondo, que se intensifica si los errores aumentan.

Los receptores de la Piconet comparan las señales que reciben con el código de acceso, si éstas no coinciden, el paquete recibido no es considerado como válido en el canal y el resto de su contenido es ignorado.

Enlace físico.

En la especificación Bluetooth se definen dos tipos de enlace, el enlace de sincronización de conexión orientada (SCO) y el enlace asíncrono de baja conexión (ACL).

En el enlace de sincronización de conexión orientada (SCO) se da una conexión simétrica punto a punto entre maestro y esclavo. El maestro utiliza slots de tiempo reservados a intervalos regulares. El maestro puede soportar más de 3 enlaces simultáneos mientras que los esclavos 3 como máximo. Los paquetes de configuración nunca son retransmitidos. Este tipo de enlace se utiliza para transmitir

Información de voz con un ratio de transmisión de 64kB/s.

En el enlace asíncrono de baja conexión (ACL) las conexiones pueden ser simétricas o asimétricas punto-multipunto entre maestro y esclavo. La conexión es utilizada para la transmisión de datos. Se aplica retransmisión de paquetes, y se definen para este tipo de conexión los slots 1,3 y 5. La máxima velocidad de envío es de 721 kbit/s en una dirección y 57.6 kbit/s en la otra.

Establecimiento de una conexión

Para establecer la Piconet, la unidad maestra debe conocer la identidad del resto de unidades que están en modo Standby en su radio de cobertura. El maestro o aquella unidad que inicia la Piconet, transmite el código de acceso continuamente en periodos de 10 ms, que son recibidas por el resto de unidades que se encuentran en Standby. El tren de 10 ms de códigos de acceso de diferentes saltos de portadora, se transmite repetidamente hasta que el receptor responde o bien se excede el tiempo de respuesta.

Cuando una unidad emisora y una receptora seleccionan la misma portadora de salto, la receptora recibe el código de acceso y devuelve una confirmación de recibo de la señal, es entonces cuando la unidad emisora envía un paquete de datos que contiene su identidad y frecuencia de reloj actual. Después de que el receptor acepta éste paquete, ajustará su reloj para seleccionar el canal de salto correcto determinado por emisor. De éste modo se establece una Piconet en la

que la unidad emisora actúa como maestra y la receptora como esclava. Después de haber recibido los paquetes de datos con los códigos de acceso, la unidad maestra debe esperar un procedimiento de requerimiento por parte de las esclavas, diferente al proceso de activación, para poder seleccionar una unidad específica con la que comunicarse.

En la figura 3.5 se observan los estados existentes para realizar una conexión.

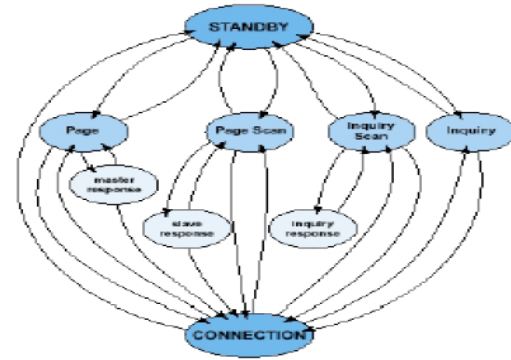


Figura 3.5 Estados de una conexión.

Arquitectura de protocolo.

La especificación Bluetooth define una arquitectura de protocolos semejante a la pila de protocolos de la OSI de redes. Estos protocolos son utilizados dependiendo el tipo de sistema que se quiere implementar. Cada protocolo se puede tratar como una capa independiente con sus correspondientes interfaces para que el sistema sea homogéneo, ver figura 3.6.

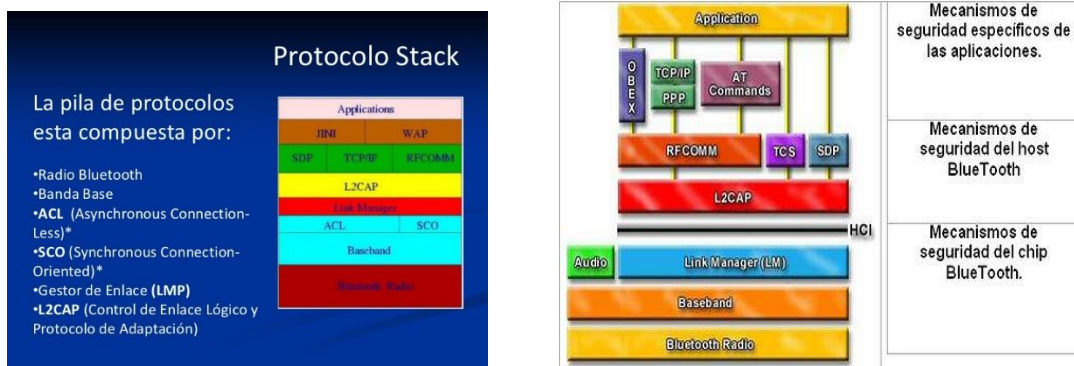


Figura 3.6 Pila de protocolos Bluetooth.

Perfiles.

Para asegurar la interoperabilidad entre dispositivos Bluetooth de diferentes fabricantes, en el estándar bluetooth se han definido una serie de perfiles.

Los perfiles definen los roles y capacidades para aplicaciones específicas. Diferentes perfiles pueden abarcar diferentes capas y protocolos para diferentes grados de seguridad.

En la figura 3.7 se observa la estructura de los perfiles que se definen para el estándar Bluetooth.

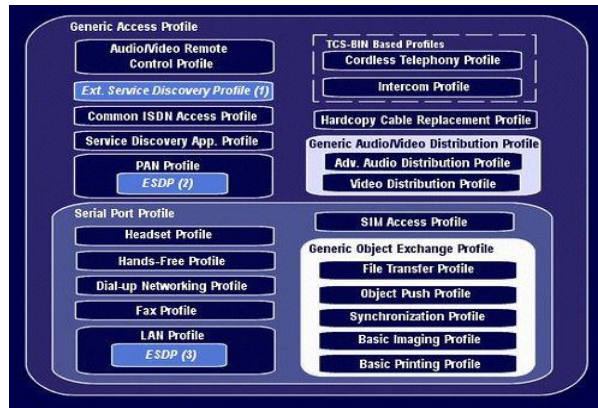


Figura 3.7. Estructura de los perfiles

Seguridad.

Los Niveles de seguridad definidos en el estándar Bluetooth son para asegurar la protección de la información.

Las medidas de seguridad se basan en tres puntos: Una rutina de pregunta-respuesta, para autenticación; Una corriente cifrada de datos, para encriptación; y Generación de una clave de sesión (que puede ser cambiada durante la conexión).

Tres entidades son utilizadas en los algoritmos de seguridad: la dirección de la unidad Bluetooth, que es una entidad pública; una clave de usuario privada, como una entidad secreta; y un número aleatorio, que es diferente por cada nueva transacción.

La dirección Bluetooth se puede obtener a través de un procedimiento de consulta. La clave privada se deriva durante la inicialización y no es revelada posteriormente. El número aleatorio se genera en un proceso pseudo-aleatorio en cada unidad Bluetooth.

CAPÍTULO 4

ANÁLISIS Y DISEÑO DEL KIP

En este capítulo se describe el análisis y diseño del sistema KIP.

4.1. ANÁLISIS

En el presente trabajo se persigue el diseño e implementación de un sistema con módulos periféricos de E/S para la experimentación en las materias de Diseño Digital y Microcontroladores. Además de establecer las bases de comunicación inalámbrica entre un módulo maestro y un módulo esclavo, utilizando tecnología Bluetooth.

Esto indica que el KIP debe contener periféricos de entrada y periféricos de salida. Hay muchos periféricos que se pueden conectar a un sistema digital y a un microcontrolador, pero hay periféricos de E/S comunes que soportan prácticas de ambas materias. Dentro de los periféricos de salida se tienen a leds para presentar información binaria, a los Displays para presentar información en BDC o hexadecimal, al LCD para presentar el código alfanumérico, a los servomotores para controlar motores con características especiales. Dentro de los periféricos de entradas digitales se tienen al teclado lineal y sensores ópticos, los cuales suministran datos binarios por tecla o por sensor.

Para la parte de comunicación inalámbrica, se utilizan microcontroladores PIC con módulos Bluetooth para establecer una comunicación punto a punto entre un módulo maestro y un módulo esclavo.

El KIP debe considerar las siguientes características.

En cuanto a alimentación.

- Que el KIP se pueda alimentar con su propia fuente de alimentación o por USB conectado a la computadora.

En cuanto a los módulos o periféricos.

- Que sean independientes.
- Que se conecten con buses (macho o hembra) o cable del No. 22.
- Que se utilicen indistintamente en aplicaciones de microcontroladores o de Digitales.
- Que estén en impreso.
- Que los componentes de los módulos sean los más comunes en el mercado.
- Que sean replicables o fáciles de producir en serie.
- Que se monten en una mampara pero que sean movibles.

En cuanto al microcontrolador.

- Que sea el más utilizado y el más barato
- Que permita trabajar la parte académica (didáctica).
- Que el firmware de las tarjetas inalámbricas se pueda actualizar o modificar por el usuario.
- Que el software para realizar las prácticas de microcontroladores se disponga en un CD así como su documentación.

En cuanto a las prácticas.

- Que haya un conjunto de prácticas de digitales y de microcontroladores para manejar los periféricos.
- Que el conjunto de prácticas sean una guía para poder desarrollar más prácticas por parte de profesores o alumnos.
- Que las prácticas estén documentadas.

4.2. DISEÑO DEL KIP

Partiendo de las características que se quieren del KIP (o módulos), se presentan los aspectos del diseño.

4.2.1 Alimentación

Esta parte del sistema es por default, sin la cual, ninguna aplicación electrónica funciona. Para la alimentación se propone un módulo único de una fuente de 5Vcc o una conexión a USB conectado a una computadora.

4.2.2 Módulos de Entrada Salida (E/S)

Este es el primer aspecto a considerar ya que influye en el diseño. De acuerdo a la característica de que los módulos sean independientes, el diseño debe considerar módulos únicos y que permitan desarrollar de manera fácil diferentes prácticas.

Esta propuesta permite flexibilidad en la manipulación y conexión de periféricos, reparación o sustitución de una tarjeta dañada sin afectar a las demás.

4.2.3 Microcontrolador

La manipulación de microcontroladores requiere se cubran requisitos físicos, mecánicos, eléctricos, protocolos que el sistema KIP requiere.

El microcontrolador, debe satisfacer los requisitos señalados en la unidad de Bluetooth y los requisitos del proyecto. Para tomar la decisión se tendrán en cuenta los siguientes parámetros:

Velocidad de cálculo. La CPU debe ejecutar instrucciones a una velocidad suficiente como para realizar las operaciones de E/S y los cálculos necesarios en un intervalo de tiempo máximo. En la actualidad casi todos los microcontroladores tienen una frecuencia de reloj variable que se puede ajustar según los requisitos para llegar a un compromiso entre potencia de cómputo y consumo eléctrico.

Módulos integrados. El proyecto requiere la elección de un microcontrolador que tenga incorporados módulos que le permitan realizar ciertas tareas como comunicación Bluetooth y manejo de puertos paralelos y seriales de E/S. En la actualidad muchos microcontroladores contienen estos módulos y más.

Factor de forma y encapsulado. Resulta importante que el encapsulado del chip sea adecuado para la fabricación del KIP, esto es, que tenga un tamaño y una configuración de pines apta para su montaje en un circuito impreso. Los fabricantes suelen ofrecer distintos encapsulados para el mismo dispositivo, con lo que es posible utilizar uno para la fase de construcción del prototipo y otro diferente para el producto final.

Herramientas de desarrollo y documentación disponibles. El fabricante debe proporcionar la documentación y las herramientas de desarrollo, para desarrollar diversos proyectos y ejemplos disponibles, lo cual es de gran ayuda.

Pines de E/S. El microcontrolador debe contar con suficientes pines de entrada y salida para la aplicación, lo que debe evaluarse teniendo en cuenta la arquitectura del sistema que se haya diseñado para poder cuantificar esta necesidad.

Consumo. El consumo debe ser adecuado para que pueda funcionar exclusivamente con la corriente suministrada por el bus USB, principalmente. Muchos microcontroladores están preparados para funcionar con poca corriente o con baterías y disponen de modos especiales de bajo consumo.

Memoria. Debido a que las aplicaciones son pequeñas, la cantidad de memoria no es mucho problema.

Arquitectura. En muchos casos, los microcontroladores de 8 bits son suficientes, pero hay que evaluar la potencia de cálculo necesaria para seleccionar la arquitectura de menor coste que se adapte a los requisitos del proyecto.

4.2.4 Herramientas de programación

Es importante seleccionar los lenguajes de programación adecuados a la aplicación. El lenguaje influye mucho en el tiempo de desarrollo del software.

Lenguaje de programación para el microcontrolador.

Dentro de los lenguajes más utilizados para programar microcontroladores destacan C, Basic y ensamblador.

Lenguaje C para microcontroladores. C ha demostrado por décadas su flexibilidad, puesto que con él se han programado compiladores, juegos, sistemas operativos, drivers, etc. C es un lenguaje de medio nivel, está cerca del lenguaje máquina (bajo nivel) y está cerca del usuario (alto nivel), permitiendo trabajar punteros, direcciones de memoria.

Basic. Desarrollado inicialmente como una herramienta fácil de usar para principiantes, con una sintaxis sencilla y cercana al lenguaje natural o pseudocódigo. Por ello es el preferido de aficionados que no son programadores. Por esta razón se encuentran en la red gran cantidad de recursos relacionados con el lenguaje.

Ensamblador. Es el lenguaje de más bajo nivel y más cercano a la máquina, lo cual lo hace dependiente de la arquitectura del microcontrolador. Se escribe código eficiente, por lo cual hay que conocer muy bien la arquitectura. Es muy poco portable y difícil de mantener. Su uso está extendido para especialistas en programar aplicaciones para optimizar el rendimiento y el consumo.

4.2.5 Componentes del diseño

A partir de las alternativas presentadas para el desarrollo de este proyecto, se pasa a la toma de decisiones sobre la más apropiada en cada caso.

Primero se debe definir cómo van a quedar los módulos y que permitan la flexibilidad de aplicaciones, situaciones ante fallos, producción en serie, comunicación inalámbrica, entre otras.

La propuesta de los periféricos es que estén cada uno en su propia placa. Esto tiene sus ventajas: los periféricos son independientes., si falta una placa o se quiere hacer una nueva esta se incorpora., hay flexibilidad en incorporación de nuevos periféricos., se arman y prueban rápidamente las prácticas., ante un fallo sólo se atiende la tarjeta de interés y no se dañan las demás., los periféricos son manipulables., el circuito impreso es muy fácil de hacer con técnicas caseras., los periféricos se utilizan sin la necesidad de utilizar a los demás periféricos. Dentro de las desventajas son que el espacio que se ocupa por tarjeta es grande, sin embargo, vale la pena ya que es para cuestiones didácticas.

Por lo anterior se propone desarrollar el KIP con tarjetas o módulos independientes de periféricos de E/S

En cuanto a microcontroladores, hay muchas firmas que los producen como Intel, Microchip, Arduino, AVR, Raspberry, Motorola, etc. Se escoge a Microchip porque dispone de documentación suficiente. Ofrece herramientas gratis para desarrollar proyectos así como ejemplos. Tiene una página Web en donde se puede consultar y recibir información de todo tipo. Hay una gran variedad de programadores de microcontroladores baratos, incluso con ejemplos y asesorías. Hay diferentes gamas y la gama elegida, ofrece todo para el desarrollo del proyecto.

4.2.6 Diseño

En la figura 4.1 se muestran los módulos que conforman al sistema Kit inalámbrico de periféricos. En ella se puede observar los módulos de entrada y módulos de salida.

KIT INALÁMBRICO DE PERIFÉRICOS

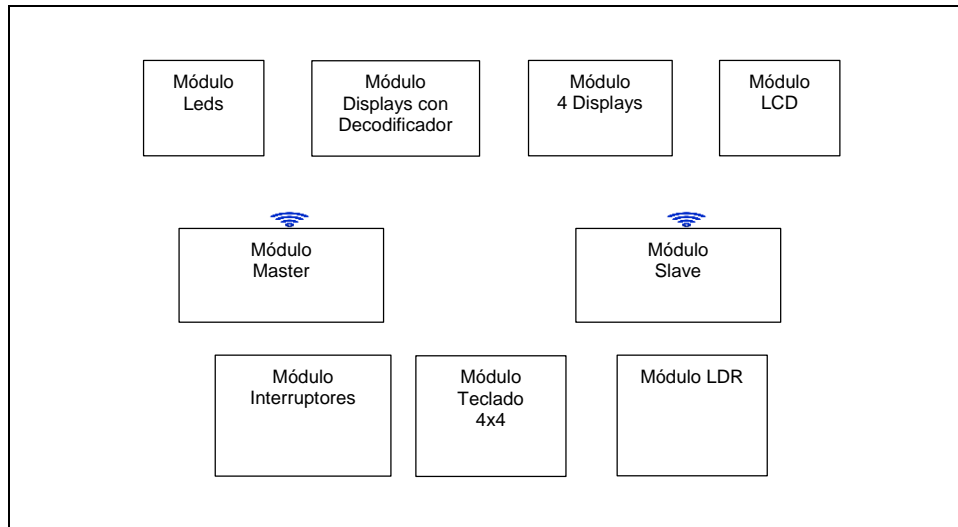


Figura 4.1 Arquitectura del KIP

De la arquitectura general del KIP, se observa en la parte superior todos los módulos periféricos de salida. En la parte de en medio los módulos de comunicación Bluetooth maestro y esclavo, y en la parte inferior, los módulos periféricos de entrada.

4.3 DESCRIPCIÓN DE CADA MÓDULO

4.3.1 MÓDULOS DE SALIDA.

Módulo Leds

Funcionalidad. Este módulo se requiere para presentar información binaria. Por ello, los valores eléctricos discretos o valores binarios que recibe este módulo los convierte en información luminosa por medio de diodos emisores de luz (led's). Un valor binario 0 se representa por un led apagado y un valor binario 1 se representa por un led encendido. El módulo podrá representar 8 valores binarios desde el valor "00000000" hasta el "11111111". El módulo permitirá trabajar de forma individual, por grupos de 4 bits (2 nibbles) o en su totalidad (1 byte) todos los leds. Este módulo soporta prácticas de Diseño Digital y Microcontroladores.

Datos de entrada. El módulo de leds recibe hasta 8 niveles de información binaria provenientes de un circuito Digital o un Microcontrolador con niveles de voltaje discretos de 0Vcc para el 0 lógico y 5Vcc para el 1 lógico.

Datos de salida. Los datos de salida se presentan en un arreglo lineal de 8 leds. Un led encendido representa un valor lógico 1 y, un led apagado representa un valor lógico 0. La información binaria podrá representar código binario, gray, exceso a 3, etc., según sea la práctica.

Módulo Display con decodificador.

Funcionalidad. Este módulo se requiere para representar información en BCD, por ello, el módulo recibe 4 valores binarios (nibble), los cuales los codifica a su forma decimal para ser representados en un display de 7 segmentos. Este módulo permitirá representar valores decimales 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 de las 15 posibles combinaciones binarias que se dan en la entrada. Este módulo soporta prácticas de Digitales y Microcontroladores.

Datos de entrada. El módulo recibe dos grupos de 4 bits provenientes de un circuito Digital o un Microcontrolador con niveles discretos de 0Vcc y 5 Vcc.

Datos de salida. La salida al display de 7 segmentos requiere 7 valores binarios y estos los proporciona el decodificador de BCD.

Módulo 4 Displays.

Funcionalidad. Este módulo se requiere para representar el código en hexadecimal (0 hasta F) en display de 7 segmentos, además de trabajarlos para representar hasta 4 valores hexadecimales. Por ello, este módulo recibe dos tipos de valores binarios, un tipo de valores para habilitar directamente los 7 segmentos de los displays y generar valores hexadecimales en los displays de 7 segmentos, y otros para seleccionar el display donde se desplegará el valor generado. Como se manejan directamente los segmentos de los displays, se puede codificar los valores 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. este módulo soporta prácticas de Digitales y microcontroladores

Datos de entrada. El módulo recibe dos grupos de valores binarios. 7 para generar el código del 0 a F, y 4 para controlar uno de los 4 displays donde se exhibirá el dato codificado. Los datos binarios pueden provenir de un circuito Digital o un Microcontrolador.

Datos de salida. La salida será un grupo de 4 Displays que presentará el valor Hexadecimal generado por la entrada en el display elegido.

Módulo LCD.

Funcionalidad. Este módulo se utilizará para desplegar mensajes u otro tipo de información que cambie constantemente. Para ello el módulo recibe los datos a desplegar en el LCD así como un conjunto de señales de control y de alimentación. El LCD a utilizar es de 2 líneas de 16 caracteres cada una. Este módulo avanzado soporta prácticas de microcontroladores.

Datos de entrada. El módulo recibe el dato de 8 bits a desplegar, así como las señales de control de habilitación (E), Selección de dato/control (RS) y lectura/escritura en LCD (R/W). Los datos binarios provienen de un Microcontrolador.

Datos de salida. La salida será un conjunto de caracteres distribuidos en las dos líneas de 16 caracteres cada una del LCD.

4.3.2 MÓDULOS DE ENTRADA.

Módulo de Interruptores

Funcionalidad. Este módulo se utiliza para proporcionar niveles de voltajes binarios de 0Vcc y 5Vcc para el 0 y 1 lógico respectivamente. El módulo ofrece 8 valores binarios que van desde el "00000000" hasta el "11111111". Dependiendo de las aplicaciones los interruptores podrán manejarse de manera individual, por grupos de 4 bits (2 nibbles) o todos (1 byte). La información binaria generada puede representar código binario, gray, exceso a 3, etc., según sea la práctica. Este módulo se utiliza en prácticas de Diseño Digital y Microcontroladores.

Datos de entrada. El módulo no recibe dato alguno, sólo requiere 5Vcc para generar valores binarios.

Datos de salida. El módulo es un arreglo lineal de 8 interruptores, y por cada interruptor se genera un nivel alto (5Vcc) o un nivel bajo (0Vcc).

Módulo teclado 4x4

Funcionalidad. Este módulo es un arreglo matricial de 4 filas por 4 columnas. Cada tecla conecta una fila y una columna que el usuario debe de detectar, decodificar y generar el código de la tecla oprimida. Esto se puede hacer por hardware o por software, dependiendo de la práctica. EL teclado genera 16 valores y se recomienda para prácticas con microcontroladores.

Datos de entrada. El módulo no recibe dato alguno, sólo requiere 5Vcc y un arreglo matricial de 4 filas y 4 columnas para generar valores binarios de 8 bits.

Datos de salida. Este módulo matricial genera 16 valores de 8 bits cada uno. Estos valores el usuario debe manipularlos según su práctica.

Módulo LDR

Funcionalidad. Este módulo se utiliza para proporcionar niveles de voltajes binarios de 0Vcc y 5Vcc para el 0 y 1 a partir de manipular el paso o no paso de luz. El módulo ofrece 8 valores binarios que van desde el "00000000" hasta el "11111111". Dependiendo de las aplicaciones los interruptores podrán manejarse de manera individual, por grupos de 4 bits (2 nibbles) o todos (1 byte). La información binaria generada puede representar código binario, gray, exceso a 3, etc., según sea la práctica. Este módulo se utiliza en prácticas de Diseño Digital y Microcontroladores.

Datos de entrada. El módulo recibe o no recibe luz de alguna fuente luminosa, con lo cual se excita a 8 sensores ópticos.

Datos de salida. A partir de la luz recibida o no recibida, el módulo generará 8 niveles de voltaje discretos e 5Vcc o 0Vcc.

4.3.3 MÓDULOS MICROS.

Funcionalidad. Este módulo debe proporcionar las señales del microcontrolador que se utilizará para realizar prácticas con microcontroladores o para la comunicación bluetooth.

Las líneas de entrada y salida se deben de disponer por puertos RA, RB, RC, etc., así como las líneas de alimentación, que permitan la conexión de módulos o circuitos en general.

4.3.4 MÓDULOS BLUETOOTH

Funcionalidad. Este módulo debe proporcionar las señales del módulo Bluetooth que se conectará a los microcontroladores para soportar la comunicación bluetooth.

Las líneas de entrada y salida se deben de disponer para conectarse a los microcontroladores.

CAPÍTULO 5

IMPLEMENTACIÓN DEL KIP

En este capítulo se presenta el desarrollo de cada módulo.

5.1 MÓDULO LEDS

Este módulo tiene la función de desplegar el monitoreo de niveles lógicos provenientes de circuito Digitales, microcontroladores u otros.

En la figura 5.1 se muestra el circuito del módulo leds. El circuito está formado por diodos emisores de luz leds (E0 – E7) y 8 resistencias.

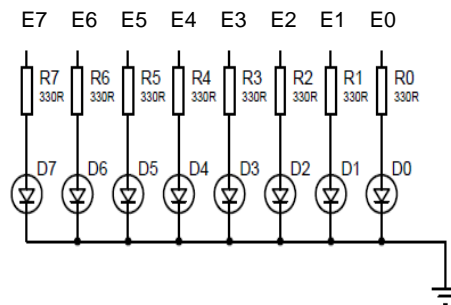


Figura 5.1 Circuito del Módulo leds.

Cada una de las entradas del circuito E_i ($i = 0, \dots, 7$) reciben dos valores de voltaje, uno de $5V_{cc}$ para un 1 lógico y $0V_{cc}$ para un 0 lógico. El voltaje alimenta al arreglo serial de resistencia y led.

Analizando la malla formada por el voltaje de entrada la resistencia y el led, se obtiene la ecuación 1.

$$V_{ENTRADA} = V_R + V_{LED} \quad \text{Ec. 1}$$

Por ley de Ohm y datos técnicos del Led se tiene:

$$\begin{aligned} \text{Ley de Ohm} & \quad V_R = R \cdot I_{LED} \\ \text{Datos del led:} & \quad I_{LED} = 10\text{mA} \\ & \quad V_{LED} = 1.4 V_{CC} \end{aligned}$$

Sustituyendo y despejando R de la ec.1 se tiene:

$$\begin{aligned} V_{ENTRADA} &= V_R + V_{LED} \\ 5V_{CC} &= R \cdot I_{LED} + 1.4V_{CC} \\ R &= (5V_{CC} - 1.4V_{CC}) / 10\text{mA} = 360 \Omega \end{aligned}$$

En el mercado sólo hay valores de resistencias de 330Ω o 390Ω , se escoge el primer valor.

Para simular se eligió el software Proteus. Este software es una herramienta CAD (computer aided design) y permite comprobar diseños, editar, capturar esquemas, simular, hacer Artworks o circuitos impresos, a parte de otras tantas funciones. En la figura 5.2 se muestra la pantalla principal de Proteus.

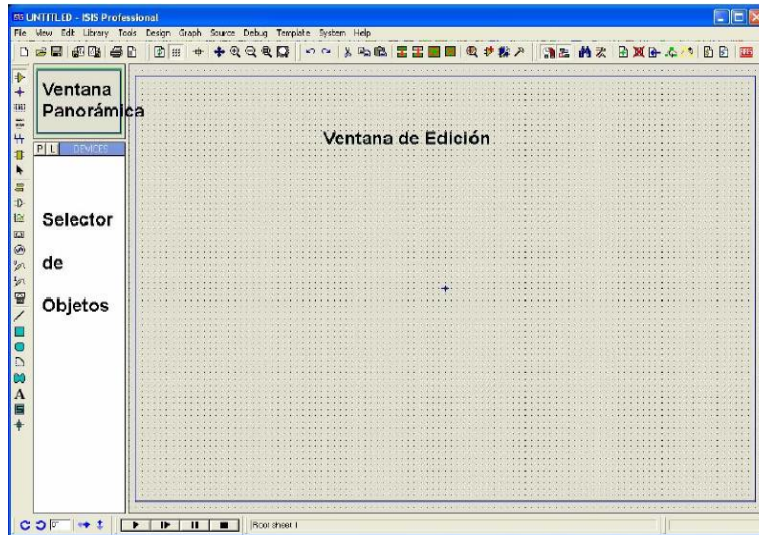


Figura 5.2 Pantalla principal de Proteus

En la ventana de edición se arrastran los componentes del circuito. Los componentes se seleccionan al dar click al ícono P (Pick devices) mostrado en la ventana de selector de objetos. Los ícono verticales y horizontales se eligen de acuerdo a la función que ofrecen.

Con este software se capturó y simuló el comportamiento el circuito del módulo LEDs, el cual, se muestra en la figura 5.3. Algunos leds están encendidos y otros apagados debido a la entrada binaria "10101010".

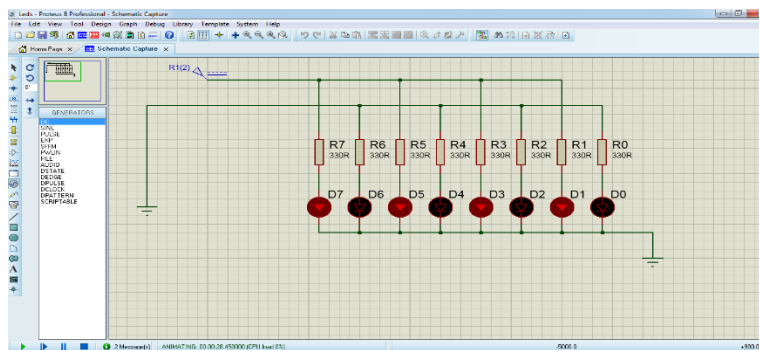


Figura 5.3 Simulación del Módulo LEDs con SW Proteus

Una parte importante de este simulador es que si los valores de las resistencias se ponen en K Ω (K = Kilos o Miles), los leds del circuito no se encienden por el alto valor de las resistencias.

La revisión rápida de lo que es Proteus sólo se describió en el desarrollo de este módulo. Para los demás módulos sólo se presentan las simulaciones.

Ahora se pasa a la parte de realizar el circuito impreso. Desde que se hace la captura esquemática hasta que se tiene la tarjeta electrónica con componentes montados y soldados. El método que se utilizó fue el método de planchado. Dicho método sólo se explica en la realización de este módulo, para los demás módulos sólo se presentan imágenes de cada paso.

Lo primero que se hizo fue conseguir el material y equipo, el cual fue:

Material del circuito del módulo Leds

- 8 Resistencias de 330 Ω a ¼ de Watt
- 8 Leds rojos de 5mm
- Un Clemax
- 1 tren de 8 pines macho
- 1 tren de 8 pines hembra

Material para impresos.

- Placa fenólica de 1 cara de cobre.
- Lija No. 500 de agua
- Papel Couche, papel encerado de revistas o acetato para impresoras laser.
- Tinaja pequeña de 3 plástico
- Percloruro férrico
- Algodón
- Lija
- Plumón punto fino de tinta permanente negro.
- Impresora Laser
- Agua.
- Detergente.
- Plancha con cubierta de teflón.

Material y equipo para montar los componentes en los circuitos impresos.

- Soldadura.
- Cautín de estación.
- Chupón.
- Pinzas de corte.
- Pinzas de punta.

Procedimiento para hacer el circuito impreso.

1.- Lo primero que se hizo fue utilizar el software PCB Wizard, con el cual, se realizaría la captura del circuito del módulo de leds para obtener el Artwork o pistas a grabarse en la placa de cobre.

La pantalla principal del software PCB Wizard se muestra en la figura 5.4.

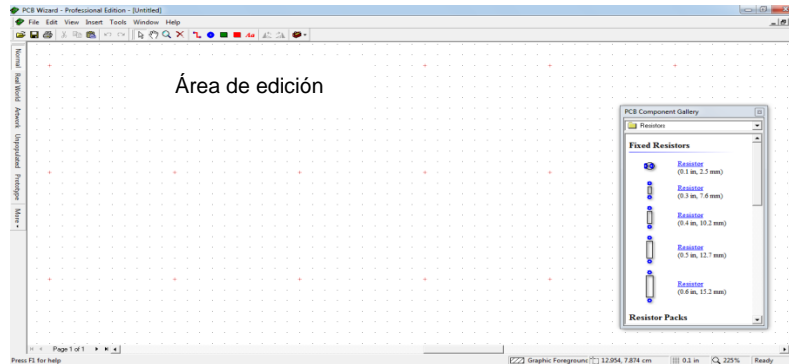
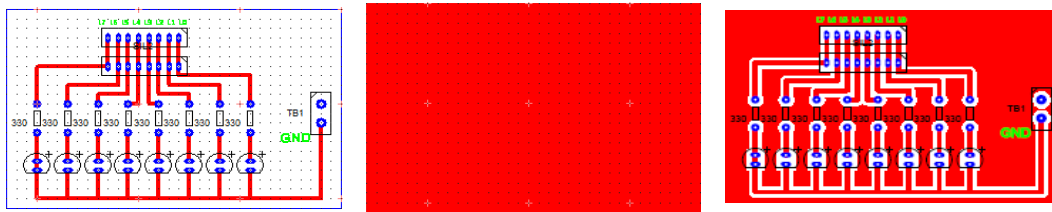


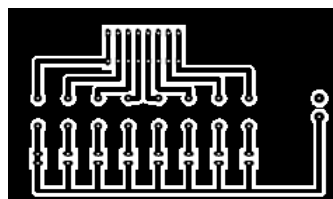
Figura 5.4 Pantalla principal de PCB wizard

El área de edición permite que en ella se realice la captura esquemática, para lo cual, en dicha área se arrastran los componentes a utilizar y luego se conectan de forma manual (cuestiones didácticas). Los componentes se seleccionan del submenú Galería de Componentes del PCB (PCB Components Gallery).

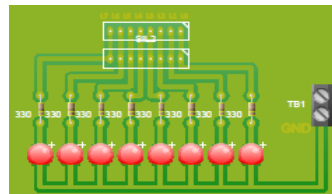
En la figura 5.5.a se muestra a la izquierda la captura esquemática (vista normal) con el cuadro que delimita el área, las islas (color rojo) definen espacios de cobre que se quedan en el impreso pero que no conectan nada. En la figura 5.5.b se muestra la vista Artwork o pistas que se grabarán en la placa de cobre y en la figura 5.5.c se muestra una vista en 3D (mundo real).



a) Captura del circuito del módulo leds



b). Artwork

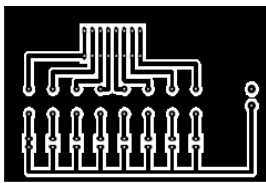


c). 3D

Figura 5.5 Vistas diferentes del módulo Leds con PCB Wizard

La vista Artwork se manda a imprimir en un acetato, en papel encerado de revista o papel cuche. La impresora debe ser laser.

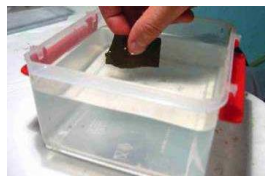
Para grabar la impresión en la placa fenólica primero se recortó el área de pistas (Artwork), figura 5.6a. Se cortó la placa fenólica al tamaño de la impresión del Artwork, figura 5.6b. Se lavó la placa con detergente y agua para eliminar grasa de la placa y de las manos. Después se lijó la placa hasta que quedara una superficie homogénea y reflejante sin nada de manchas en la placa, figura 5.6c. Después se secó la placa, se le puso encima del cobre la impresión Artwork y se planchó, figura. 5.6d. Se retiró el papel (o acetato) de la placa y se metió al percloruro férrico, figura 5.6e.



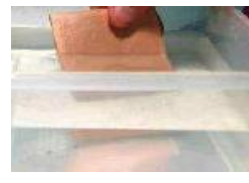
a) Recorte de Artwork



b) Recorte placa



c) Lijado en agua de la placa fenólica



d) Secado y planchado para grabar Artwork



e) Se quita papel y se mete la tarjeta al percloruro férrico.



Figura 5.6 Pasos para grabar el circuito impreso

Después de 15 minutos (o más según la saturación de cobre del percloruro férrico) se retiró la placa del percloruro férrico con pinzas de plástico, se enjuaga y reviso si el grabado quedó bien, tal y como en el Artwork. Quedando bien grabado el circuito se pasó a perforar la placa sobre los orificios o donas del impreso, figura 5.7a. Se insertaron los componentes, figura 5.7.b, y se soldaron, figura 5.7c.



a).Perforación de donas



b) Inserción de componentes



c) Soldado de componentes.

Figura 5.7 Perforación de donas, inserción y soldado de componentes

Finalmente se probó el circuito, el cual se aprecia en la figura 5.8.

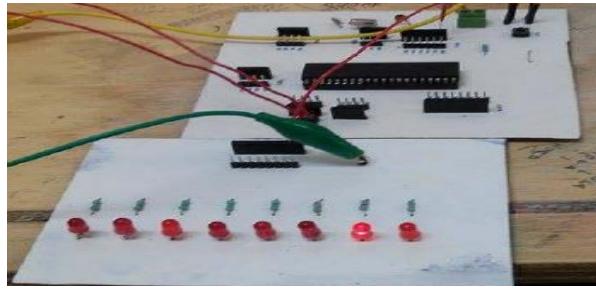


Figura 5.8 Prueba de la funcionalidad de la tarjeta final Módulo leds.

El proceso de grabado hasta el soldado de componentes se siguió con todas las tarjetas, las cuales se muestran en la figura 5.9.

Cabe mencionar que para todas las entradas y/o salidas de todos los módulos, se utilizaron dos trenes de pines, machos y hembras, conectados en paralelo, para cuestiones didácticas.



Figura 5.9 Grabado, montaje y soldado de todas las tarjetas.

5.2 MÓDULO DISPLAY CON DECODIFICADOR

Este módulo tiene la función de mostrar valores decimales del 0 al 9, figura 5.10.a, a partir de valores binarios procedentes de circuitos digitales, microcontroladores u otros.

El circuito que despliega valores decimales a partir de valores binarios se muestra en la figura 5.10b. El circuito está conformado por un decodificador de BCD a 7 segmentos y un Display de 7 segmentos de ánodo común.

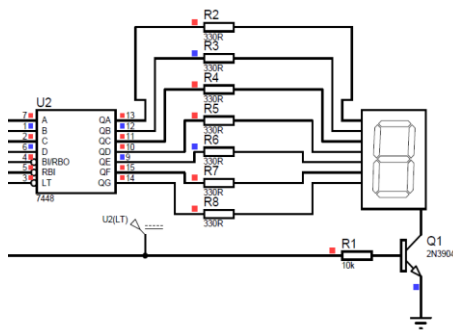
El decodificador de BCD a 7 segmentos (74LS48) tiene siete entradas. Cuatro entradas binarias (A, B, C, D) definen un valor binario del "0000" al "1111". Si las tres entradas restantes (LT', RBI, RBO') están en alto (1) permitirán que el 74LS48 genere en sus siete salidas (a, b, c, d, e, f, g), el código decimal que se mostrará en un display de 7 segmentos.

Las siete salidas del 74LS40 se conectan a través de resistencias al display de ánodo común, de acuerdo al segmento que corresponden. El transistor NPN conectado al display sirve para proporcionarle mayor corriente a los segmentos y se pueda visualizar con buena intensidad los números decimales mostrados.

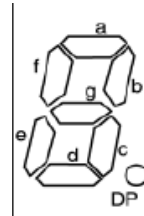
El 74LS48 genera señales en alto activo, por lo que se conecta a un display de cátodo común, figura 5.10c.



a) Decimales (0-9) mostrados en un display



b) Circuito Display con decodificador



c). Nombres de los segmentos del display

Figura 5.10 Circuito del Módulo Display con decodificador.

Tabla 5.1 Tabla de verdad del 74LS48

**TRUTH TABLE
SN54/74LS48**

DECIMAL OR FUNCTION	INPUTS						OUTPUTS							NOTE
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	D	C	B	A	$\overline{\text{BI/RBO}}$	a	b	c	d	e	f	
0	H	H	L	L	L	L	H	H	H	H	H	H	L	1
1	H	X	L	L	L	H	H	L	H	H	L	L	L	1
2	H	X	L	L	H	L	H	H	H	L	H	H	L	H
3	H	X	L	L	H	H	H	H	H	H	L	L	H	
4	H	X	L	H	L	L	H	L	H	H	L	L	H	H
5	H	X	L	H	L	H	H	H	L	H	H	L	H	H
6	H	X	L	H	H	L	H	L	L	H	H	H	H	H
7	H	X	L	H	H	H	H	H	H	H	L	L	L	L
8	H	X	H	L	L	L	H	H	H	H	H	H	H	H
9	H	X	H	L	L	H	H	H	H	L	L	L	H	H
10	H	X	H	L	H	L	H	L	L	L	H	H	L	H
11	H	X	H	L	H	H	H	L	L	H	H	L	L	H
12	H	X	H	H	L	L	H	L	H	L	L	L	H	H
13	H	X	H	H	L	H	H	H	L	L	H	L	H	H
14	H	X	H	H	H	L	H	L	L	L	H	H	H	H
15	H	X	H	H	H	H	H	L	L	L	L	L	L	L
BI	X	X	X	X	X	X	L	L	L	L	L	L	L	2
RBI	H	L	L	L	L	L	L	L	L	L	L	L	L	3
LT	L	X	X	X	X	X	H	H	H	H	H	H	H	4

La simulación del módulo Display con decodificador realizada con Proteus se muestra en la figura 5.11

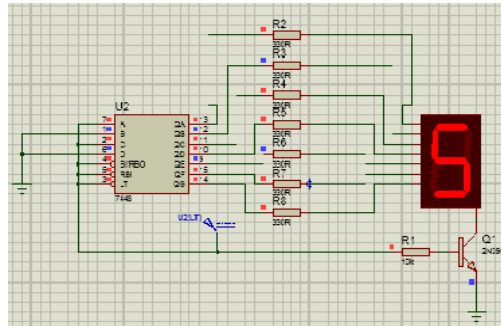


Figura 5.11 Simulación del módulo Display con decodificador con Proteus

El circuito capturado del módulo Display con decodificador, el Artwork y la vista en 3D se muestran en la figura 5.12

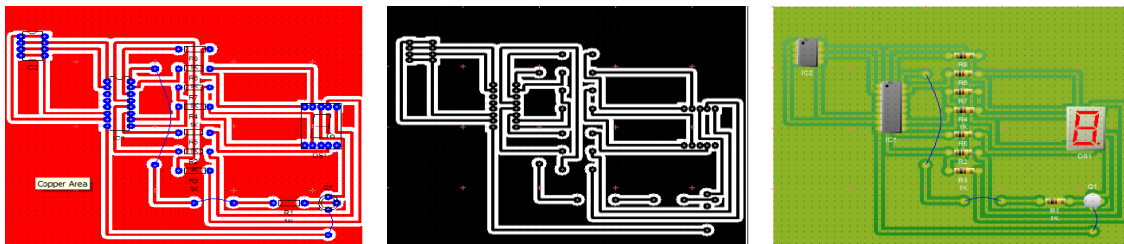


Figura 5.12 Captura del circuito, Artwork y vista en 3D del módulo Display con decodificador

Después de que se grabó el Artwork a la placa, se perforó, se insertaron los componentes y soldaron, se obtuvo la tarjeta final del módulo Display con decodificador y se muestra en la figura 5.13.

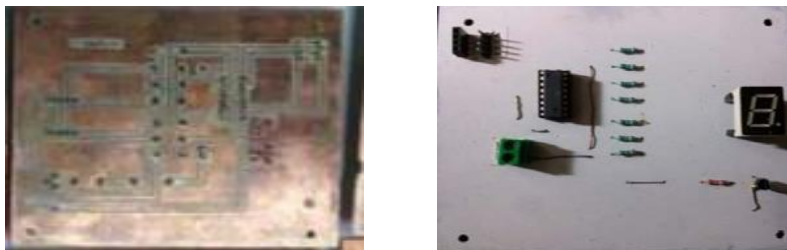


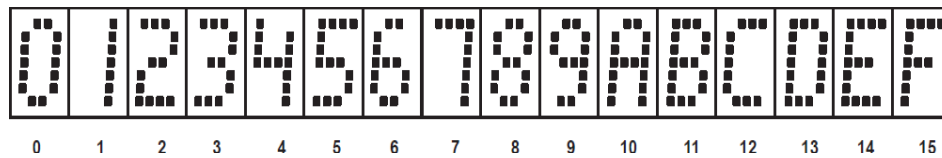
Figura 5.13 Circuito impreso y tarjeta del módulo Display con decodificador

5.3 MÓDULO 4 DISPLAYS

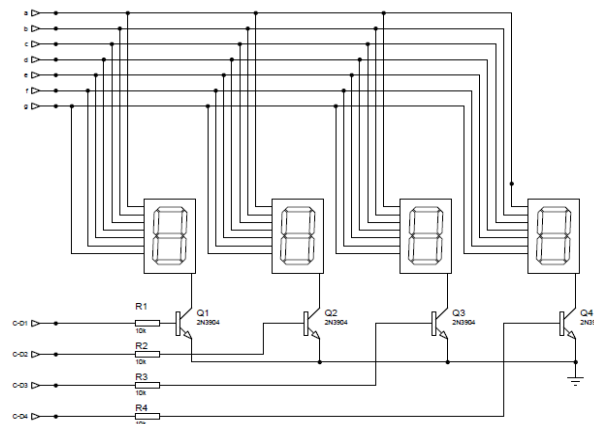
Este módulo tiene la función de desplegar códigos en hexadecimal de 0 a F, figura 5.14a, en 4 displays de 7 segmentos, de información proveniente de circuitos Digitales, microcontroladores u otros.

El circuito del módulo 4 displays se muestra en la figura 5.14b. Este circuito está conformado por cuatro display de 7 segmentos y cada display tiene su propio transistor. Este circuito tiene once entradas, siete de las cuales corresponden a los segmentos (a, b, c, d, e, f, g) de los displays, los cuales se encuentran en paralelo. Es decir, si por la entrada “a” llega un valor alto (1), este valor llega a todos los segmentos “a” de todos los displays.

Las otras cuatro entradas (C-D1, C-D2, C-D3, C-D4) habilitan a un transistor y este al display correspondiente, si el valor que llega es un alto, ver figura 5.14b. El transistor NPN da mayor corriente al display y se mejora la intensidad de luz de los segmentos.



a) Código hexadecimal en un display de 7 segmentos



b) Circuito del Módulo 4 Displays

Figura 5.14 Código hexadecimal y Circuito del Módulo 4 Displays

El valor del resistor de la base del transistor es de 10 KΩ., que es un valor común para este tipo de configuración.

La simulación del módulo 4 Displays con Proteus se muestra en la figura 5.15.

El circuito capturado del módulo 4 Displays, el Artwork y la vista en 3D se muestran en la figura 5.16

Después de que se grabó el Artwork a la placa, se perforó, se insertaron los componentes y soldaron, la tarjeta final del módulo 4 Displays se muestra en la figura 5.17.

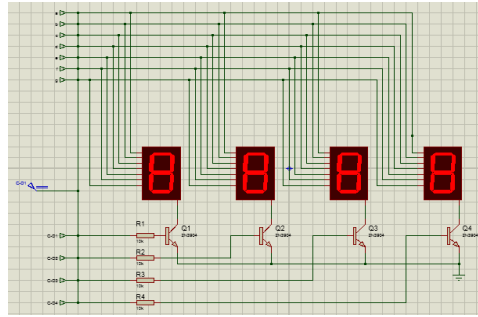


Figura 5.15 Simulación del Módulo 4 Displays

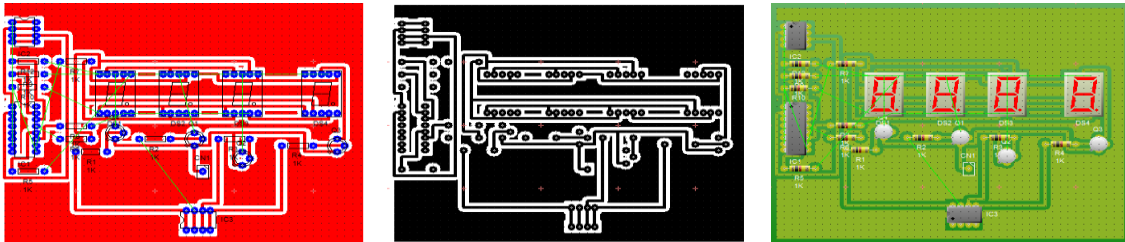


Figura 5.16 Captura del circuito, Artwork y vista en 3D del módulo 4 Displays.



Figura 5.17 Circuito impreso y Tarjeta del módulo 4 Displays.

5.4 MÓDULO LCD

Este módulo tiene la función de desplegar mensajes en un LCD (Display de Cristal Líquido), de información proveniente de circuitos Digitales, microcontroladores u otros. Hay varios tipos de LCDs y se eligió al de 2 líneas por 16 caracteres, que es el más común para cuestiones didácticas. En la figura 5.18. se muestra al LCD 2x16 y su distribución de líneas.

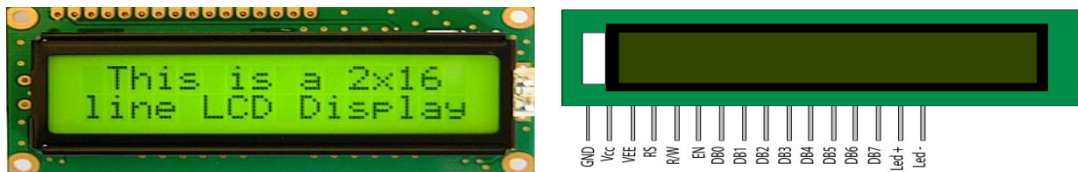


Figura 5.18 LCD 2x16

El circuito del módulo LCD, se muestra en la figura 5.19. En dicha tarjeta se identifican al LCD y un potenciómetro. El potenciómetro es sólo para ajustar el contraste.

El funcionamiento del LCD está sujeto a las funciones de sus pines, estos se muestran en la tabla 5.2.

Tabla 5.2 Funciones de los pines del LCD

Terminal	Símbolo	Nombre o función
1	Vss	Tierra - GND
2	Vdd	Alimentación +5V
3	Vo	Ajuste de voltaje de contraste
4	(RS)´	Selección Dato/Control
5	R/W´	Lectura/Escritura´
6	E	Enable (habilitación)
7	D0	D0 Bit Menos Significativo
8	D1	D1
9	D2	D2
10	D3	D3
11	D4	D4
12	D5	D5
13	D6	D6
14	D7	D7 Bit Más Significativo

El circuito del módulo LCD se muestra en la figura 5.19.

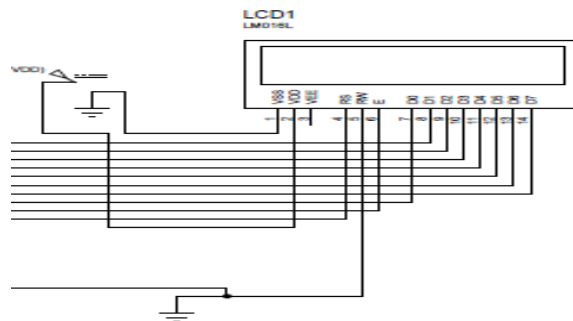


Figura 5.19 Circuito del Módulo LCD

De acuerdo a la operación que se desee realizar sobre el módulo de cristal líquido o LCD, los pines de control E, RS´ Y R/W deben tener un estado determinado. Además, se debe tener en el bus de datos <D7:D0> el código que indique un carácter para mostrar en la pantalla o una instrucción de control. En la figura 5.20 se muestra el diagrama de tiempos que se debe cumplir para manejar el módulo.

El LCD responde a un conjunto especial de instrucciones, enviadas por un sistema de control o un, según la operación que se requiera. En la figura 5.21 se muestran las instrucciones del módulo.

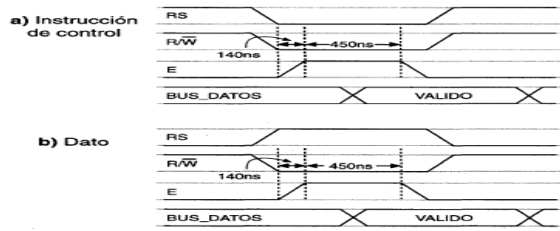


Figura 5.20 Diagrama de tiempo para leer o escribir en el LCD

INSTRUCCIONES	Señal de control		DATO/DIRECCION								DESCRIPCION
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
Borrar pantalla	0	0	0	0	0	0	0	0	0	1	Copia todo display y sitúa el cursor a la posición de inicio (Inicio de línea).
Cursor a casa	0	0	0	0	0	0	0	0	1	-	Mueve el cursor a la posición de inicio (Inicio de línea). También sitúa el display, desplazándolo a la posición original. Los contenidos de la RAM (D0) permanecen sin cambios.
Seleccionar modo	0	0	0	0	0	0	0	1	UD	0	Configura la dirección de movimiento del cursor y si se desplaza o no el display. Esta operación se realiza durante operaciones de lectura y escritura.
Encender/apagar pantalla	0	0	0	0	0	0	1	D	C	B	Configura el estado ON/OFF de todo el display (D), el cursor (C) y el parpadeo del cursor en la pantalla (B).
Desplazar Cursor/Pantalla	0	0	0	0	0	1	S/C	RL	-	-	Mueve el cursor y desplaza el display sin cambiar los contenidos de la RAM (D0).
Affix función	0	0	0	0	1	DL	N	F	-	-	Configura el tamaño de la interfaz (DL), el número de líneas del display (N) y la línea del cursor (F). Mod0 (línea 1), 2 (línea 2).
CG RAM	0	0	0	1	Dirección generador de RAM					Ajusta la dirección del generador de caracteres. Si dato CG RAM es erróneo y recibida después de ese punto.	
DD RAM	0	0	1	Direcciones de datos RAM					Ajusta la dirección de la RAM (D0). La dirección se normaliza y emitida después de este punto.		
Bandera de ocupado	0	0	BF	AC					Lectura de la bandera Busy Flag, indicando que operaciones internas son realizadas, y lectura de los contenidos del contador de direcciones.		
Escritura CG RAM/DD RAM	1	0	Escritura de Datos					Escritura de datos en la RAM (D0) o en la RAM (D).			
Lectura CG RAM/DD RAM	1	1	Lectura de Datos					Lectura de datos desde la RAM (D0) o la RAM (D).			

Significado de las abreviaturas	
I/D	= 1 Incrementa = 0 Decrementa
S	= 1 Desplaza el mensaje en la pantalla = 0 Mensaje fijo en la pantalla
D	= 1 Encender (activar) la pantalla = 0 Apagar la pantalla (desactivar)
C	= 1 Activar cursor = 0 Desactivar cursor
B	= 1 Parpadea caracter señalado por el cursor = 0 No parpadea el caracter
S/C	= 1 Desplaza pantalla = 0 Mueve cursor
RL	= 1 Desplazamiento a la derecha = 0 Desplazamiento a izquierda
DL	= 1 Datos de ocho bits = 0 Datos de cuatro bits
BF	= 1 Durante operación interna del módulo = 0 Finalizada la operación interna

Figura 5.21 Instrucciones del módulo LCD

Los caracteres que se envían a exhibir se almacenan en la memoria RAM LCD. Existen posiciones de memoria RAM, cuyos datos son visibles en la pantalla y otras que no son visibles, estas últimas se pueden utilizar para guardar caracteres que luego se desplazan hacia la parte visible. En la figura 5.22 se muestran las direcciones de memoria visibles y no visibles, que conforman las dos líneas de caracteres del LCD.

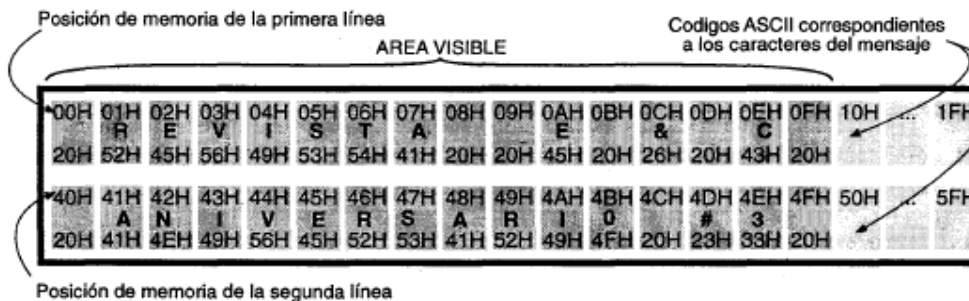


Figura 5.22 Mapa del memoria del LCD

Es importante mencionar que sólo se pueden mostrar caracteres ASCII de 7 bits, por lo tanto algunos caracteres especiales no se pueden ver, por lo que se debe tener a una tabla de los caracteres ASCII para conocer los datos que son prohibidos. Por otra lado, se tiene la opción de crear caracteres especiales (creados por el programador) y almacenarlos en la memoria RAM que posee el LCD.

La simulación del módulo LCD con Proteus se muestra en la figura 5.23.



Figura 5.23 Simulación del Módulo LCD con Proteus.

El circuito capturado del módulo LCD, el Artwork y la vista en 3D se muestran en la figura 5.24.

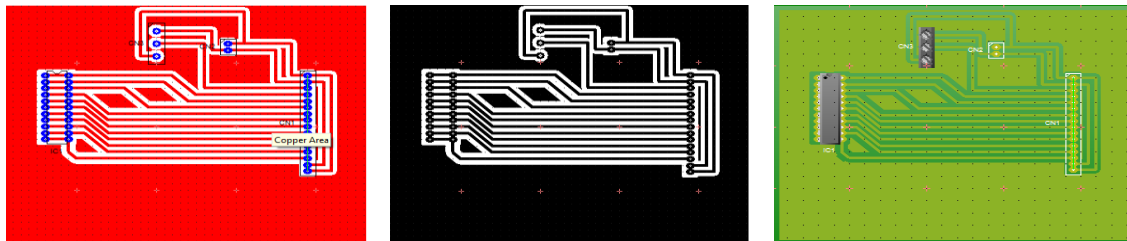


Figura 5.24 Captura del circuito, Artwork y vista en 3D del módulo LCD.

Después de que se grabó el Artwork a la placa, se perforó, se insertaron los componentes y soldaron, la tarjeta final del módulo LCD se muestra en la figura 5.25.



Figura 5.25 Tarjeta final del módulo LCD.

5.5 MÓDULO TECLADO LÍNEAL

Este módulo tiene la función de proporcionar niveles lógicos para circuitos Digitales, microcontroladores u otros. En la figura 5.26 se muestra el circuito del módulo teclado lineal.

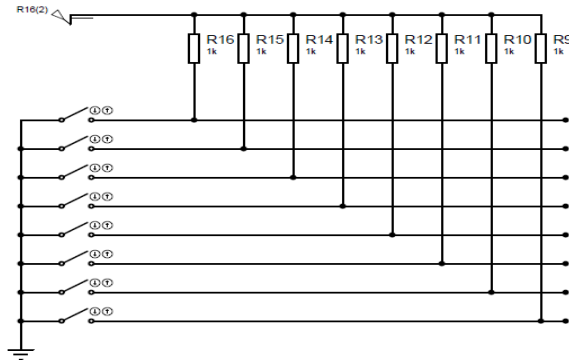


Figura 5.26 Circuito del módulo teclado lineal

El circuito requiere 5Vcc para funcionar. Este voltaje alimenta 8 arreglos seriales y cada arreglo serial está formado por 1 resistencia y 1 switch. La resistencia está conectada por una terminal al voltaje de 5Vcc y la otra terminal a una terminal del switch. La otra terminal del switch se conecta a tierra. La salida se toma de la unión resistencia switch.

Cuando el switch está abierto, en la unión se presenta un nivel de voltaje de 5Vcc (1 lógico) con una corriente de 5mA ($I = V/R = 5V_{cc} / 1000\Omega$), que es una magnitud más que suficiente para proporcionar como entrada a un circuito digital, microcontrolador u otro. Cuando el switch se cierra, en la unión resistencia switch se presenta 0Vcc (0 lógico).

La simulación del módulo teclado lineal realizada con Proteus se muestra en la figura 5.27, para ver que se suministran niveles lógicos altos y bajos, se utilizó el módulo led, y un circuito digital con inversores. Así que cuando un switch se cierra, el módulo teclado lineal envía un 0 lógico, pero invertido por los inversores, a la salida se tienen niveles lógicos 1.

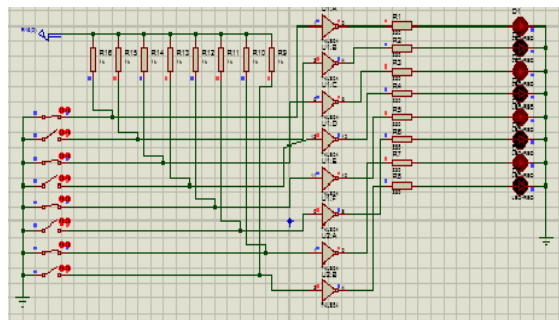


Figura 5.27 Simulación del módulo Teclado lineal con Proteus

El circuito capturado del módulo teclado lineal, el Artwork y la vista en 3D se muestran en la figura 5.28.

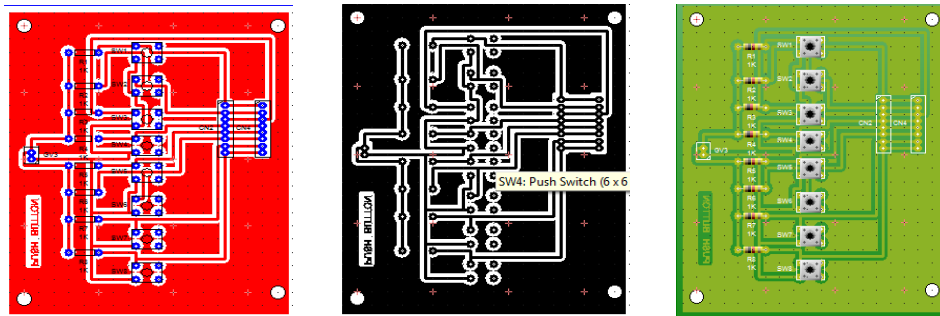


Figura 5.28 Captura del circuito, Artwork y vista en 3D del módulo teclado lineal.

Después de que se grabó el Artwork a la placa, se perforó, se insertaron los componentes y soldaron, se obtuvo la tarjeta final del módulo teclado lineal y se muestra en la figura 5.29.

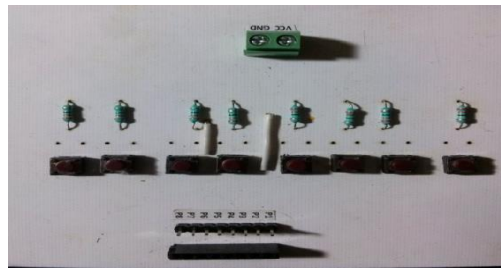


Figura 5.29 Tarjeta del módulo teclado lineal

5.6 MÓDULO TECLADO 4x4

Este módulo tiene la función de proporcionar hasta 16 niveles lógicos, a partir de un arreglo matricial de 4 filas x 4 columnas, para proporcionarlos a circuitos Digitales, microcontroladores u otros. En la intersección entre fila y columna se pone un push button. En la figura 5.30 se muestra el circuito del módulo teclado matricial.

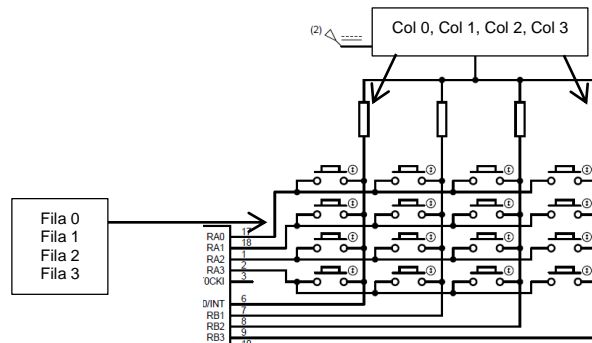


Figura 5.30 Circuito del módulo teclado matricial

Del circuito, las resistencias de 2.7K conectadas a las columnas permiten se tenga un nivel alto.

El manejo de este teclado está sujeto al método de barrido por filas. Las filas y columnas están numeradas, para las filas como fila 0, fila 1, fila 2, fila 3, y las columnas como Col 0, Col 1, Col 2 y Col 3. El método consiste en enviar por las filas un cero (0) y tres unos (111), pero el cero va cambiando de posición o barriendo fila por fila, es decir, primero se envía 0111, luego 1011, luego 1101 y finalmente 1110.

Si se envía 0111, y se oprime una tecla y esta corresponde a la columna 0, se podrá leer de las columnas un valor 0111, 1011, 1101, o 1110. Con esto se detecta la posición de la tecla oprimida. Es decir, una tecla oprimida tendrá una posición de (0111, 0111). Para cualquier tecla oprimida, habrá en los valores de filas un 0 y tres 1's, lo mismo en las columnas. Todos 1's en filas o columnas, o dos 0's en filas o columnas o más, serán combinaciones no válidas.

Si no se detecta oprimida con la combinación filas 0111, hay que continuar con el barrido de fila, es decir continuar sacando los datos 1011, 1101 y 1110. Hasta detectar posición de la tecla oprimida. Después de detectarse la tecla oprimida hay que asociarle un código como 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F o algún otro código.

El manejo del teclado matricial es complejo, por lo cual, se recomienda utilizarlo con microcontroladores.

La simulación del módulo teclado matricial con Proteus se muestra en la figura 5.31, Para esta simulación se utilizó el módulo display.

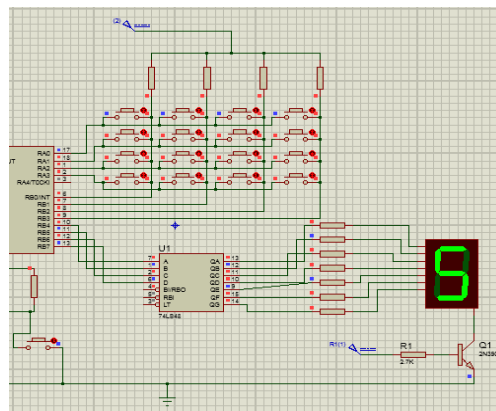


Figura 5.31 Simulación del Módulo Teclado Matricial

El circuito capturado del módulo teclado matricial, el Artwork y la vista en 3D se muestran en la figura 5.32

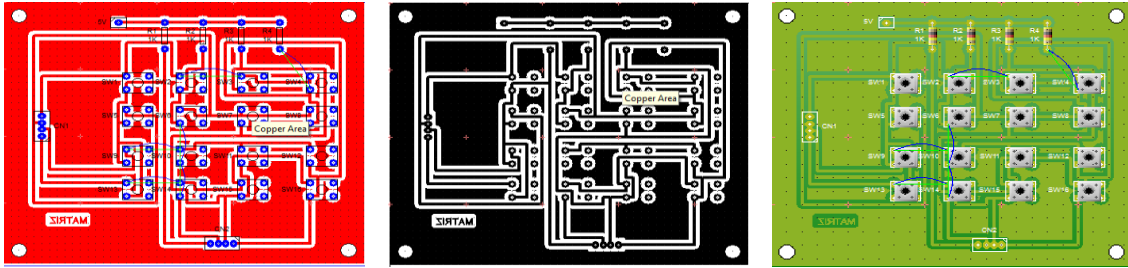


Figura 5.32 Captura del circuito, Artwork y vista en 3D del módulo teclado matricial.

Después de que se grabó el Artwork a la placa, se perforó, se insertaron los componentes y soldaron, se obtuvo la tarjeta final del módulo teclado lineal y se muestra en la figura 5.33.



Figura 5.33 Tarjeta del módulo teclado Matricial

5.7 MÓDULO TECLADO LINEAL LDR

Este módulo tiene la función de proporcionar 8 niveles lógicos, a partir de un arreglo lineal de 8 fotorresistencias. Los niveles lógicos se pueden utilizar en circuitos Digitales, microcontroladores u otros. La idea de este módulo es mostrarle a maestros y alumnos que otras tecnologías se pueden ajustar a las aplicaciones de este trabajo y, además de que es tecnología que se utiliza en áreas nuevas como la domótica, la inmótica, por mencionar algunas.

En la figura 5.34 se muestra el circuito del módulo teclado lineal óptico.

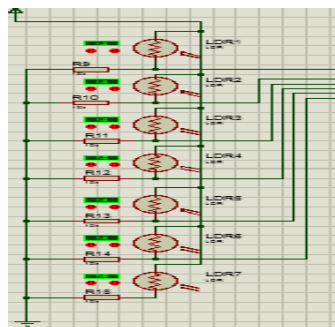


Figura 5.34 Circuito del módulo teclado lineal óptico

Del circuito, se tiene un arreglo serial entre la fotorresistencia (LDR) y resistencia. De la unión del LDR y fotorresistencia se deriva la salida lógica.

El LDR (*Light Dependent Resistor*) o resistencia dependiente de la luz o también fotocélula, o fotorresistencia, es una resistencia que varía su resistencia en función de la luz que incide sobre su superficie. Cuando mayor sea la intensidad de la luz que incide en la superficie del LDR menor será su resistencia y cuanto menos luz incida mayor será su resistencia, en la figura 5.35 se muestra el símbolo, imagen y gráfica de un LDR.

Los materiales fotosensibles más utilizados para la fabricación de las resistencias LDR son, el sulfuro de talio, el sulfuro de cadmio, el sulfuro de plomo, y el seleniuro de cadmio.

Cuando la LDR no está expuesta a radiaciones luminosas los electrones están firmemente unidos en los átomos que la conforman pero cuando sobre ella inciden radiaciones luminosas esta energía libera electrones con lo cual el material se hace más conductor, y de esta manera disminuye su resistencia.

Las resistencias LDR solamente reducen su resistencia con una radiación luminosa situada dentro de una determinada banda de longitudes de onda. Las construidas con sulfuro de cadmio son sensibles a todas las radiaciones luminosas visibles, las construidas con sulfuro de plomo solamente son sensibles a las radiaciones infrarrojas.

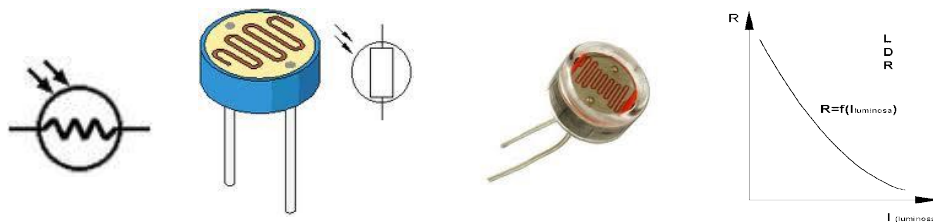


Figura 5.35 Símbolo, imagen y gráfica de un LDR

La resistencia en la obscuridad de un LDR está cercana a MegaOhms ($M\Omega$) y expuestas a la luz se acerca a los 100Ω . El tiempo de respuesta típico de un LDR es del orden de una décima de segundo.

El LDR por estos valores extremos se utiliza como interruptor para encendido y apagado de alumbrado, en alarmas, cámaras fotográficas, medidores de luz, control de máquinas y procesos de conteo, detección de objetos, etc.

En nuestro circuito funciona muy semejante al teclado lineal. Es decir si se emite luz al LDR se genera un valor lógico alto (1), pero si no hay suficiente luz, se genera un valor lógico bajo (0).

La simulación del módulo teclado lineal LDR con Proteus se muestra en la figura 5.36, para esta simulación se utilizó el Módulo leds.

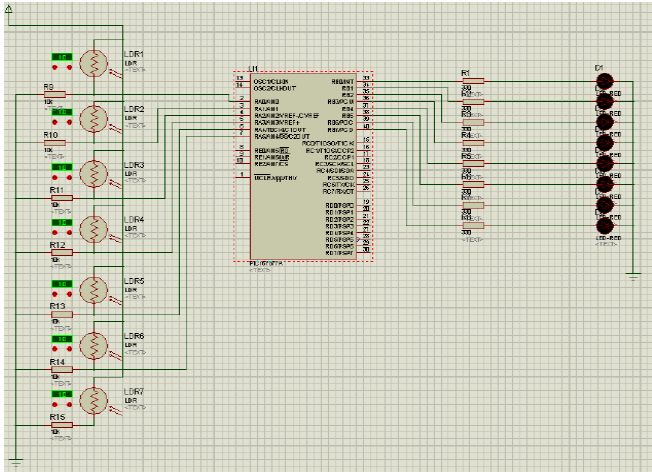


Figura 5.36 Simulación del Módulo Teclado Lineal LDR

El circuito capturado del módulo teclado lineal LDR, el Artwork y la vista en 3D se muestran en la figura 5.37

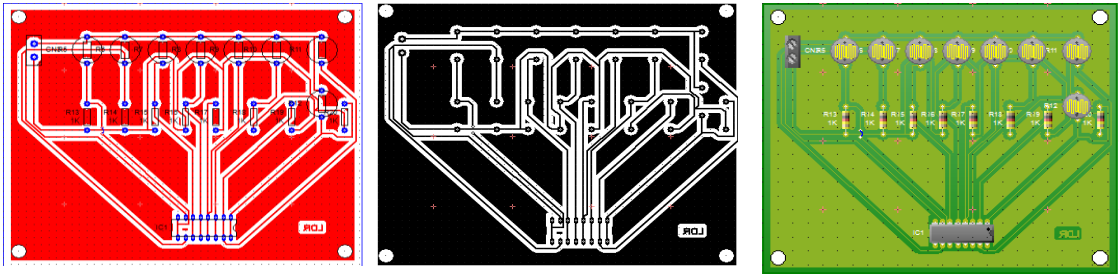


Figura 5.37 Captura del circuito, Artwork y vista en 3D del módulo teclado LDR.

Después de que se grabó el Artwork a la placa, se perforó, se insertaron los componentes y soldaron, se obtuvo la tarjeta final del módulo teclado lineal y se muestra en la figura 5.38.



Figura 5.38 Tarjeta del Módulo teclado LDR

5.8 MÓDULO BLUETOOTH

EL módulo bluetooth tiene la función de convertir la señal eléctrica en señal electromagnética, para comunicar entidades punto a punto.

Los módulos más utilizados para comunicación con bluetooth son el HC-05 y HC-06 y se conectan a microcontroladores. Estos dispositivos tienen la característica de que se pueden insertar en un protoboard o se puede cablear directamente a cualquier microcontrolador, ver figura 5.39.

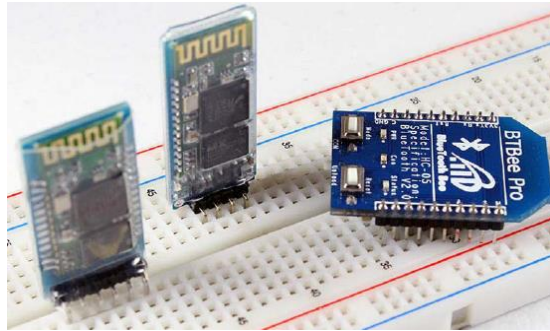


Figura 5.39 Módulos Bluetooth

De la figura, en la parte izquierda están los módulos HC-05 y HC-06 en formato amigable para protoboard y un módulo HC-05 en formato compatible con zócalos para Xbee (módulo bluetooth bee).

La tecnología Bluetooth trabaja con el modelo maestro esclavo y tecnología de comunicación punto a punto. Un maestro puede conectarse a un esclavo o hasta 7 esclavos, ver figura 5.40.

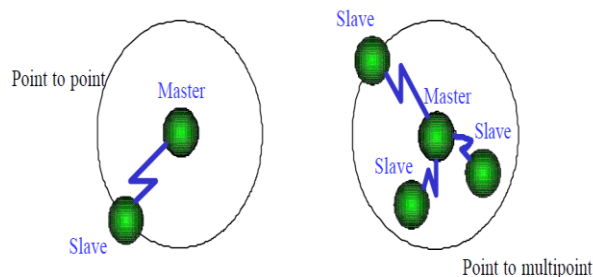


Figura 5.40 Conexión maestro a esclavo o punto a punto.

El módulo Bluetooth *HC-05* es el que ofrece una mejor relación de características, ya que es un módulo maestro-esclavo, es decir, que además de recibir conexiones desde una PC o tablet, también es capaz de generar conexiones hacia otros dispositivos bluetooth. Esto permite, por ejemplo, conectar dos módulos de bluetooth y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos, que es la parte didáctica de este trabajo.

El *HC-05* tiene el modo de comandos AT, el cual, debe de activarse mediante un valor alto en el PIN34 mientras se enciende (o se resetea) el módulo. En las versiones para protoboard este pin viene marcado como "Key". Una vez que se está en el modo de comandos AT, se puede configurar el módulo bluetooth y cambiar parámetros como el nombre del dispositivo, password, el modo maestro/esclavo, etc.

Para comunicarnos con el módulo bluetooth y configurarlo, es necesario tener acceso al módulo mediante una interfaz serial. Se puede usar un Microcontrolador, Arduino o PIC, con un par de cables, aprovechando el puente USB-Serial del Arduino, o un MAX3232 en el puerto serie de la PC.

El *HC-06* tiene un firmware distinto con respecto al HC-05 y también un funcionamiento distinto en cuanto a su modo de configuración. Para poder configurar el HC-06 es necesario que este no este emparejado ni siendo usado por ningún dispositivo. De igual forma que el HC-05 es necesario conectarlo a la PC y usar un programa de terminal para darle instrucciones de configuración (Comandos AT), aunque también se puede escribir un programa al Microcontrolador Arduino u otro, para configurarlo.

Para conectarlo con la PC se utiliza un adaptador USB serial (arduino como adaptador) como se muestra en la figura 5.41.

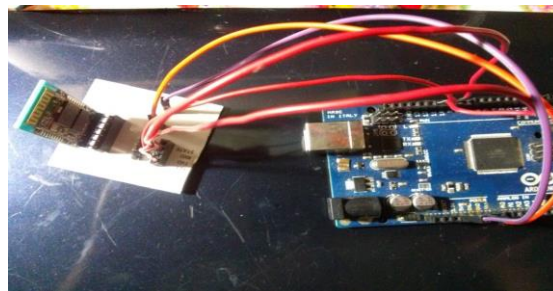


Figura 5.41 Microcontrolador Arduino y módulo Bluetooth para configuración

El módulo HC-06 acepta un set básico de comandos, que permite pocas configuraciones, pero sin duda satisface las necesidades de aplicación que se buscan.

Los comandos que soporta son:

- Prueba de funcionamiento:

Envíar: AT

Recibe: OK

- Configurar el Baud rate:

Envíar: AT+BAUD<Numero>

El parámetro número es un caracter hexadecimal de '1' a 'c' que corresponden a los siguientes velocidades o Baud Rates: 1=1200, 2=2400,

3=4800, 4=9600, 5=19200, 6=38400, 7=57600, 8=115200, 9=230400,
A=460800, B=921600, C=1382400.

Recibe: OK<baudrate>

- Configurar el Nombre de dispositivo Bluetooth:

Envíar: AT+NAME<Nombre>

Recibe: OKsetname

La conexión bluetooth con Arduino sólo requiere conectar como mínimo la alimentación y los pines de transmisión y recepción serial (TX y RX). Las conexiones son cruzadas, ver figura 5.42.

TX Bluetooth -> RX de Arduino

RX Bluetooth -> TX de Arduino.

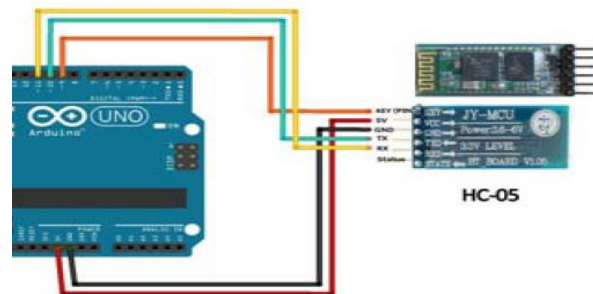


Figura 5.42 Conexión Arduino con Bluetooth HC-05

Para esta configuración se utiliza el hardware del UART (Transmisor Receptor Universal Asíncrono), por lo que la comunicación con la PC no será posible. Para poder utilizar la UART la PC se comunica mediante USB, y es necesario utilizar un software emulador del UART, de manera que los pines de comunicación con el módulo bluetooth queden en pines distintos.

El código para la comunicación a través del bluetooth que permite encender o apagar el led de la tarjeta arduino mediante el módulo bluetooth es:

```
SoftwareSerial BTSerial(10,11);
void setup() {
  pinMode(9,OUTPUT);
  digitalWrite(9,HIGH);
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  BTSerial.begin(38400);
}
void loop()
{
  if(BTSerial.available())
    Serial.write(BTSerial.read());
  if (Serial.available())
    BTSerial.write(Serial.read());
}
```

// Se establecen los parámetros AT, y
//define la velocidad de 38400 Baudios

// Se lee el dato serial y se guarda

El código en lenguaje C para un PIC PIC16F877, para realizar una vinculación entre dos módulos HC-05 es.

//Código Maestro

```
#include <16f877a.h>
#fuses NOWDT, XT, NOPUT, NOPROTECT, NODEBUG, // Se configuran fusibles,
NOBROWNOUT, NOLVP, NOCPD, NOWRT //oscilador, y comunicación
#use delay(clock = 4000000) //serial a 9600 bps, con 8 bits
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8 ) //de dato.
void main()
{
char x;
while(true)
{
if(input(pin_a0)==1) // Se lee dato por pin A0 si es 1
{
x=1; //Se transmite x = 1
putc(x);
delay_ms(500);
}
}
}
```

//Código Esclavo

```
#include <16f877a.h>
#fuses NOWDT, XT, NOPUT, NOPROTECT, NODEBUG, // Se configuran fusibles,
NOBROWNOUT, NOLVP, NOCPD, NOWRT //oscilador, y comunicación
#use delay(clock = 4000000) //serial a 9600 bps, con 8 bits
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8) //de dato.
int dato;
#int_rda //activar recepcion
void rda_isr()
{
dato=getc(); // Se lee dato
}
void main() //Se transmite x = 1
{
enable_interrupts(int_rda); //Se habilitan interrupciones
enable_interrupts(global);
while(true)
{
if(dato==1) //Si dato leído es 1
{
Output_high(pin_b0); //Encender led por 500
delay_ms(500); //milisegundos y luego apagarlo
Output_low(pin_b0);
}
}
}
```

La simulación del módulo Bluetooth con Proteus se muestra en la figura 5.43.



Figura 5.43 Simulación Bluetooth M/S con 2 PICs 16F877

El circuito capturado del módulo Bluetooth, el Artwork y la vista en 3D se muestran en la figura 5.44

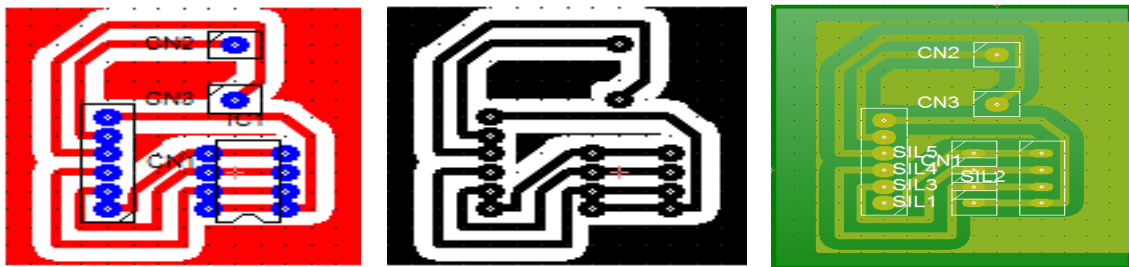


Figura 5.44 Captura del circuito, Artwork y vista en 3D del módulo Bluetooth.

Después de que se grabó el Artwork a la placa, se perforó, se insertaron los componentes y soldaron, se obtuvo la tarjeta final del módulo bluetooth y se muestra en la figura 5.45



Figura 5.45 Circuito impreso y tarjeta del módulo Bluetooth

En la figura 5.46 se muestra la prueba del módulo bluetooth. Parra ello se utilizaron dos módulos de microcontrolador, dos módulos de Bluetooth y el módulo de leds.

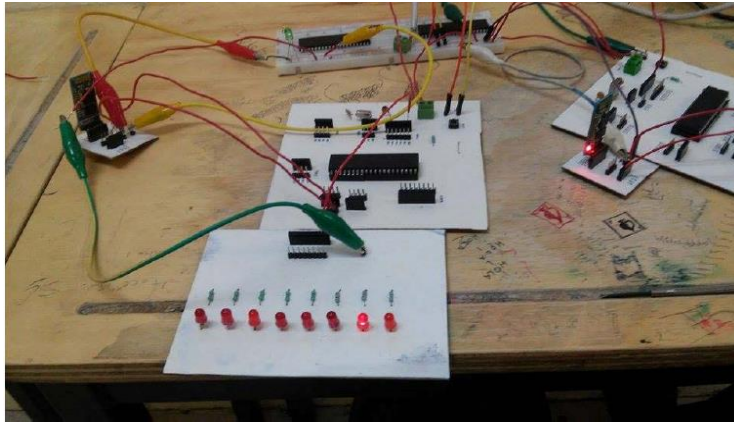


Figura 5.46 Prueba del módulo Bluetooth M/S para encender un led.

5.9 MÓDULO MICRO

Este módulo tiene la función de proporcionar las señales del microcontrolador 16F877, el cual, se puede utilizar para realizar las prácticas de microcontroladores, o para establecer la comunicación bluetooth maestro esclavo.

Este microcontrolador tiene las siguientes características:

- Tiene 40 pines
- Memoria de programa : FLASH, 8 K de instrucciones de 14 bits c/u.
- Memoria de datos : 368 bytes RAM, 256 bytes EEPROM.
- Pila (Stack) : 8 niveles (14 bits).
- Fuentes de interrupción : 13
- Instrucciones : 35
- Encapsulado : DIP de 40 pines.
- Frecuencia oscilador : 20 MHz (máxima)
- Temporizadores/Contadores:
 - 1 de 8 bits (Timer 0)
 - 1 de 16 bits (Timer 1)
 - 1 de 8 bits (Timer 2) con pre y post escalador.
 - Un perro guardián (WDT)
- Líneas de E/S :
 - 6 del puerto A
 - 8 del puerto B
 - 8 del puerto C
 - 8 del puerto D
 - 3 del puerto E
 - Además de 8 entradas análogas.
- Dos módulos de Captura, Comparación y PWM:
 - Captura: 16 bits. Resolución máx. = 12.5 nseg.
 - Comparación: 16 bits. Resolución máx. = 200 nseg.
 - PWM: Resolución máx. = 10 bits.

- Convertidor Análogo/Digital de 10 bits multicanal (8 canales de entrada).
- Puerto serial síncrono (SSP) con bus SPI (modo maestro) y bus I²C (maestro/esclavo).
- USART (Universal Synchronous Asynchronous Receiver Transmitter) con dirección de detección de 9 bits.
- Corriente máxima absorbida/suministrada (sink/source) por línea (pin): 25 mA
- Oscilador : Soporta 4 configuraciones diferentes: XT, RC, HS, LP.
- Tecnología de Fabricación: CMOS
- Voltaje de alimentación: 3.0 a 5.5 V DC

La configuración de pines y la distribución por puertos del PIC 16F6877 se muestra en la figura 5.47.

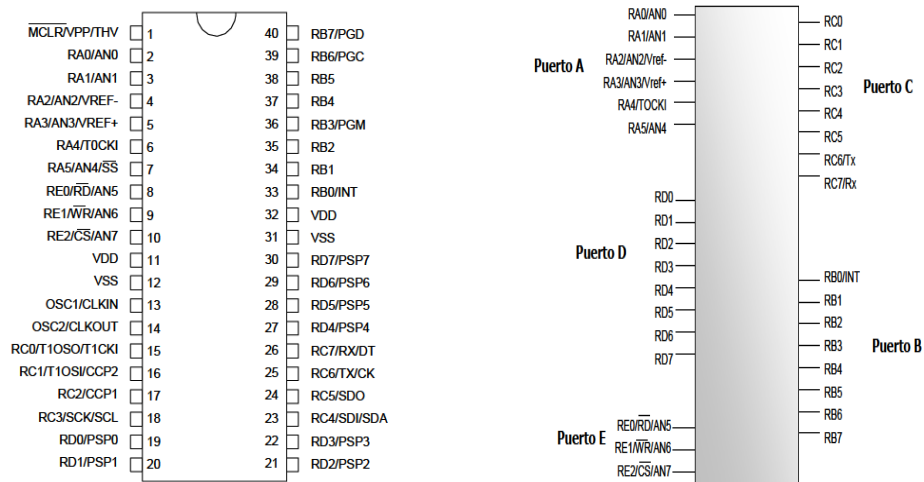


Figura 5.47. PIC 16F6877 Configuración de pines y distribución por puertos

El circuito capturado del módulo Microcontrolador, el Artwork y la vista en 3D se muestran en la figura 5.48

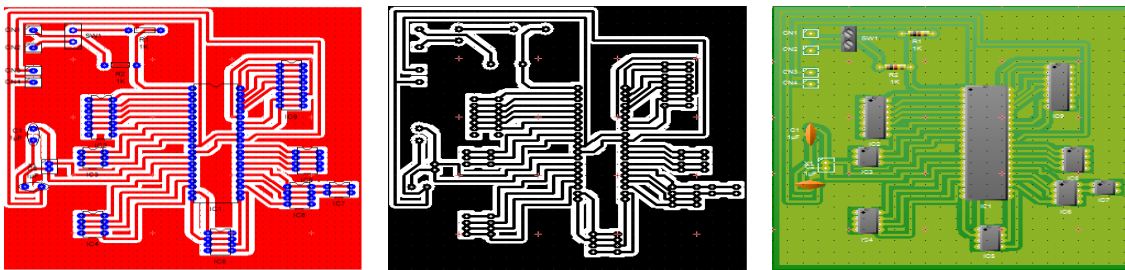


Figura 5.48 Captura del circuito, Artwork y vista en 3D del módulo Microcontrolador.

Después de que se grabó el Artwork a la placa, se perforó, se insertaron los componentes y soldaron, se obtuvo la tarjeta final del módulo Microcontrolador que se muestra en la figura 5.49.

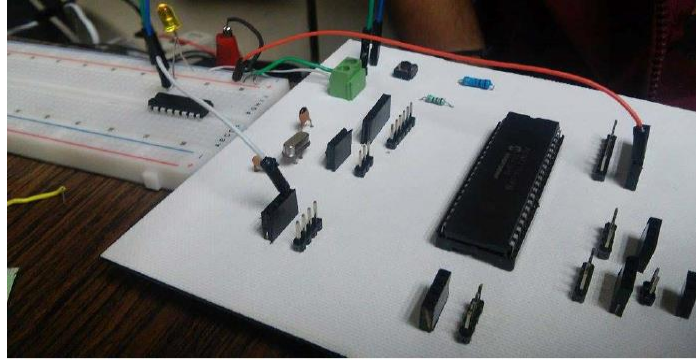


Figura 5.49 Tarjeta del módulo Microcontrolador.

CAPÍTULO 6

APLICACIONES

En este capítulo se presentan aplicaciones de las áreas de Sistemas Digitales y Microcontroladores utilizando los módulos del KIP.

6.1 USO DE LOS MÓDULOS

Los módulos se pueden utilizar de forma individual o se puede utilizar una mampara donde están todos distribuidos de acuerdo a la arquitectura propuesta del mismo.

Trabajar los módulos de manera individual, significa manipular los módulos a nuestra conveniencia, sólo se manejan los que se utilizan en las prácticas. Trabajar los módulos integrados en una mampara, significa que los módulos están fijos y distribuidos de tal manera que no habrá flexibilidad para manipularlos.

Para las prácticas que se presentan, los módulos se trabajarán de manera individual.

Conexiones

Todos los módulos disponen de trenes de pines machos y hembras, conectados en paralelo, para las conexiones con otros módulos o circuitos. Si el usuario tiene un bus plano con pines macho, podrá utilizar, de los módulos, los trenes de pines hembra para conectar, y viceversa, también puede utilizar cable del Np. 22, o caimanas. Los pines están etiquetados, ver figura 6.1.

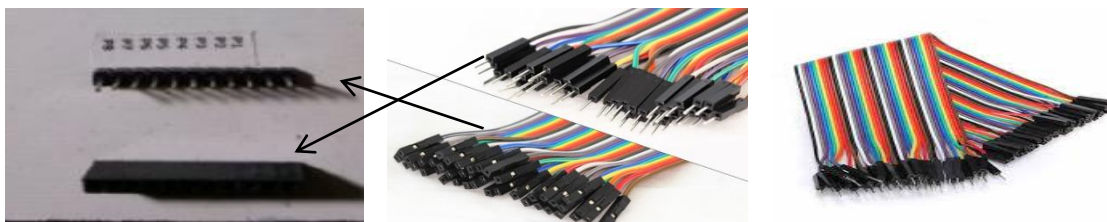


Figura 6.1 Conectores Macho y Hembra de losmódulos

Todas las tarjetas tiene un Clemax, con sus respectivas etiquetas, para conectar la alimentación de 5Vcc y 0Vcc. Ver figura 6.2.



Figura 6.2 Conexión de alimentación en las tarjetas a través del Clemax

Si utiliza las tarjetas de los módulos, de manera individual, no los ponga sobre metal, pedazos de alambres pelados, pinzas u otro medio que a la hora de conectarlos a la alimentación, porque pudieran ocasionar un corto.

6.2 APLICACIONES CON SISTEMAS DIGITALES (SD).

A continuación se presentan un conjunto de prácticas de sistemas digitales que utilizan los módulos.

PRÁCTICA 1 SD

CIRCUITOS DIGITALES DE MÚLTIPLES ENTRADAS Y UNA O MÚLTIPLES SALIDAS

1.1 OBJETIVO

Que el alumno conozca y aplique técnicas para simplificar funciones booleanas, de varias entradas y una o varias salidas, capture y simule dichas funciones, las implemente con compuertas y use los módulos del KIP.

1.2 INTRODUCCIÓN

Los circuitos digitales pueden resolver problemas de diversa complejidad, desde una simple comparación de datos hasta implementar algoritmos inteligentes, redes neuronales, procesamiento de imágenes, etc.

En ésta práctica se ataca el problema de diseñar y desarrollar circuitos digitales de varias entradas y una o varias salidas.

Esta práctica se guía para resolver un problema, se parte de la especificación funcional del circuito, se diseña y luego se arma y se utilizan los módulos. La solución requiere primero plantear la tabla de verdad, en seguida se obtiene de la tabla la función booleana, se simplifica, se simula e implementa.

Con esta práctica el alumno adquirirá habilidades para diseñar circuitos digitales con la menor cantidad de compuertas lógicas.

1.3 MATERIAL Y EQUIPO NECESARIO

Hardware:

- 1 Fuente de 5Vcc o equivalente.
- 3 IC 74LS04
- 5 IC 74LS08
- 5 IC 74LS32
- 1 Módulo Teclado Lineal
- 1 Módulo leds

Software

1.4 METODOLOGÍA

En teoría se deben conocer las propiedades del algebra booleana; tablas de verdad, simplificación de funciones aplicando las propiedades; compuertas lógicas TTL y uso de Proteus.

1.4.1 CIRCUITOS DE MÚLTIPLES ENTRADAS UNA SALIDA

A continuación se presenta una serie de ejercicios prácticos para que experimente los circuitos digitales de varias entradas y una salida.

1.4.1.1 Números nones.

Para ello realice los siguientes pasos

i.- Se diseñará e implementará un circuito digital que detecte si un número digital de tres bits es impar.

Para ello en la tabla 6.1 se muestran los valores que corresponden al dato (ABC) y la salida (F). F es 1 si es que el número dado por la entrada es impar en otro caso es 0.

Tabla 6.1 Tabla de verdad para detectar números impares o nones.

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Analizando de la tabla en donde $F = 1$ se tiene:

$$F = A'B'C + A'BC + AB'C + ABC$$

Simplificando la función booleana utilizando las propiedades del algebra booleana se obtiene a F, ecuación 6.1.

$F = A'B'C + A'BC + AB'C + ABC$ $F = (A'B'C + A'BC) + (AB'C + ABC)$ $F = A'C(B' + B) + AC(B' + B)$ $F = A'C + AC$ $F = C(A' + A)$ $F = C$	<p><i>función original</i></p> <p><i>agrupar y factor común</i></p> <p><i>Propiedad: $B+B'=1$</i></p> <p><i>factor común</i></p> <p><i>Propiedad: $A+A'=1$</i></p> <p style="text-align: right;">ec. 6.1</p>
---	--

Donde el circuito final se muestra en la figura 6.3. La salida F es C.

ii.- Capture y simule el circuito detector de números impares en Proteus

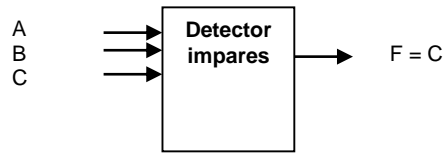


Figura 6.3 Circuito digital que detecta si un número de 3 bits es impar.

iii.- Arme el circuito detector de números impares en un protoboard y conecte a la entrada del circuito el módulo teclado lineal y a la salida el módulo de leds, y realice pruebas. En la figura 6.4 se muestran los módulos del circuito detector de números nones.

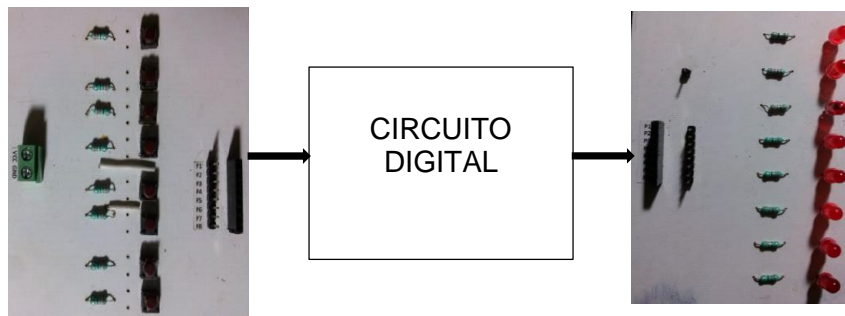


Figura 6.4 Circuito digital detector de números nones con módulos del KIP.

1.4.1.2 Números primos

i.- Diseñe e implemente un circuito digital que detecte si un número digital de cuatro bits es un número primo.

Para ello realice lo siguiente:

- Primero establezca la tabla de verdad.
- Segundo obtenga la función F y simplifíquela.
- Tercero dibuje el circuito.

ii.- Capture y simule el circuito detector de números primos en Proteus

iii.- Arme el circuito detector de números primos en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas

1.4.1.3 Disparo de una alarma.

i.- Considere que tiene cuatro sensores, y lo que se quiere es diseñar un circuito digital que dispare una alarma cuando los sensores, en sus combinaciones den un número par o un múltiplo de 3.

Para ello realice lo siguiente:

- Primero establezca la tabla de verdad.
- Segundo obtenga la función F y simplifíquela.
- Tercero dibuje el circuito.

ii.- Capture y simule el circuito que dispara una alarma en Proteus

ii.- Arme el circuito que dispara una alarma en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas

1.4.2 CIRCUITOS DE MÚLTIPLES ENTRADAS Y MÚLTIPLES SALIDAS

A continuación se presenta una serie de ejercicios prácticos que corresponden a circuitos digitales de varias entradas y varias salidas.

1.4.2.1 Cuadrado de un número

i.- Diseñe un circuito digital que calcule el cuadrado de un número de 3 bits. Esto significa que el circuito tiene 3 entradas y 6 salidas, tabla 6.2, la cual se pide al alumno llene.

6.2. Tabla de verdad del cuadrado de un número de 3 bits

D ₂	D ₁	D ₀	F ₅	F ₄	F ₃	F ₂	F ₁	F ₀
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

Obtenga las ecuaciones de cada una de las funciones y simplifíquelas.

F₅ =

F₄ =

F₃ =

F₂ =

F₁ =

F₀ =

ii.- Capture y simule el circuito que obtiene el cuadrado de un número en Proteus

iii.- Arme el circuito que obtiene el cuadrado de un número en un protoboard y conéctele los módulos de E/S del KIP y realice pruebas

1.4.3 PROPUESTAS DE NUEVAS PRÁCTICAS.

En cada una de las siguientes propuestas se da la especificación y debe realizar los pasos dados en las prácticas anteriores, para resolver cada caso.

- i.- Diseñe un circuito que permita comparar dos datos de 2 bits cada uno, indicando si el primero es mayor, menor o igual al segundo.
- ii.- Diseñe un circuito digital que genere código hexadecimal y se vea en un display.
- iii.- Diseñe un circuito digital que al leer un dato del 1 al 9 genere equivalente binario.

1.5 PREGUNTAS

i.- ¿Qué otros ejemplos de circuitos digitales de múltiples entradas y múltiples salidas daría?

ii.- ¿Puede diseñarse un procesador con esta metodología y cómo se imagina que sería el trabajo?

iii.- ¿Dónde aplicaría circuitos de múltiples entradas y una salida?

1.6 REPORTE DEL ALUMNO (Resultados)

El indicado por el profesor

1.7 BIBLIOGRAFÍA PRELIMINAR

- Tocci Ronald, Sistemas Digitales, Octava edición, Prentice – Hall, México 2006.
- John F. Wakerly, Diseño Digital Principios y Prácticas, Tercera edición, Pearson, 2001.
- Morris Mano, M., Diseño Digital, Tercera edición, Pearson Educación, 2003.

PRÁCTICA 2 SD

SUMADORES Y RESTADORES

2.1 OBJETIVO

Que el alumno conozca, diseñe y simule sumadores completos de datos de n bits a partir de sumadores completos de 1 bit, además de implementarlas con compuertas.

2.2 INTRODUCCIÓN

Muchos circuitos integrados (ICs) tienen una estructura y funcionamiento complejo, sin embargo, dicha estructura tiene módulos básicos sencillos que se iteran o repiten como los sumadores, restadores, comparadores entre otros. En ésta práctica se pretende abordar el diseño de los sumadores y restadores completos de dos datos de 4 bits cada uno con carry (borrow para la resta) de entra y carry (borrow) de salida a partir de sumadores o restadores completos de dos datos de 1 bit cada uno con carry (borrow) de entrada y carry (borrow) de salida.

Para trabajar con módulos MSI (Medium Scale Integration) primeramente se resolverá el problema de diseñar módulos básicos para la suma o resta a partir de módulos SSI (Small Scale Integration). Después dicho módulo se iterará para formar el sumador completo de dos datos de 4 bits cada uno con carry de entrada y carry de salida y finalmente se utilizará un IC MSI que contenga al sumador y se realicen aplicaciones.

La ventaja de hacer el módulo básico del sumador es con la idea de que el alumno tenga la habilidad de diseñar sumadores o restadores de 16 bits o más. La desventaja de diseñar con módulo básico a la larga puede ser muy tedioso y grande el diseño.

Los puntos que a continuación se tratan son el material a utilizar, la metodología a seguir, las sugerencias didácticas, así como un reporte que el alumno debe realizar desde el punto de vista de resultados y la bibliografía preliminar que se ha consultado

2.3 MATERIAL Y EQUIPO NECESARIO

Hardware

- 1 Fuente 5Vcc
- 2 IC 74LS83
- 2 IC 74LS04
- 3 IC 74LS32
- 3 IC 74LS048
- 1 Módulo Teclado Lineal
- 1 Módulo leds
- 1 Módulo display condecoder

Software

Proteus

2.4 METODOLOGÍA

Para realizar esta práctica debe manejar mapas de karnaugh y conocer el funcionamiento de los sumadores y restadores.

2.4.1 Sumadores

2.4.1.1. Semi-sumador binario.

i.- La primero es diseñar un semi-sumador binario. Este circuito suma en binario dos datos de 1 bit cada uno (A_i , B_i) y da como salidas a la suma (S_i) y carry de salida (C_i). En la tabla 6.3 se proporcionan sus valores de verdad.

Tabla 6.3. Tabla del semisumador

A_i	B_i	C_i	S_i
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Las ecuaciones del semisumador, ecuaciones 6.1, se obtienen de la tabla anterior.

$$S_i = A_i B_i + A_i B_i' = A_i \oplus B_i$$

$$C_i = A_i B_i$$

ecs. 6.2

El circuito del semisumador se presenta en la figura 6.5

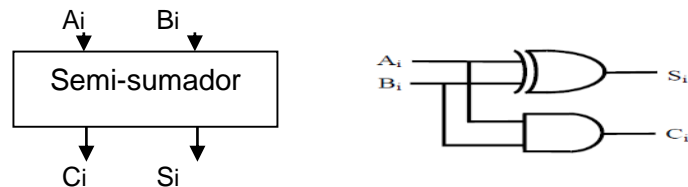


Figura 6.5 Diagrama a bloques y circuito de semi-sumador

ii.- En proteus simule el circuito semisumador.

iii.- Arme en un protoboard el circuito semisumador y conéctele los módulos E/S del KIP y realice pruebas.

2.4.1.2 Sumador completo de dos datos de 1 bit cada uno con carry de entrada y un carry de salida.

i.- Ahora se quiere diseñar un sumador completo binario de dos datos de 1 bit cada uno con carry de entrada y carry de salida. Al sumador completo se le puede considerar como el semi-sumador con carry de entrada.

Para ello realice lo siguiente:

Primero establezca la tabla de verdad, tabla 6.4.

Tabla 6.4.- Tabla del sumador completo

Ai	Bi	Ci	Ci+1	Si
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Segundo obtenga las funciones Si y Ci.

Ci =

Si =

Tercero dibuje el circuito del sumador completo con sus compuertas. En la figura 6.6 se presenta el diagrama a bloques del sumador completo.

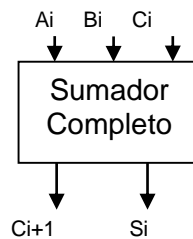


Figura 6.6 Diagrama a bloques del sumador completo

ii.- Capture y simule el circuito sumador completo en Proteus

iii.- Arme el circuito sumador completo en un protoboard y conéctele los módulos de E/S del KIP y realice pruebas.

2.4.1.3 Sumador completo de dos datos de 2 bits cada uno.

i.- Diseñe un sumador completo de 2 bits a partir de sumadores completos de 1 bit. Los sumadores completos de 1 bit se pueden conectar en cascada.

Para ello realice lo siguiente:

Establezca los valores del sumador completo de 2 bits en la tabla 6.5

Tabla 6.5 Tabla del sumador completo de dos datos de 2 bits cada uno.

A1	B1	A0	B0	C0	C2	S1	S0
0	0	0	0	0			
0	0	0	0	1			
0	0	0	1	0			
0	0	0	1	1			
0	0	1	0	0			
0	0	1	0	1			
0	0	1	1	0			
0	0	1	1	1			
0	1	0	0	0			
0	1	0	0	1			
0	1	0	1	0			
0	1	0	1	1			
0	1	1	0	0			
0	1	1	0	1			
0	1	1	1	0			
0	1	1	1	1			
1	0	0	0	0			
1	0	0	0	1			
1	0	0	1	0			
1	0	0	1	1			
1	0	1	0	0			
1	0	1	0	1			
1	0	1	1	0			
1	0	1	1	1			
1	1	0	0	0			
1	1	0	0	1			
1	1	0	1	0			
1	1	0	1	1			
1	1	1	0	0			
1	1	1	0	1			
1	1	1	1	0			
1	1	1	1	1			

Obtenga las ecuaciones para C_2 , S_1 y S_0 . No utilice mapas de Karnaugh sino por simple inspección.

$C_2 =$

$S_1 =$

$S_0 =$

Dibuje el circuito del sumador completo de dos datos, cuyo diagrama a bloques se muestra en la figura 6.7.

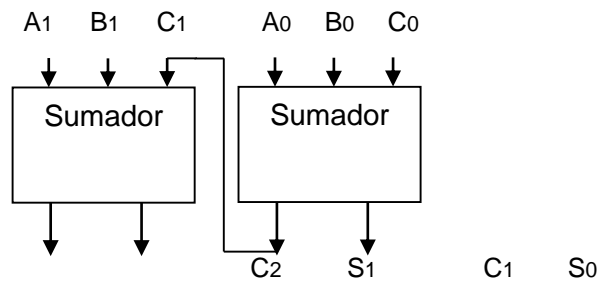


Figura 6.7 Diagrama a bloques del circuito sumador completo de dos datos de 2 bits.

ii.- Capture y simule el circuito sumador completo en Proteus

iii.- Arme el circuito sumador completo en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas.

2.4.1.4 Sumador en BCD

i.- Se quiere diseñar un sumador en BCD de dos datos de 4 bits cada uno a partir de 2 sumadores MSI (74LS83) y compuertas.

Con un primer sumador sume los dos datos A y B y su salida conéctala a una entrada del segundo sumador. Si detecta que la suma del primer sumador es mayor a 9, sume un 6 en la segunda entrada del segundo sumador, con lo cual hará un ajusta para obtener la suma en BCD. Utilice la tabla 6.6 y complétela para detectar un número mayor a 9.

Tabla 6.6 Tabla para detectar un número mayor a 9

S ₃ S ₂ S ₁ S ₀	F0
1010	
1011	
1100	
1101	
1110	
1111	

De su análisis encuentre la función para detectar el número mayor a 9.

F =

El diagrama a bloques de la figura 6.8 muestra cómo quedará el sumador en BCD utilizando dos sumadores completos 74LS83.

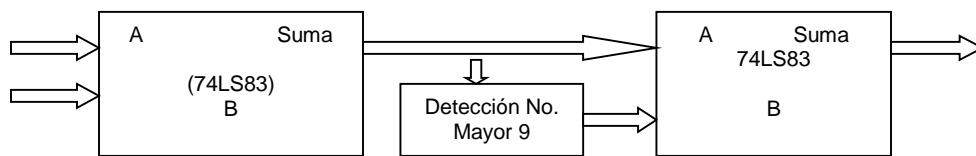


Figura 6.8. Diagrama a bloques del circuito sumador en BCD.

ii.- Con Proteus simule el circuito sumador en BCD.

iii.- Arme en un protoboard el circuito sumador en BCD y conéctele los módulos de E/S del KIP y realice pruebas.

2.4.2 Diferenciadores o restadores

2.4.2.1. Sustractor

i.- Un circuito digital diferenciador o restador es un circuito digital que resta del minuendo (A_i) un sustraendo (B_i) y puede haber acarreo denominado borrow (b_i). Las salidas se llaman diferencia (D) o resta y borrow de salida (b_{i+1}). Lo que se quiere hacer es diseñar e implementar.

En la tabla 6.7 se muestran los datos de entrada y salida del restador.

Tabla 6.7 Tabla del diferenciador

A_i	B_i	b_i	b_{i+1}	D_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

De la tabla se obtienen las ecuaciones 7.2, que son para obtener la resta D_i y el borrow b_{i+1} .

$$b_{i+1} = A_i'b_i + A_i'B_i + B_ib_i$$

$$D_i = A_i \oplus B_i \oplus b_i$$

ecs. 3

El circuito del restador o diferenciador se muestra en la figura 6.9.

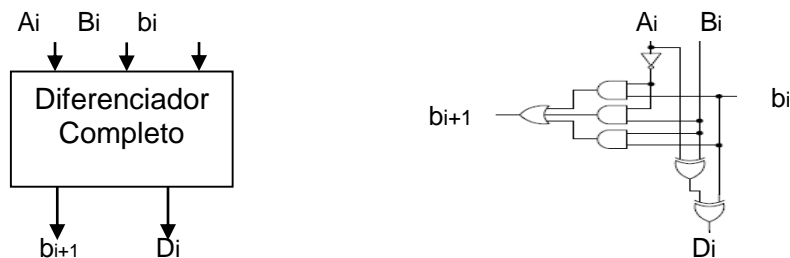


Figura 6.9 Diagrama a bloques y circuito del diferenciador completo

ii.- Con Proteus simule el circuito restador o diferenciador.

iii.- Arme en un protoboard el circuito restador o diferenciador, conéctele los módulos de E/S del KIP y realice pruebas.

2.4.2.2 Sumador Restador

Para desarrollar esta parte realice lo siguiente:

i.- Se quiere diseñar un sumador restador a partir de un IC sumador completo y que una señal si tiene valor 0 habilite la suma y si tiene valor 1 realice la resta.

En la tabla 6.8 se muestra ésta situación.

Tabla 6.8 Tabla del sumador restador

Señal s	Función
0	Suma
1	Resta

Para obtener las ecuaciones o circuito que habilite la suma o resta, se considera lo siguiente:

Si la señal s está en 0 hay una suma aritmética ($a + b$).

Si la señal s está en 1 hay una resta ($a - b$). Y para hacer la resta se trabaja con complemento a 2.

El complemento a 2 significa negar el sustraendo y sumarle 1, luego se suma con el minuendo y se obtiene la resta.

Con éste análisis, si se utiliza una señal s para habilitar la suma y la resta, se requiere de un circuito que cuando s esté en 0, el dato pase. Y cuando s esté en 1, se niegue al dato y se sume un 1.

Para ello se utilizarán compuertas XOR en el segundo dato b, exclusivas, que permite llevar a cabo lo descrito arriba.

Así, el dato a se conecta a la primera entrada del sumador 74LS83. Los bits del dato b se conecta a una entrada de 4 compuertas XOR y la salida de estas compuertas se conectan al segundo dato del sumador. La señal externa se conecta a las segundas entradas de las XORs y al carry de entrada del sumador.

El circuito Sumador restador, en bloques, se muestra en la figura 6.10.

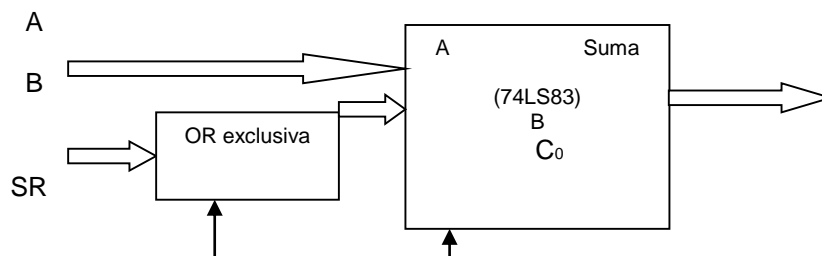


Figura 6.10 Diagrama a bloques del sumador restador

ii.- Con Proteus simule el circuito sumador restador.

iii.- Arme en un protoboard el circuito sumador restador conéctele los módulos de E/S del KIP y realice pruebas.

2.5. PREGUNTAS

i.- ¿De su punto de vista cuando se diseña con circuitos MSI y LSI?.

ii.- Escriba que otros circuitos MSI y LSI hay en el mercado

2.6 REPORTE DEL ELUMNO (Resultados)

El indicado por el profesor

2.7 BIBLIOGRAFÍA PRELIMINAR

- Frederick J. Hill. Gerald R. Peterson. Teoría de conmutación y diseño lógico. LIMUSA
- Ronald J. Tocci. Sistemas Digitales: Principios y aplicaciones. 5a Edición. Prentice Hall
- John F. Wakerly, Diseño Digital Principios y Prácticas, Tercera edición, Pearson, México 2001.
- Morris Mano, M., Diseño Digital, Tercera edición, Pearson Educación, México 2003

PRÁCTICA 3 SD

FLIP FLOPS

3.1 OBJETIVO

Que el alumno conozca y comprenda el funcionamiento de los flip flops, que utilice herramientas CAD para que los diseñe, capture y simule, además de implementarlos con compuertas.

3.2 INTRODUCCIÓN

Los circuitos secuenciales se conforman de lógica combinacional y elementos de memoria. Los circuitos combinacionales se han visto a lo largo de las prácticas anteriores. Los elementos de memoria son la parte fundamental de las máquinas secuenciales. La función de memorizar la realizan los Flip Flops (FF) y es lo que se tratará en esta práctica.

Un flip-flop (FF) es un circuito digital o elemento de memoria que puede estar en uno de dos estados binarios: 0 o 1, lo que significa que un FF está en estado 0 o estado 1. Hay 4 tipos básicos de FFs: SR, D, JK y T.

Se inicia trabajando al FF SR, a partir del cual se desprenderán los demás. Se presentará el problema, se diseñará y luego se comprobará.

En ésta práctica se guía al estudiante para que diseñe, capture, simule e implemente los cuatro tipos de Flip Flops básicos.

Los puntos que se tratarán en la presente práctica son: el material a utilizar para la práctica, la metodología a seguir, las sugerencias didácticas, así como un reporte que el alumno debe realizar desde el punto de vista de resultados y la bibliografía preliminar que se ha consultado

3.3 MATERIAL Y EQUIPO NECESARIO

Hardware

- 1 Fuente 5Vcc
- 2 IC 74LS00
- 2 IC 74LS02
- 1 IC 74LS04
- 1 IC Mux 2 a 1
- 1 IC 74L71
- 1 IC 74LS73
- IC 74LS74
- 1 Módulo Teclado Lineal
- 1 Módulo leds

Software

Proteus

3.4 METODOLOGIA

Para realizar esta práctica debe tener los conocimientos de los Flip-Flops SR, D, JT y T.

3.4.1 Flip FLOp (FF) y Latch (SR)

A.- FF Set Reset (FF SR) con compuertas NAND y NOR

i.- Diseñe un flip flop SR (FF SR) con compuertas NAND y con compuertas NOR. Un FF se caracteriza porque tiene dos entradas como el FF SR y el FF JK, o una entrada como el FF D y el FF T. Todos tienen dos salidas denominadas Q y Q'.

En las tablas 6.9a y 6.9b se muestran las tablas de los FF SR para compuertas NAND y compuertas NOR y en figura 6.10 se presentan los circuitos de los flip flops respectivos, junto con su símbolo y diagrama de tiempos.

Tabla 6.9 Tabla de los FF SR con compuertas a) NAND y b) NOR.

a) Tabla FF SR con NANDs

S'R'Q	Q ₊₁	Q' ₊₁
0 0 x	1(∞)	1(∞)
0 1 x	1	0
1 0 x	0	1
1 1 0	0	1 (Q ₀)
1 1 1	1	0 (Q ₀)

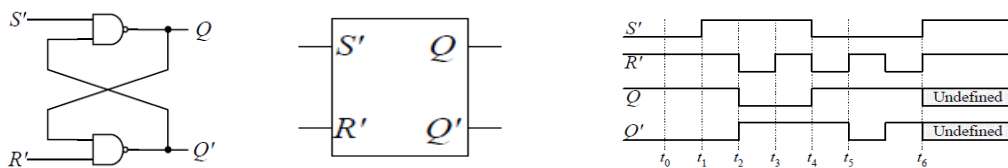
b) Tabla FF SR con NORs

S R Q	Q ₊₁	Q' ₊₁
0 0 0	0	1 (Q ₀)
0 0 1	1	0 (Q ₀)0
0 1 x	0	1
1 0 x	1	0
1 1 x	0(∞)	0(∞)

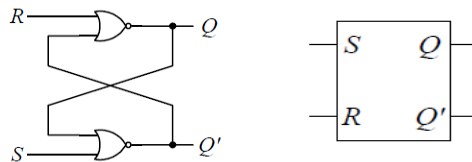
De ambas tablas, el flip flop SR, para una combinación e entradas SR la salida Q₊₁ conserva el estado por lo que Q₊₁=Q₀, para una segunda y tercera combinación la salida es forzada a ser 0 y 1, es decir, Q₊₁=0 y Q₊₁=1. En una última combinación las salidas Q₊₁ y Q'₊₁ son iguales, por lo que esa combinación está indefinida, ya que Q₊₁ y Q'₊₁ son de valores opuestos.

Las salidas de los flip fLops se denominan Q₊₁ y Q'₊₁, que significa el estado próximo, lo cual tendrá sentido cuando se incluya reloj a los flop flops.

En las figuras 6.11 a) y b) se presentan las estructuras internas de los FF SR implementados con compuertas NAND y con compuertas NOR. Además de su símbolo y diagrama de tiempos del FF SR implementado con compuertas NAND.



a) FF SR, su estructura interna con NANDs, símbolo y diagrama de tiempos.



b) FF SR. Su estructura interna con NORs y símbolo.

Figura 6.11 FF SR con compuertas a) NANDs y b) NORs

ii.- En Proteus simule los FF SR cuyas estructuras internas están implementadas con compuertas NAND y NOR.

iii.- Arme en un protoboard los FF SR implementadas con compuertas NAND y NOR , utilice los módulos de E/S del KIP y realice pruebas.

B.- Latch SR con señal de habilitación E (enable)

i.- Diseñe un Latch SR con compuertas NOR y con señal de habilitación E (enable). Un latch con señal de habilitación es un FF que se habilita por nivel, alto o bajo, pero qué durante el tiempo en que dura el nivel, pueden suceder cambios en las salidas Q_{+1} y Q'_{+1} .

En la tabla 6.10 se muestran los valores de verdad del latch SR para que sea implementada con compuertas NOR. De dicha tabla, si la señal de habilitación $E = 0$, no importa que valores tengan las entradas SR, la salida Q_{+1} será el valor que contenga Q . Cuando E está en 1, el latch SR se habilita para que funcione de acuerdo a las combinaciones de las entradas, ya sea manteniéndose el estado, forzando a que el estado del FF sea 0 o 1, o esté indefinido.

Tabla 6.10 Tabla de verdad del Latch SR con NORs.

E S R Q	Q_{+1}	Q'_{+1}
0 x x 0	0	1
0 x x 1	1	0
1 0 0 0	0	1 (Q_0)
1 0 0 1	1	0 (Q_0)
1 0 1 x	0	1 (Q_0)
1 1 0 x	1	0
1 1 1 x	$1(\infty)$	$1(\infty)$

En la figura 6.12 se muestra el circuito latch SR con señal de habilitación E cuya estructura contiene compuertas NOR, el símbolo y diagrama de tiempos.

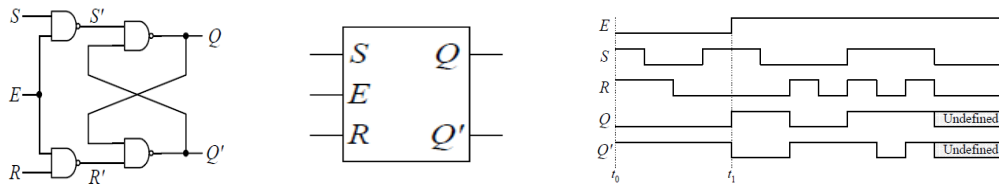


Figura 6.12 Latch SR con compuertas NORs, símbolo y diagrama de tiempos.

ii.- En Proteus simule el latch SR presentado en la figura 6.12

iii.- Arme en un protoboard el latch presentado en la figura 6.12 y realice pruebas.

3.4.2 Latch D y Latch D con señal de habilitación E

A.- Latch D

i.- Diseñe un latch D. Un latch D es un latch que trabaja con una entrada denominada D y la salida Q_{+1} toma los valores de la entrada D, es decir, $Q_{+1} = D$. Esto se muestra en la tabla 6.11.

Tabla 6.11 Tabla de verdad del Latch D.

D Q	Q_{+1}	Q'_{+1}
0 x	0	1
1 x	1	0

El circuito del latch D se muestra en la figura 6.13. Dicho latch D se apoya en la estructura del latch S'R', implementado con compuertas NAND, donde la señal de D se conecta directamente a la entrada R' y D se conecta a S'.. Además se el símbolo del latch D.

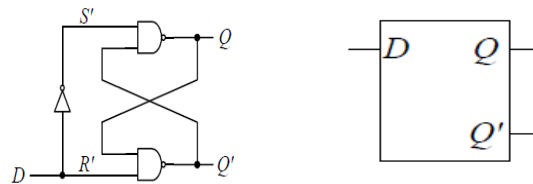


Figura 6.13 Latch D y símbolo.

ii.- En Proteus simule el circuito latch D.

iii.- Arme en un protoboard el latch D, utilice los módulos de E/S del KIP y realice pruebas.

B.- Latch D con señal de habilitación (E)

i.- Diseñe un latch tipo D con señal de habilitación E. Es decir es un latch D que si la señal E esta en bajo, $E = 0$, el latch D no funcionará correctamente. Pero si la señal E está en alto, el latch D funcionará correctamente, ver la tabla 6.12.

Tabla 6.12 Tabla de verdad del Latch D con señal de habilitación.

E D Q	Q_{+1}	Q'_{+1}
0 x 0	0	1
0 x 1	1	0
1 0 x	0	1
1 1 x	1	0

En la figura 6.14 se muestra el latch D con señal de enable E, su símbolo y diagrama de tiempos. De la estructura del latch D, la señal de habilitación se conecta a la entrada de un Mux 2 a 1. Si la señal E es 0, la salida Q_{+1} depende del valor de Q, no importando el valor de D, lo cual es un funcionamiento incorrecto. Si E está en 1, la salida Q_{+1} depende del valor de D, es decir $Q_{+1} = D$, lo cual es el funcionamiento correcto del latch D.

En el segundo circuito, extremo derecho, se muestra una segunda estructura del latch D, donde la señal de habilitación E y la entrada D se conectan a un arreglo de compuertas NAND.

ii.- En Proteus simule los dos circuitos del latch D con señal de habilitación E.

iii.- Arme en un protoboard los dos circuitos del latch D con señal de habilitación E, utilice los módulos de E/S y realice pruebas.

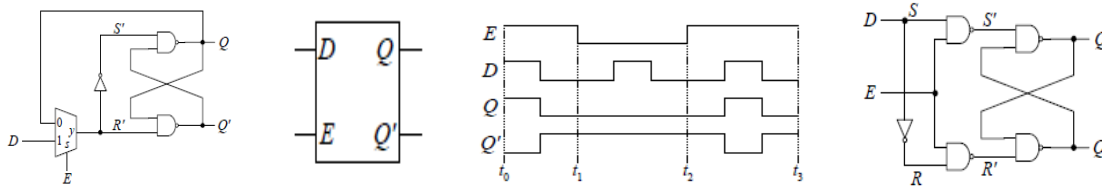


Figura 6.14 Latch D con señal de habilitación, símbolo y diagrama de tiempos

3.4.3 Flip Flop D MS Y SEÑALES ASÍNCRONAS

A.- Reloj

i.- Diseñe un FF D, que no sea sensible al nivel de la señal de habilitación E. Para ello se debe trabajar en dos aspectos una señal de sincronización para que uno o más elementos de memoria trabajen juntos y un circuito no sensible a nivel.

La sincronización se logra utilizando una señal de reloj. Para que realmente se logre una sincronización se debe trabajar en la transición positiva o negativa del reloj. Ya que se tendrán, en un circuito secuencial varios elementos de memoria.

Y para obtener un circuito no sensible a nivel se utilizará al flip flop. Un flip flop FF se compone de dos latch D, denominados maestro y esclavo. El Latch D maestro se habilita cuando el reloj (clk) está en 0. El latch D esclavo se habilita cuando clk está en 1. EL arreglo maestro esclavo del FF D permite que este funcione correctamente y no sea sensible a nivel.

En la tabla 6.13 se muestran los valores de verdad para el FF D.

Tabla 6.13 Tabla de verdad del FF D.

Clk D Q	Q ₊₁ Q' ₊₁
0 x 0	0 1
0 x 1	1 0
1 x 0	0 1
1 x 1	1 0
↑ 0 x	0 1
↑ 1 x	1 0

En la figura 6.15 se muestra el reloj, la estructura del FF D Maestro esclavo (MS), el símbolo del FF D y el diagrama de tiempos.

ii.- En Proteus simule el circuito FF D.

iii.- Arme en un protoboard el circuito FF D, conéctele los módulos de E/S del KIP y realice pruebas.

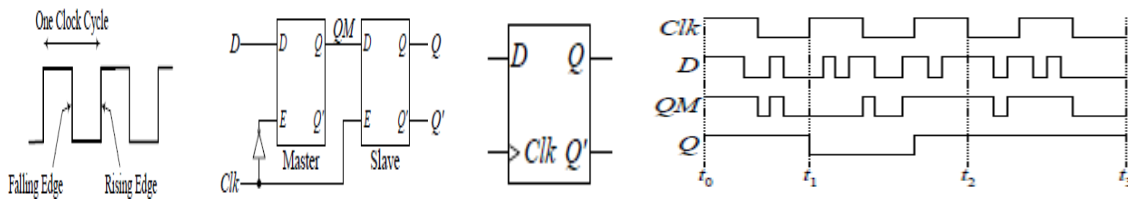


Figura 6.15 Reloj y FF D. Estructura, símbolo y diagrama de tiempos

B.- FF D Y SEÑALES ASÍNCRONAS

i.- Diseñe un FF D con señales externas asíncronas Set (S') y Clear (C'). Las señales externas S' y C' en un FF D lo fuerzan a que Q₊₁ sea 0 o sea 1 y no dependen del reloj. Estas señales nunca pueden ser 0 a la vez. La salida Q₊₁ será 0 si S' y C' tienen la combinación 10. Q₊₁ será 1 si S' y C' tienen la combinación 01, Si ambas son 1, el FF D funciona como tal. En la tabla 6.14 se muestra los valores de verdad para el FF D con S' y C'.

Tabla 6.14 Tabla de funcionamiento FF D con S y C'.

S' C' Clk D Q	Q ₊₁ Q' ₊₁
0 0 x x x	INVALIDO
1 0 x x x	0 1
0 1 0 x 0	1 0
1 1 0 x 0	0 1

1 1 0 x 1	1 0
1 1 1 x 0	0 1
1 1 1 x 0	1 0
1 1 ↑ 0 x	0 1
1 1 ↑ 1 x	1 0

El circuito que funciona de acuerdo a la tabla 6.14 se muestra en la figura 6.16 junto con su símbolo.

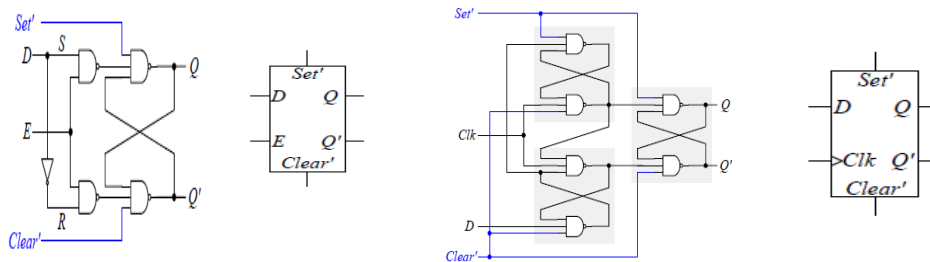


Figura 6.16 FF D con S'C' y símbolo. FF D con S'C' reloj y símbolo.

ii.- En Proteus simule el FF D con S'C' y reloj.

iii.- Arme en un protoboard el circuito FF D con S'C' y reloj, conecte los módulos de E/S y realice pruebas.

3.4.4 FLIP FLOPs EN CIRCUITOS INTEGRADO

A.- IC FF D 74LS74

i.- Estudie el IC 74LS74 FF D y realice lo siguiente.

Escriba la tabla de verdad del IC 74LS74

Dibuje el símbolo del IC del 74LS74

ii.- En Proteus simule el IC 74LS74.

iii.- En un protoboard pruebe al 74LS74, y utilice los módulos de E/S del KIP.

B.- IC FF JK 74LS73

i.- Estudie el IC 74LS73 FF JK y realice lo siguientes.

Escriba la tabla de verdad del IC 74LS73

Dibuje el símbolo del IC del 74LS73

ii.- En Proteus simule el IC 74LS73.

iii.- En un protoboard pruebe al IC 74LS73, utilice los módulos de E/S del KIP.

C.- IC FF JK 74L71

Para desarrollar esta parte realice lo siguiente:

i.- Estudie el IC 74L71 FF SR y realice lo siguiente.

 Escriba la tabla de verdad del IC 74L71

 Dibuje el símbolo del IC del 74L71

ii.- En Proteus simule el IC 74L71.

iii.- En un protoboard pruebe al IC 74L71 y utilice los módulos de E/S delKIP.

3.5 PREGUNTAS

i.- ¿Cómo realizaría un FF T a partir de un IC 74LS73?.

ii.- ¿Qué ventaja hay al disparar un FF por transición o flanco con respecto a dispararlo por nivel?.

iii.- ¿Por qué se consideran a los FF´s elementos de memoria?

3.6 REPORTE DEL ELUMNO (Resultados)

El indicado por el profesor

3.7 BIBLIOGRAFÍA PRELIMINAR

- Design Engineers. The TTL DataBook. 2a Ed. Texas Instruments.
- David G. Maxines y Jessica Alcala Jara. El arte de programar sistemas digitales. ITESM. Compañía Editorial Continental.
- Ronald J. Tocci. Sistemas Digitales: Principios y aplicaciones. 5a Edición. Prentice Hall
- John F. Wakerly, Diseño Digital Principios y Prácticas, Tercera edición, Pearson, México 2001.
- Morris Mano, M., Diseño Digital, Tercera edición, Pearson Educación, México 2003.

6.3 APLICACIONES CON MICROCONTROLADORES

A continuación se presentan un conjunto de prácticas de Microcontroladores que utilizan los módulos.

PRÁCTICA 1 MCU

ENTRADAS Y SALIDAS BÁSICAS

1.1 OBJETIVO

Que el alumno conozca y aplique técnicas para realizar aplicaciones con microcontroladores utilizando puertos, programando en ensamblador, que capture, simule, arme el circuito y realice pruebas.

1.2 INTRODUCCIÓN

Los circuitos con microcontroladores pueden resolver problemas utilizando la filosofía de cambiar el programa y mantener el mismo hardware para diferentes aplicaciones. Esto en contraste con los sistemas digitales, donde una aplicación nueva requiere un nuevo circuito digital.

En esta práctica se realiza la conexión de módulos de E/S, teclado lineal y leds respectivamente, al microcontrolador, para ejercitar el manejo de puertos. Esta práctica se guía para resolver un problema.

1.3 MATERIAL Y EQUIPO NECESARIO

Hardware:

- 1 Fuente 5Vcc.
- 1 Módulo Teclado Lineal
- 1 Módulo leds
- 1 MCU 16F84A, o 16F877 o entrenador 18F4550, u otro.

Software

- Proteus
- MPLAB

1.4 METODOLOGÍA

En teoría se deben conocer la arquitectura del microcontrolador a utilizar, los puertos, memoria, instrucciones, entre otros. Además de programar en ensamblador, así como el manejo de Proteus y MPLAB.

1.4.1 ENTRADA Y SALIDA POR PUERTOS

A continuación se presenta una serie de ejercicios prácticos para que experimente los circuitos con microcontroladores de entrada y salida básica.

1.4.1.1 Seguidor de niveles lógicos.

i.- Diseñe e implemente un circuito con microcontrolador que muestre por un puerto los valores de entrada que entran por otro puerto.

Para ello vea la figura 6.17, donde se muestra la generalidad de manejo del MCU con los módulos de E/S del KIP.

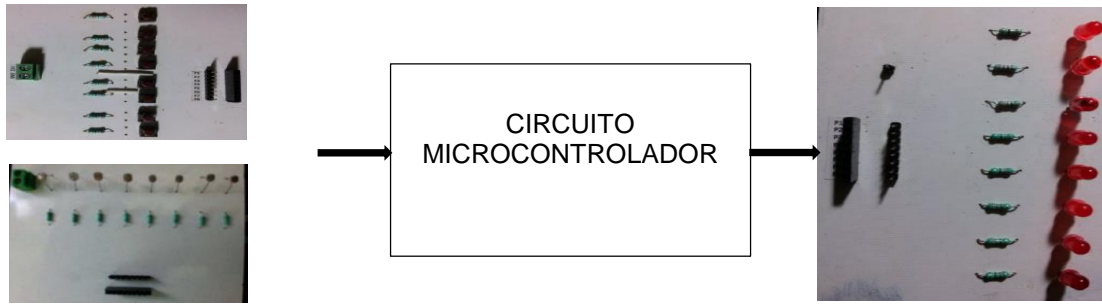


Figura 6.17 Circuito con Microcontrolador con E/S básica con módulos del KIP

La teoría de los módulos ya se conoce, el módulo teclado lineal suministra niveles lógicos y el módulo leds monitorea dichos niveles. Para el ejercicio se utilizará el MCU 16F84A, donde el puerto A (RA) tiene 5 líneas y el puerto B (RB) tiene 8 líneas. El puerto RA será de entrada y el puerto RB será de salida.

Algoritmo

- 1.- inicio
- 2.- Programar los puertos RA entrada, RB de salida
- 3.- Leer puerto RA (Módulo teclado lineal)
- 4.- Pasar el dato al RB (módulo leds)
- 5.- Regresar a 2

El código en ensamblador es

```

;Programa que lee el estado de las entradas del puerto RA y que se envían al puerto RB
;para monitorearse.
;
;Definición de registros (puertos, )
pc      equ    02h      ; contador de programa
status  equ    03h      ; registro e status para configurar puertos
ptoa    equ    05h      ;puerto A con dirección 05h
ptob    equ    06h      ; puerto B con dirección 06h
trisa   equ    85h      ; trisa con dirección 85h
trisb,  equ    86h      ; tris b condirección 86h
w       equ    00h.     ; w para guardar resultado

reset   org    00      ; vector de reset en la dirección de pila 00h
        goto   inicio  ; salta al inicio del programa

inicio  org    5       ; el programa inicia en la dirección de memoria 05h
        bsf    status, 5 ; Se ubica en el segundo banco de la RAM
        movlw 01f      ; se carga al registro w con ff
        movwf trisa    ; se programan los 5 pines del puerto RA como entrada
        movlw 0e0      ; se carga al registro w con e0
        movwf trisb    ; se programan los primero 5 pines del puerto RB como salida
        bcf   status,5 ; se ubica en el primer banco de memoria RAM

ciclo   movf   ptoa,w   ; el valor de entrada del Puerto RA lo pasa a w
        movwf ptob     ; el valor de w lo envía al Puerto RB

```

```
goto ciclo
end
```

```
;----- Fusibles de programación -----
:      Osc XT                -
;      Watchdog OFF          -
;      Code protect OFF      -
;      Power-Up-Timer ON     -
;-----
```

ii.- Capture y simule el circuito seguidor de niveles lógicos con Proteus

iii.- Arme el circuito seguidor de niveles lógicos en un protoboard, conecte los módulos de E/S, respectivos, del KIP y realice pruebas.

1.4.1.2 Inversor de datos leídos

i.- Diseñe e implemente un circuito con microcontrolador que lea una entrada de 5 bits e invierta su valor binario y lo muestre.

Esta aplicación es semejante a la del seguidor de niveles lógicos, sólo debe utilizar la instrucción “xorlw 0ff” para invertir el dato leído del puerto RA depositado en w.

ii.- Capture y simule el circuito inversor de datos leídos con Proteus.

iii.- Arme el circuito inversor de datos leídos en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas

1.4.1.3 Disparo de una alarma.

i.- Considere que tiene cinco sensores, y lo que se quiere es diseñar un circuito con MCU que dispare una alarma cuando hayan habilitados al menos cuatro sensores.

ii.- Capture y simule el circuito que dispara una alarma en Proteus

iii.- Arme el circuito que dispara una alarma en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas

1.4.2 PROPUESTAS DE NUEVAS PRÁCTICAS.

En cada una de las siguientes propuestas realice los pasos dados en las prácticas anteriores, para resolver cada caso.

i.- Diseñe un circuito con MCU que permita comparar dos datos de 2 bits cada uno, indicando si el primero es mayor, menor o igual al segundo.

ii.- Diseñe un circuito con MCU que genere código hexadecimal y se vea en un display.

iii.- Diseñe un circuito con MCU que al leer un dato del 0 al 9 genere su código exceso a 3.

1.5 PREGUNTAS

i.- ¿Qué otros ejemplos de circuitos con MCU daría?

ii.- ¿Puede diseñar aplicaciones para controlar algo en la casa con las prácticas vistas, comente.?

iii.- ¿En donde más aplicaría este tipo de circuitos.?

1.6 REPORTE DEL ALUMNO (Resultados)

El indicado por el profesor

1.7 BIBLIOGRAFÍA PRELIMINAR

- Eduardo Garcia Brejo, Compilador C CCS y simulador PROTEUS para Microcontroladores PIC, 1ª ed., Alfaomega Grupo Editor, S.A. de C.V., México
- José Ma. Angulo Usategui, Ignacio Angulo Martínez., Microcontroladores «PIC» Diseño Práctico de Aplicaciones, 2ª ed. Mc Graw Hill.
- John Iovine, PIC Microcontroller, Project Book, McGraw-Hill, 2000

PRÁCTICA 2 MCU

TECLADO 4x4

2.1 OBJETIVO

Que el alumno conozca y aplique técnicas para conectar los módulos teclado matricial y displays del KIP a los puertos de los microcontroladores, que programe en ensamblador, que capture, simule, arme el circuito y realice pruebas.

2.2 INTRODUCCIÓN

Los teclados conectados a los microcontroladores son de gran utilidad porque con un arreglo de N filas por M columnas se tienen hasta MxN teclas, esto ayuda mucho si se tienen pocas líneas disponibles en el microcontrolador. A continuación se muestra como conectar a un MCU un teclado matricial y un display.

2.3 MATERIAL Y EQUIPO NECESARIO

Hardware:

- 1 Fuente de 5Vcc.
- 1 Módulo Teclado Matricial 4x4
- 1 Módulo Display
- 1 MCU 16F84A, o 16F877 o entrenador 18F4550, u otro.

Software

- Proteus
- MPLAB

2.4 METODOLOGÍA

En teoría se deben conocer, además de las características del MCU a utilizar, el método de barrido de filas para detectar la posición de una tecla oprimida y la generación de código BCD en un Display. Además de programar en ensamblador, así como el manejo de Proteus y MPLAB.

2.4.1 Módulos Teclado 4x4 y display conectados al MCU.

i.- Diseñe e implemente un circuito con microcontrolador que se le conecta un teclado matricial de 4x4 y un display y que dependiendo de la tecla oprimida genere los códigos del Display en BCD, es decir 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, otros códigos.

Para ello apóyese en la figura 6.18, donde se muestra el circuito a armar, el MCU a utilizar y los puertos donde se conectan los módulos de E/S del KIP.

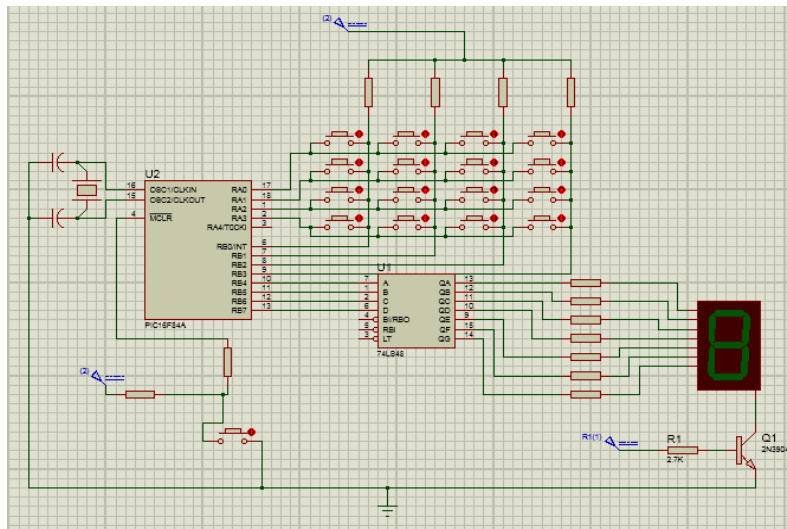


Figura 6.18 Módulos Teclado 4x4 y Display conectado a un MCU

Algoritmo

- 1.- inicio (Lectura de teclado)
- 2.- Programar los puertos RA<3:0> de salida filas del teclado y RB<3:0> de entrada de columnas del teclado.

- 3.- Colocar un cero en la primera fila RA<0:0>
- 4.- Leer columnas
- 5.- Si alguna columna leída esta en 0 continúa con el paso 8
Sino salta al paso 7
- 6.- Rutina de tratamiento de tecla
- 7.- Si el cero está en la cuarta fila pasa al paso 3
Sino desplazar cero en filas y salta al paso 4
- 8.- Terminar

El código en ensamblador es

```

;Programa que lee un teclado matricial y genera su código en un display en BCD.
;
;Definición de registros (puertos, )
pc      equ    02h      ;contador de programa
status  equ    03h      ;Registro de estados
ptoa    equ    05h      ;el puerto A esta en la dirección 05 de la RAM
ptob    equ    06h      ;el puerto B esta en la dirección 06 de la RAM
tecla   equ    0ch      ;contienen el valor de la tecla oprimida
loops   equ    0dh      ;utilizado en retardos (milisegundos)
loops2  equ    0eh      ;utilizado en retardos
rota    equ    0fh      ;registro que rota para enviar unos a las filas
filas   equ    10h      ;contiene el número de la fila a probar
trisa   equ    85h      ;Registro de configuración del puerto A
trisb   equ    86h      ;Registro de configuración del puerto B
z       equ    02h      ;bandera de cero del del registro de estados
c       equ    00h      ;bandera de carry del registro de estados
w       equ    00h      ;indica que el resultado se guarda en w

reset   org     0        ;el vector de reset es la direccion 00
        goto   inicio    ;Se salta al inicio del programa
        org    5         ;El programa empieza en la dirección 5 de la RAM

retardo                               ;subrutina de retardo de 100 milisegundos
        movlw  D'100'     ;El registro loops contiene el numero
        movwf  loops      ;de milisegundos del retardo
top2    movlw  D'110'     ;
        movwf  loops2     ;
top     nop
        nop
        nop
        nop
        nop
        nop
        decfsz loops2     ;pregunta si termino un ms
        goto  top         ;
        decfsz loops     ;Pregunta si termina el retardo
        goto  top2
        retlw  0

tabla   addwf  pc         ;sumar W al pc
        nop
        retlw  0         ;primera columna
        retlw  1         ;Segunda columna
        nop
        retlw  2         ;Tercera columna

```

```

nop
nop
nop
retlw 3 ;cuarta columna

inicio bsf status,5 ;Se ubica en el segundo banco de RAM
movlw 0f0h ;se carga el registro W con 0f0h
movwf trisa ;Se programa el puerto A como salida
movlw 0fh ;Se carga el registro W con 0FF
movwf trisb ;Se programa el puerto B como entrada y salida
bcf status,5 ;Se regresa al primer banco de memoria RAM
movlw 00h ; Cero para mostrar en el display
movwf tecla ;
ciclo swapf tecla,w ; intercambia 4 bits altos y bajos y quedan en W
movwf ptob ; pasa el valor de W al puerto B (display)
call retardo ; retardo
escan clrf filas ; se prepara para enviar ceros a las filas
movlw b'1110' ;
movwf rota ;
probar movf rota,w ; envia el dato a las filas
movwf ptoa ;
nop ; tiempo para estabilidad de las líneas
leer movf ptob,w ; leer las columnas conectadas al puerto B
andlw 0fh ; elimina 1a parte alta del byte leído
xorlw 0fh ; invierte el dato para ver si hay algun cero
btfss status,z ; pregunta si el resultado es cero (alguna tecla)
goto salir ; si hay tecla, mostrar en display
btfss rota,3 ; consulta si ya van 4 rotaciones
goto escan ; si termino, vuelve a empezar el escan de teclado
bsf status,c ; coloca bit de carry en 1
rlf rota ; para rotar el 0 que va a ir hacia las filas
movlw 4 ; carga W con 4 para sumarlo al valor de filas
addwf filas,1 ;
goto probar ; va a hacer la próxima prueba con el 0 rotado
salir call tabla ; para obtener valor de la columna
addwf filas,w ; sumar columna y filas para obtener el dato real
movwf tecla ; muestra el dato en el display
goto ciclo
end

;----- Fusibles de programación -----
; Osc XT -
; Watchdog OFF -
; Code protect OFF -
; Power-Up-Timer ON -
;-----

```

ii.- Capture y simule el circuito con Proteus

iii.- Arme el circuito en un protoboard, conecte los módulo de E/S del KIP y realice pruebas.

2.4.2. Inversión de datos leídos

i.- Diseñe e implemente un circuito con MCU que al leer una tecla del teclado matricial genere su código exceso a 3, Para ello utiulice como salida el módulo 4 displays.

ii.- Capture y simule el circuito inversor de datos leídos con Proteus.

iii.- Arme el circuito inversor de datos leídos en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas

2.4.3 Cerrojo Alarma.

i.- Con un código válido dado por teclado al MCU, abra un cerrojo, si no dispare una alarma.

ii.- Capture y simule el circuito que dispara una alarma en Proteus

ii.- Arme el circuito que dispara una alarma en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas

2.4.4 PROPUESTAS DE NUEVAS PRÁCTICAS.

En cada una de las siguientes propuestas realice los pasos dados en las prácticas anteriores, para resolver cada caso.

i.- Diseñe un circuito con MCU y Teclado matricial que permita realizar sumas de dos datos de 1 bit cada uno

2.5 PREGUNTAS

i.- ¿Qué otros ejemplos de circuitos de teclado matricial y MCU daría?

ii.- ¿Puede diseñar aplicaciones para controlar algo en la casa con las prácticas vistas, comente.?

iii.- ¿En dónde más aplicaría este tipo de circuitos.?

2.6 REPORTE DEL ALUMNO (Resultados)

El indicado por el profesor

2.7 BIBLIOGRAFÍA PRELIMINAR

- Eduardo Garcia Brejo, Compilador C CCS y simulador PROTEUS para Microcontroladores PIC, 1ª ed., Alfaomega Grupo Editor, S.A. de C.V., México
- José Ma. Angulo Usategui, Ignacio Angulo Martínez., Microcontroladores «PIC» Diseño Práctico de Aplicaciones, 2ª ed. Mc Graw Hill.
- John Iovine, PIC Microcontroller, Project Book, McGraw-Hill, 2000

PRÁCTICA 3 MCU

LCD

3.1 OBJETIVO

Que el alumno conozca y aplique técnicas para conectar el módulo LCD del KIP al MCU, que programe en ensamblador, que capture, simule, arme el circuito y realice pruebas.

3.2 INTRODUCCIÓN

Los LCD han permitido desplegar mensajes completos, a diferencia de los arreglos en paralelo con los displays. A continuación se muestra como conectar a un MCU un LCD.

3.3 MATERIAL Y EQUIPO NECESARIO

Hardware:

- 1 Fuente de 5Vcc.
- 1 Módulo LCD
- 1 MCU 16F84A, o 16F877 o entrenador 18F4550, u otro.

Software

- Proteus
- MPLAB

3.4 METODOLOGÍA

En teoría se deben conocer, además de las características del MCU a utilizar, la teoría del LCD de 2 x 16. Además de programar en ensamblador, así como el manejo de Proteus y MPLAB.

2.4.1 Módulos LCD conectado al MCU.

i.- Diseñe e implemente una aplicación con un MCU al que se le conecta el módulo LCD. Para ello apóyese en la figura 6.19, donde se muestra el circuito a armar, el MCU a utilizar y los puertos donde se conectan los módulos del KIP.


```

r0c    equ    0ch
r0d    equ    0dh
r13    equ    13h
z      equ    2h
c      equ    0h
w      equ    0h
r      equ    1h
e      equ    1h
rs     equ    0h

        org    00
        goto   inicio
        org    05h

retardo movlw  0ffh
        movwf  r13
decre   decfsz r13,r
        goto   decre
        retlw  0
control bcf    ptoa,rs
        goto   dato2
dato    bsf    ptoa,rs
dato2   bsf    ptoa,e
        movwf  ptob
        call   retardo
        bcf    ptoa,e
        call   retardo
        retlw  0

tabla2  addwfw pc,r
        retlw  "K"
        retlw  "I"
        retlw  "P"
        retlw  ""
        retlw  "K"
        retlw  "I"
        retlw  "T"
        retlw  ""
        retlw  "I"
        retlw  "N"
        retlw  "A"
        retlw  "L"
        retlw  "A"
        retlw  "M"
        retlw  "B"
        retlw  "R"
        retlw  "I"
        retlw  "C"
        retlw  "O"
        retlw  ""
        retlw  "D"
        retlw  "E"
        retlw  ""
        retlw  "P"
        retlw  "E"
        retlw  "R"

```


iii.- Arme el circuito en un protoboard, conecte el módulo LCD del KIP y realice pruebas.

2.4.2. Inversión de datos leídos

i.- Diseñe e implemente un circuito con MCU que al leer una tecla del teclado matricial genere su código exceso a 3, Para ello utiulice como salida el módulo 4 displays.

ii.- Capture y simule el circuito inversor de datos leídos con Proteus.

iii.- Arme el circuito inversor de datos leídos en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas

2.4.3 Cerrojo Alarma.

i.- Con un código válido dado por teclado al MCU, abra un cerrojo, si no dispare una alarma.

ii.- Capture y simule el circuito que dispara una alarma en Proteus

ii.- Arme el circuito que dispara una alarma en un protoboard, conéctele los módulos de E/S del KIP y realice pruebas

2.4.4 PROPUESTAS DE NUEVAS PRÁCTICAS.

En cada una de las siguientes propuestas realice los pasos dados en las prácticas anteriores, para resolver cada caso.

i.- Diseñe un circuito con MCU y Teclado matricial que permita realizar sumas de dos datos de 1 bit cada uno

3.5 PREGUNTAS

i.- ¿Qué otros ejemplos de circuitos de teclado matricial y MCU daría?

ii.- ¿Puede diseñar aplicaciones para controlar algo en la casa con las prácticas vistas, comente.?

iii.- ¿En dónde más aplicaría este tipo de circuitos.?

3.6 REPORTE DEL ALUMNO (Resultados)

El indicado por el profesor

3.7 BIBLIOGRAFÍA PRELIMINAR

- Eduardo Garcia Brejo, Compilador C CCS y simulador PROTEUS para Microcontroladores PIC, 1ª ed., Alfaomega Grupo Editor, S.A. de C.V., México
- José Ma. Angulo Usategui, Ignacio Angulo Martínez., Microcontroladores «PIC» Diseño Práctico de Aplicaciones, 2ª ed. Mc Graw Hill.
- John Iovine, PIC Microcontroller, Project Book, McGraw-Hill, 2000

6.3 APLICACIONES INALÁMBRICAS

A continuación se presentan un conjunto de prácticas de Microcontroladores que utilizan los módulos.

PRÁCTICA 1 INALÁMBRICA

COMUNICACIÓN CLIENTE SERVIDOR

1.1 OBJETIVO

Que el alumno conozca y aplique la comunicación punto a punto M/S con tecnología Bluetooth para control remoto, programando en C, que capture, simule, arme el circuito y realice pruebas.

1.2 INTRODUCCIÓN

En esta práctica se realiza la conexión de módulos de Bluetooth a MCUs para comunicar un maestro y un esclavo y controlar de manera remota el encendido y apagado de leds.

1.3 MATERIAL Y EQUIPO NECESARIO

Hardware:

- 1 Fuente 5Vcc.
- 2 Módulo Bluetooth
- 1 Módulo leds
- 2 módulo MCUs
- 2 16F877

Software

- Proteus
- MPLAB
- PIC C

1.4 METODOLOGÍA

En teoría se debe conocer el funcionamiento de la tecnología Bluetooth y la configuración de maestro y esclavo, además de programar en lenguaje C, así como el manejo de Proteus y MPLAB.

1.4.1 Comunicación M/S y control remoto de un led

i.- Diseñe e implemente un circuito para comunicación M/S con tecnología Bluetooth, para controlar el encendido de leds

Para ello vea la figura 6.20, donde se muestra la generalidad de manejo del MCU con los módulos de E/S del KIP.

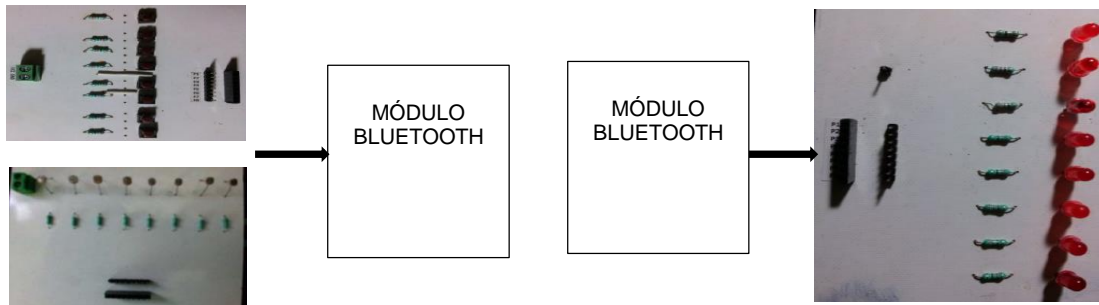


Figura 6.20 Módulos Bluetooth para control remoto de encendido de leds

Algoritmo

- 1.- inicio (Maestro PIC 16F877)
- 2.- Definir fusibles, velocidad del cristal y comunicación serial.
- 3.- Declaración variable de escritura (x) y
- 4.- Mientras que el valor de entrada por el pin A0 sea igual a 1
 - X asigne 1 (x = 1)
 - Envía x
 - Espera 0.5 segundos
 - Regresa al mientras
- 5.- Fin

Algoritmo

- 1.- inicio (Esclavo PIC 16F877)
- 2.- Definir fusibles, velocidad del cristal y comunicación serial, activar recepción
- 3.- Declaración variable de lectura (dato)
- 4.- Invocar a la función rda_isr y leer dato
- 5.- main()
- 6.- Habilita interrupciones (int_rda)
- 7.- Habilita interrupciones (global)
- 8.- Mientras sea verdadero
 - Si dato es igual a 1 (dato==1)
 - Envía 1 al pin_B6
 - Espera 0.5 segundos
 - Envía un 0 al pin_B6
 - Regresa al mientras
- 9.- Fin

El código en C es

```
/* Códigos en lenguaje C para conectar los módulos Bluetooth HC-05 y HC-06 a los
PIC16F877 para establecer la comunicación M/S
*/
```

```
//Código Maestro
```

```
#include <16f877a.h>
#fuses NOWDT, XT, NOPUT, NOPROTECT, NODEBUG, NOBROWNOUT, NOLVP,
NOCPD, NOWRT
#use delay(clock = 4000000)
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8 )
void main()
{
char x;
while(true)
{
if(input(pin_a0)==1)
{
x=1;
putc(x);//transmitir el dato
delay_ms(500);
}
}
}
```

```
//Código Esclavo
```

```
#include <16f877a.h>
#fuses NOWDT, XT, NOPUT, NOPROTECT, NODEBUG, NOBROWNOUT, NOLVP,
NOCPD, NOWRT
#use delay(clock = 4000000)
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8)
int dato;
#int_rda //activar recepcion
void rda_isr()
{
dato=getc();
}
void main()
{
enable_interrupts(int_rda);//interrupciones
enable_interrupts(global);
while(true)
{
if(dato==1)
{
Output_high(pin_b0);
delay_ms(500);
Output_low(pin_b0);
}
}
}
```

ii.- Capture y simule el circuito seguidor de niveles lógicos con Proteus, en la figura 6.21 se muestra una forma.

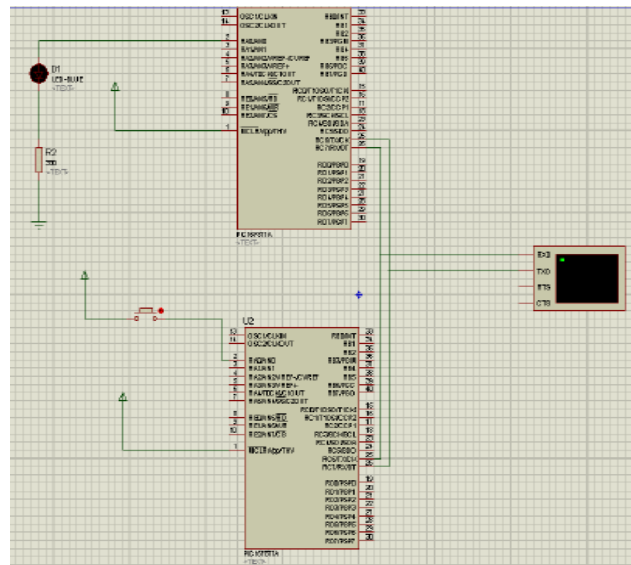


Figura 6.21 Conexión serial M/S de dos 16F877

iii.- Arme el circuito seguidor de niveles lógicos en un protoboard, conecte los módulos del KIP y realice pruebas. En la figura 6.22 se muestra una prueba de esta práctica.

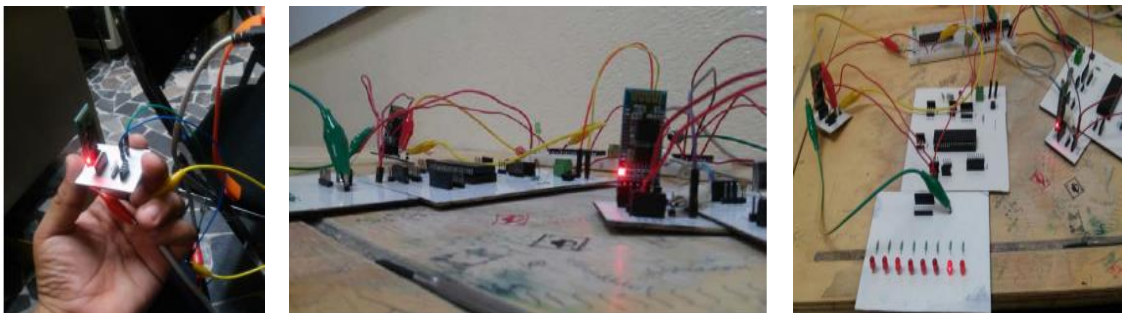


Figura 6.22 Prueba del módulo Bluetooth, conexiones y prueba led 6 encendido

1.4.2 Control remoto de 8 leds con tecnología Bluetooth y 16F877

i.- Modifique el software del ejemplo anterior para controlar de manera remota el encendido y apagado de 8 leds.

Esta aplicación es semejante a la del control remoto de un solo led.

ii.- Capture y simule el circuito control remoto de 8 leds utilizando Bluetooth y PICs 16F877 con Proteus.

iii.- Arme el circuito control remoto de 8 leds utilizando Bluetooth y PICs 16F877 en un protoboard, conéctele los módulos del KIP y realice pruebas.

1.4.2 PROPUESTAS DE NUEVAS PRÁCTICAS.

En cada una de las siguientes propuestas realice los pasos dados en las prácticas anteriores, para resolver cada caso.

i.- Controle el envío remoto de datos para un LCD, utilizando tecnología Bluetooth y PICs 16F877.

1.5 PREGUNTAS

i.- ¿Qué otros ejemplos de tecnología Bluetooth con 16F877 daría?

ii.- ¿Puede diseñar aplicaciones para controlar algo en la casa con las prácticas vistas, comente.?

iii.- ¿En dónde más aplicaría este tipo de circuitos.?

1.6 REPORTE DEL ALUMNO (Resultados)

El indicado por el profesor

1.7 BIBLIOGRAFÍA PRELIMINAR

- Eduardo Garcia Brejo, Compilador C CCS y simulador PROTEUS para Microcontroladores PIC, 1ª ed., Alfaomega Grupo Editor, S.A. de C.V., México
- José Ma. Angulo Usategui, Ignacio Angulo Martínez., Microcontroladores «PIC» Diseño Práctico de Aplicaciones, 2ª ed. Mc Graw Hill.
- John Iovine, PIC Microcontroller, Project Book, McGraw-Hill, 2000
- Nigel Gardner, An introduction to programming: The Microchip PIC in CCS C., Bluebird Electronics 2002.

CONCLUSIONES Y PERSPECTIVAS

CONCLUSIONES

Este trabajo originalmente presentaba módulos periféricos básicos como de leds, de displays, de teclado lineal y matricial, que era lo que más se utilizaba para la entrada y salida de datos digitales, aplicados al área de Hardware como Hardware II que es Sistemas Digitales y Hardware III que era Microprocesadores. Sin embargo, a la hora de retomar el tema para concluir, después de algunos días, se encuentra con que en el mercado los periféricos ya son muy diferentes, por lo que se optó por anexarle algunos módulos más al trabajo.

El trabajo propuesto se ha cumplido de inicio a fin, proporcionando un conjunto de periféricos de entrada salida así como un conjunto de prácticas tipo para las materias señaladas, así como la comunicación inalámbrica con la tecnología Bluetooth, por lo que se puede decir que se cumplió el objetivo general al 100%.

Aparte de este objetivo, se cumplió con Investigar y desarrollar prácticas del área de Digitales y Microcontroladores que incluya los módulos periféricos de entrada salida. Se investigó y probó la tecnología Bluetooth para la comunicación inalámbrica. Se investigó y aplicaron herramientas como MPLAB, PIC C y Proteus para programar y simular las prácticas. Se investigó y aplicó la herramienta PCB Wizard para realizar los circuitos impresos utilizando el método de planchado. Se desarrollaron prácticas y se documentó el trabajo.

PERSPECTIVAS

Hace algunos años se utilizan FPGA y CPLD para desarrollar sistemas digitales, sin embargo, los periféricos de E/S no cambian su funcionamiento, sólo trabajan valores eléctricos menores (0Vcc y 3.5Vcc), por lo que los módulos presentados, con unos cálculos de resistencias y otros dispositivos exhibidores, se podrán ajustar para ser utilizados con estos dispositivos configurables o reprogramables. También se pueden utilizar con los DPS (Procesadores Digitales de señales).

BIBLIOGRAFÍA

1. John F. Wakerly, Diseño Digital Principios y Prácticas, 3ª ed., Pearson, 2001.
2. Morris Mano, M., Diseño Digital, 3ª ed., Pearson Educación, 2003.
3. Frederick J. Hill. Gerald R. Peterson. Teoría de conmutación y diseño lógico. LIMUSA
4. Ronald J. Tocci. Sistemas Digitales: Principios y aplicaciones. 5a Edición. Prentice Hall
5. Eduardo Garcia Brejo, Compilador C CCS y simulador PROTEUS para Microcontroladores PIC, 1ª ed., Alfaomega Grupo Editor, S.A. de C.V., México
6. José Ma. Angulo Usategui, Ignacio Angulo Martínez., Microcontroladores «PIC» Diseño Práctico de Aplicaciones, 2ª ed. Mc Graw Hill.
7. John Iovine, PIC Microcontroller, Project Book, McGraw-Hill, 2000
8. Nigel Gardner, An introduction to programming: The Microchip PIC in CCS C., Bluebird Electronics 2002.
9. The Engineering Staff; The TTL Data Book for Design Engineers, 2a ed., Texas Instruments
10. Watson, Optoelectronica, Limusa, Mexico
11. Douglas V. Hall, Microprocessors and Interfacing. Programming and Hardware., Mc Graw Hill, 1986.