



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
ESCUELA NACIONAL DE ESTUDIOS SUPERIORES
UNIDAD JURIQUILLA
LICENCIATURA EN ÓRTESIS Y PRÓTESIS**



MANUAL DE PRÁCTICAS

Para Circuitos Digitales en órtesis y prótesis

**Autor
Dr. Octavio Díaz Hernández**

PAPIME PE102022

Título

Manual de prácticas para Circuitos Digitales en órtesis y prótesis

Materia

Circuitos Digitales

Autor

Dr. Octavio Díaz Hernández
Universidad Nacional Autónoma de México
Escuela Nacional de Estudios Superiores unidad Juriquilla
Licenciatura en órtesis y prótesis

Agradecimientos

Se agradece a la UNAM-DGAPA que, a través del proyecto PAPIME con clave PE102022 y titulado “Diseño de prácticas de laboratorio para las asignaturas del área de electrónica en la Licenciatura de Órtesis y Prótesis”, fue realizado este trabajo.

Licencia



Este trabajo has sido compartido por la UNAM bajo la licencia *Creative Commons* Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0)

Las imágenes exhibidas son de elaboración propia.

ÍNDICE DE PRÁCTICAS

Introducción del manual	1
1. Práctica 1. Comprobación de compuertas lógicas.....	2
2. Práctica 2. Función booleana combinacional	4
3. Práctica 3. Implementación de circuitos secuenciales.....	7
4. Práctica 4. Máquinas de estado algorítmico	10
5. Práctica 5. Memorias digitales.....	13
6. Práctica 6. Dispositivos Lógicos Programables, PLD	15
7. Práctica 7. Microcontroladores: Registros	18
8. Práctica 8. Microcontroladores: Operaciones Aritméticas	21
9. Práctica 9. Microcontroladores: Temporizadores.....	25
10. Práctica 10. Microcontroladores: Convertidor analógico-digital.....	29
11. Práctica 11. Microcontroladores: Interrupciones	33
12. Práctica 12. Microcontroladores: Modulación de ancho de pulso	36
13. Práctica 13. Microcontroladores: Comunicación serial.....	40
14. Proyecto integrador 1. Circuitos Secuenciales	44
15. Proyecto 2. Control de servomotores y señales analógicas.....	48
16. Referencias del manual	51

Introducción del manual

Los circuitos digitales son parte de la electrónica que se encarga específicamente de sistemas que funcionan con lógica binaria (ceros y unos). Las señales digitales se emplean para realizar diversas tareas y cumplir con diversos requisitos, por ejemplo, control y procesamiento de información. Los circuitos digitales tienen distintos grados de integración que se entienden como la composición interna de los dispositivos, como son las compuertas lógicas (AND, OR, NOT, XOR) que realizan operaciones lógicas y están integrados principalmente por transistores. Y así sucesivamente, dispositivos más complejos se van construyendo de elementos más básicos como son compuertas, flip-flops, decodificadores, contadores, decodificadores, entre otros [1].

Los alumnos aprenderán de forma experimental cómo un sistema digital se diseña mediante el uso de varios dispositivos lógicos ya sea de forma combinatorial o secuencial. Asimismo, se tiene componentes programables (p.ej. GAL, memorias, microcontroladores) en el temario de la asignatura que agrega competencias de microprogramación a nivel de lenguaje máquina (basado en sistemas digitales) y el manejo de periféricos como temporizadores, modulación por ancho de pulso (PWM) y convertidores analógico digital (ADC).

En esta asignatura a la que se dedica el manual, el alumno aplicará dispositivos de baja, media y alta escala de integración, así como diferentes metodologías y herramientas para el diseño de sistemas digitales. Entre las competencias a formar están: 1. Identificar las características técnicas de los dispositivos digitales, así como, el uso para la implementación de circuitos combinatoriales y secuenciales. 2. Usar la técnica de máquinas de estado para la solución de problemas con múltiples entradas y salidas para la solución de circuitos secuenciales. 3. Comprender el funcionamiento y la arquitectura de un microprocesador, así como, la secuencia de operación para la ejecución de instrucciones. 4. Experimentar la operación de un microcontrolador, así como de sus periféricos a través de la programación en lenguaje ensamblador y de alto nivel.

La importancia de la asignatura radica en que los sistemas en órtesis y prótesis cada vez más requieren que el profesional en esta área pueda diseñar nuevos componentes desde la perspectiva de la ingeniería mecatrónica.

1. Práctica 1. Comprobación de compuertas lógicas

1.1. Introducción

Las compuertas lógicas son circuitos con una o más señales de entrada, pero solo una señal de salida. Se analizan con ayuda de las tablas de verdad, en las que se enlistan las combinaciones posibles de las entradas y la salida a cada combinación, existen diferentes compuertas lógicas, siendo las principales la inversora "NOT", la multiplicación "AND", la suma "OR" [1].

1.2. Objetivos

- Recordar el uso de las compuertas lógicas y su funcionamiento.
- El alumno entenderá como se comprueba el correcto funcionamiento de las compuertas.

1.3. Recursos de espacio, materiales y equipo (incluyendo personal)

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Cuatro resistencias de $1k\Omega$
- Dos resistencias de 330Ω
- Cuatro botones
- Una compuerta AND (74LS08)
- Una compuerta OR (74LS32)
- Dos LED
- Un multímetro
- Una protoboard (tableta de pruebas)
- Un metro de cable de calibre 22

Equipo

- Una fuente de poder con al menos un canal

1.4. Descripción y desarrollo de la práctica

Se establece un circuito para seguir como el siguiente:

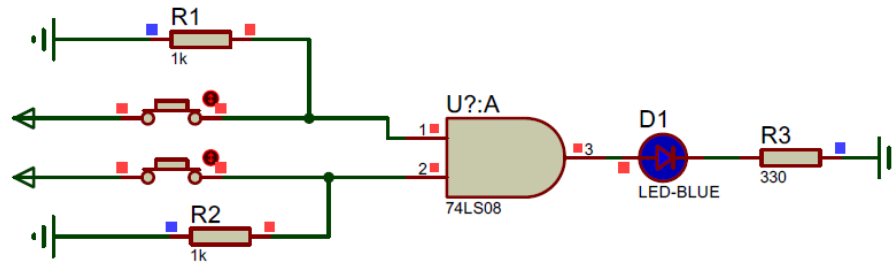


Figura 1. Circuito para comprobar la compuerta lógica AND.

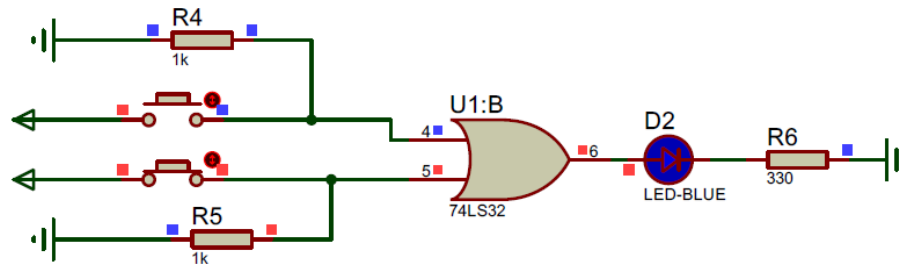


Figura 2. Circuito para comprobar compuerta lógica OR.

- I. La fuente de poder se configura a 5 V y se apaga.
- II. Se alamburan los circuitos propuestos en las Figura 1 y Figura 2.
- III. Se enciende la fuente.
- IV. Se activan los botones y se comprueba el funcionamiento de las compuertas siguiendo la tabla de verdad según sea el caso.
- V. Se comentan los resultados en el grupo.

1.5. Bibliografía recomendada

- Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.

2. Práctica 2. Función booleana combinacional

2.1. Introducción

Una función booleana es una variable binaria cuyo valor es igual al resultado de una expresión algebraica en la que se relacionan otras variables por medio de operaciones básicas, su valor lógico depende de las variables $a, b, c, \dots$ y puede ser igual a 1 o 0. Estas funciones se pueden reducir con ayuda de los mapas de Karnaugh y el álgebra de Boole [1].

2.2. Objetivos

- Comprender como hacer uso del algebra de Boole y de los mapas de Karnaugh para la reducción de las funciones booleanas.
- El alumno interpretará una función booleana y alambra el circuito de esta, con la menor cantidad de compuertas posibles.

2.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Cuatro resistencias de 1k
- Una resistencia de 330
- Cuatro botones
- Una compuerta
- Una compuerta AND (74LS08)
- Una compuerta OR (74LS32)
- Una compuerta NOT (74LS04)
- Una compuerta XOR (74LS86)
- Un LED
- Una protoboard (tableta de pruebas)
- Un metro de cable de calibre 22

Equipo

- Fuente de poder de al menos un canal

2.4. Descripción y desarrollo de la práctica

Se plantea un problema para resolver con lógica booleana. Se diseña una función booleana. Para esta práctica se ha establecido el siguiente mapa de Karnaugh y se realiza la función booleana

Tabla 1. Mapa de Karnaugh de función a trabajar

		$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
		00	01	11	10
$\bar{A}\bar{B}$	00			1	1
$\bar{A}B$	01				1
AB	11	1			
$A\bar{B}$	10	1	1		

$$\text{Función booleana} = (\bar{B} + \bar{D})(A \otimes C)$$

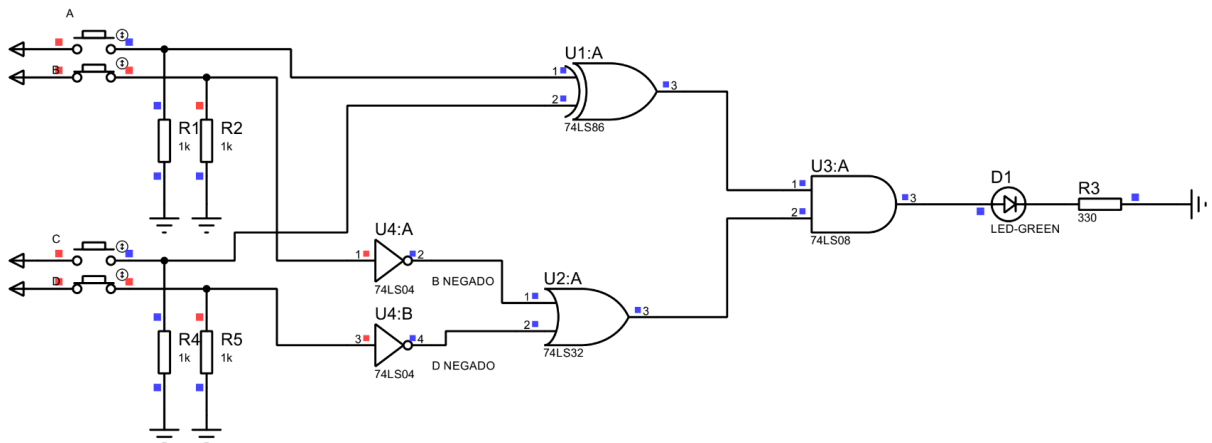


Figura 3. Esquema del circuito a alambrear según la función propuesta

- I. Se analiza el mapa de Karnaugh propuesto en busca de una función booleana.
- II. Con ayuda de las leyes del álgebra de Boole se simplifica la función según lo posible.
- III. Se conecta la fuente de poder, se regula a 5 volts y se apaga.
- IV. Se alambra el circuito que indica la función booleana de la Figura 3.
- V. Se enciende la fuente de poder.
- VI. Se comprueba el funcionamiento del circuito por medio de diferentes combinaciones en los botones observando si hay cambios en el LED.
- VII. Se apaga la fuente de poder

2.5. Bibliografía recomendada

- Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.

3. Práctica 3. Implementación de circuitos secuenciales.

3.1. Introducción

Los circuitos secuenciales son circuitos combinacionales a los que se les conecta una memoria para formar una retroalimentación. La información almacenada en los dispositivos de almacenamiento "memorias" va a definir el estado del circuito en ese momento. Estos circuitos también pueden recibir información binaria de entradas externas. Se pueden clasificar como "Síncronos" o "Asíncronos", los primeros van a funcionar como un reloj con una velocidad de operación constante, mientras que los segundos se controlan con otros eventos, tiene cambios de estado definidos por entradas y pueden tener un reloj en subcircuitos [1]. En esta práctica se usará como ejemplo un contador sincronizado a los pulsos de un tren de pulsos digitales, es cual es generado por un circuito integrado. En la Figura 4 se observa un proceso secuencial que tiene cuatro estados que se repetirán indefinidamente, en particular se observa el incremento de un contador de dos bits, iniciando por el cero (00), uno (01), dos (10), tres (11) y reinicia en cero.

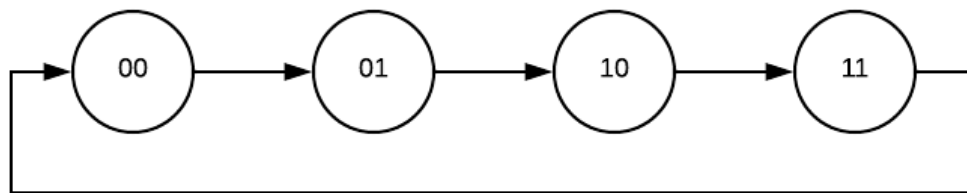


Figura 4. Estados secuenciales binarios del cero a tres.

3.2. Objetivos

- Comprender el funcionamiento de los dispositivos de almacenamiento de información binaria y su aplicación en los circuitos secuenciales.
- El alumno empleará una serie de circuitos integrados para generar un contador de pulsos digitales.

3.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Dos resistencias de 330 Ω

- Un capacitor de 100 uF
- Un capacitor de 1nF
- Un potenciómetro
- Un LED
- Un display de 7 Segmentos de cátodo común
- Un circuito integrado 74LS48 (decodificador de display de 7 Segmentos de cátodo común.
- Un circuito integrado 74LS74 (Flip Flop tipo D)
- Un circuito integrado 555
- Un circuito integrado 74LS86 (compuerta lógica XOR)
- Una protoboard

Equipo

- Una fuente de poder de 5 V.

3.4. Descripción y desarrollo de la práctica

Se establece un circuito para seguir como el siguiente que contará del 0 al 7 en forma automática:

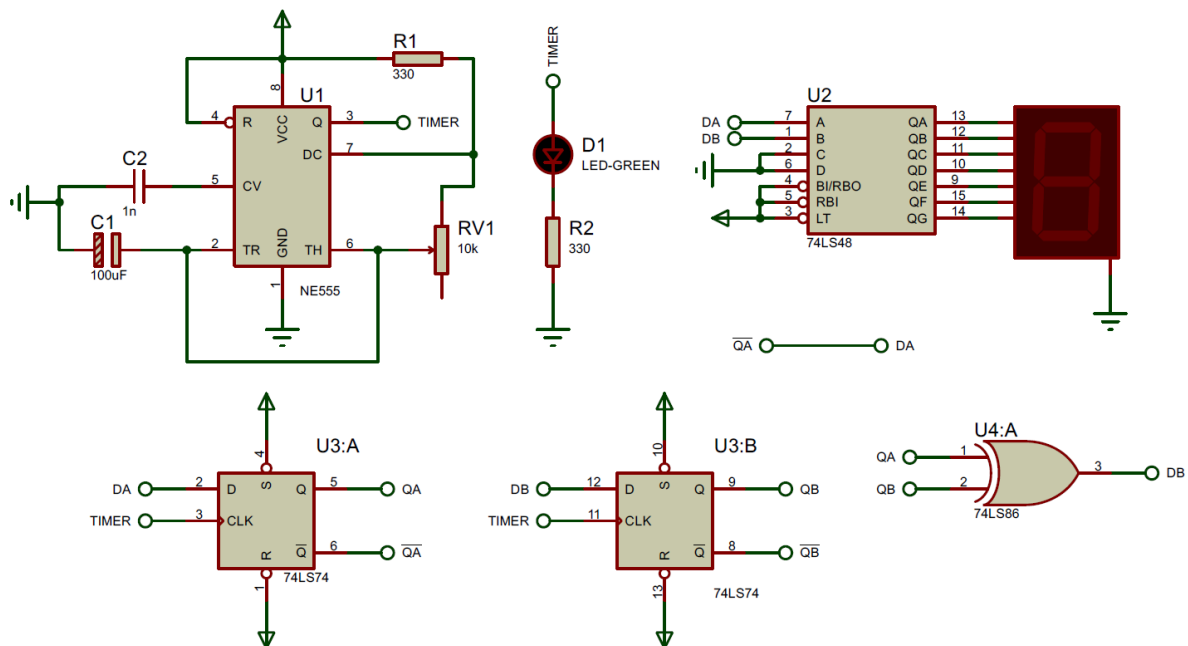


Figura 5. Esquema del circuito propuesto para alambrear

- Se conecta la fuente de poder, se regula a 5 volts y se apaga.

- II. Se alambra el circuito mostrado en la Figura 5. Se observan elementos como compuertas, generador de tren de pulsos (son los que se cuentan) y flip-flops (elementos de memoria binaria).
- III. Se enciende la fuente de poder.
- IV. Se comprueba el correcto funcionamiento del circuito observando los números desplegados en el display de siete segmentos.
- V. Se comentan los resultados en el grupo.

3.5. Bibliografía recomendada

- Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.

4. Práctica 4. Máquinas de estado algorítmico

4.1. Introducción

Las máquinas de estado algorítmico o *Algorithmic State Machine (ASM)* es un método para el diseño de circuitos digitales con estados binarios bien definidos. Se usa en forma de diagrama de estados, siendo un auxiliar para describir las operaciones por orden de un sistema digital en forma abstracta para plantear un problema que se resolverá con circuitos secuenciales. Un diagrama o carta ASM se compone de una interconexión de los tres tipos de elementos básicos: los estados, los controles condición, y salidas condicionales [1]. En esta práctica se ha definido una secuencia binaria de dos bits que seguirá un orden si la entrada (x) es igual a cero o si es igual a 1, dada cada condición, puede cambiar en forma instantánea hacia uno u otro estado, ver Figura 6.

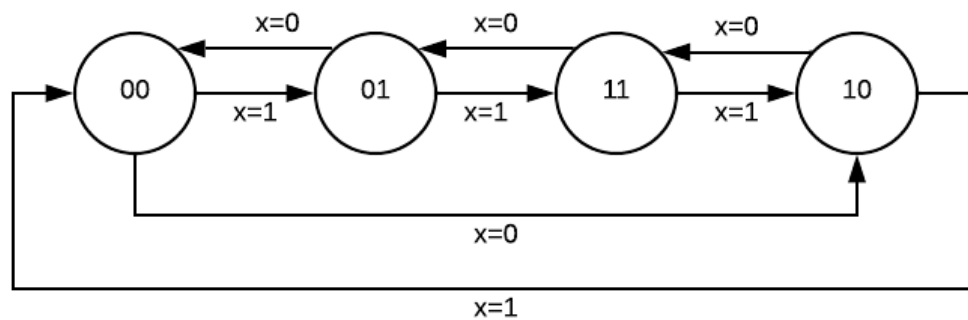


Figura 6. Secuencia binaria de dos bits que depende de una entrada x .

4.2. Objetivos

- Comprender el funcionamiento de las máquinas de estado con una entrada.

4.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Un botón pulsador
- Una resistencia de $1k\Omega$
- Un motor a pasos

- Un display de 7 Segmentos de cátodo común
- Un circuito integrado 74LS48 (decodificador de display de 7 Segmentos de cátodo común.
- Un circuito integrado 74LS74 (Flip Flop tipo D)
- Un circuito integrado 555
- Un circuito integrado 74LS86 (compuerta lógica XOR)
- Un circuito integrado 74LS04 (compuerta lógica NOT)
- Una protoboard
- Un metro de cable de calibre 22

Equipo

- Fuente de poder con al menos un canal de salida
- Generador de funciones.
- Multímetro

4.4. Descripción y desarrollo de la práctica

Se establece un circuito para seguir como el siguiente:

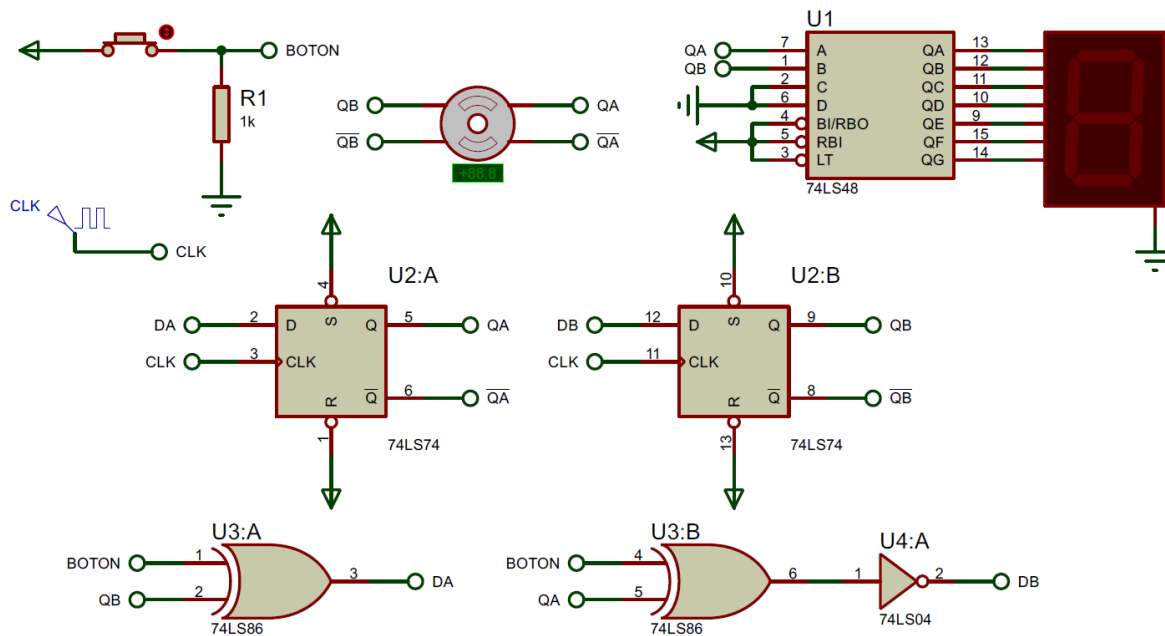


Figura 7: Esquema del circuito a alambrear. Ejemplo con un motor a pasos con control de giro.

- I. La fuente de poder regulable se configura a 5 V y se apaga.
- II. Se alambra el circuito propuesto en la Figura 7.

- III. La señal de CLK es un generador de funciones configurado como señal cuadrada y frecuencia de 10 Hz
- IV. Se enciende la fuente de poder.
- V. Se comprueba el correcto funcionamiento del circuito y sus cambios con la interacción del botón.
- VI. Se comentan los resultados en el grupo.

4.5. Bibliografía recomendada

- Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.

5. Práctica 5. Memorias digitales

5.1. Introducción

En informática, la memoria es el dispositivo que retiene, memoriza o almacena datos informáticos durante algún periodo de tiempo. La memoria proporciona una de las principales funciones de la computación moderna: el almacenamiento de información y conocimiento. Se mencionarán algunos de los tipos de memoria en circuitos digitales: Memoria RAM: *Random Access Memory*. Memoria ROM: *Read Only Memory*, Memoria de solo lectura. Memoria PROM: *Programmable ROM*, memoria programable de sólo lectura. Memoria EPROM: *Erasable PROM*, memoria borrable (con luz). Memoria EEPROM: *Electrically EPROM*, memoria borrable electrónicamente [1].

En la Figura 8 resaltan las entradas y salidas requeridas en el empleo de las memorias. Los “datos” almacenados en las distintas celdas de una memoria pueden ser extraídos usando la “dirección” específica de la celda que lo contiene.

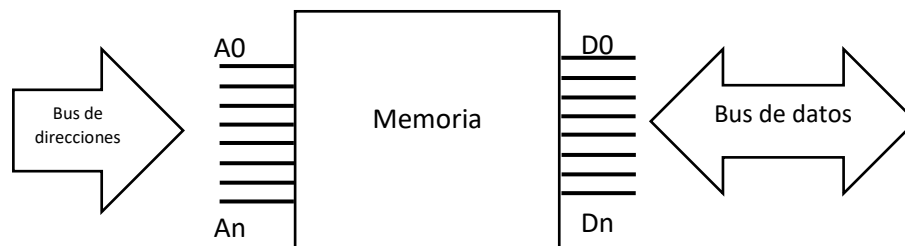


Figura 8. Esquema de una memoria.

5.2. Objetivos y competencias para formar

- Comprobar el funcionamiento de una memoria eléctricamente borrable.
- El alumno resolverá un problema de números binarios usando las celdas de la memoria.

5.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Una memoria 27C256
- Un circuito integrado contador 74LS193

- Un display de 7 segmentos BCD
- Una protoboard
- Un metro de cable de calibre 22

Equipo

- Fuente de poder
- Generador de funciones

5.4. Descripción y desarrollo de la práctica

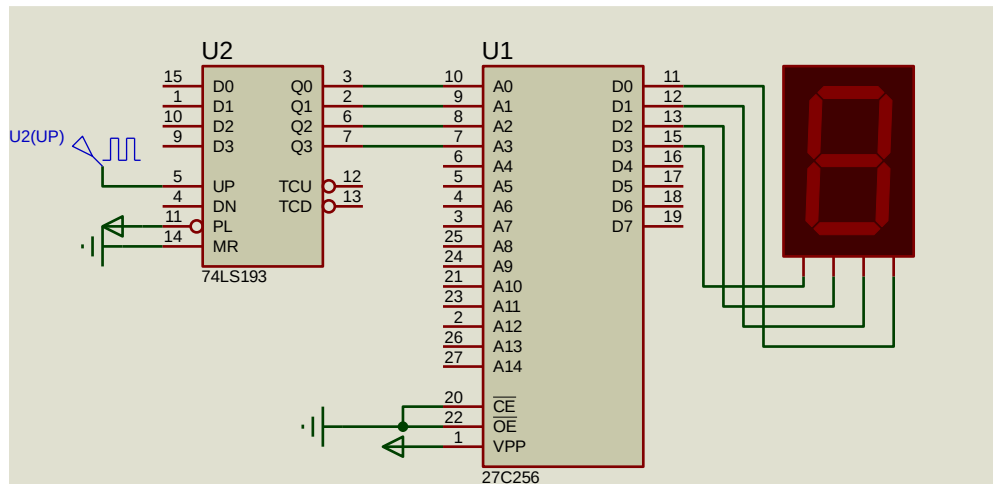


Figura 9. Circuito contador con una memoria.

- I. Para esta práctica se requerirá el apoyo de algún software (programa en computadora) para diseñar y grabar la memoria
- II. Se alambra el circuito de la Figura 9
- III. Se emplea el generador de funciones para obtener una señal cuadrada con frecuencia de 1 Hz.
- IV. Se energiza el circuito con la fuente de poder en 5V.
- V. Se observa el comportamiento y se verifica que los números o letras sea el correcto dado el diseño de los datos grabados en la memoria.
- VI. Se sugiere la siguiente secuencia: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
- VII. Se comentan los resultados en el grupo.

5.5. Bibliografía recomendada

- Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.

6. Práctica 6. Dispositivos Lógicos Programables, PLD

6.1. Introducción

Los Dispositivos Lógicos Programables (PLD) son parte de una familia de componentes que contienen arreglos de elementos lógicos (AND, OR, NOT, FLIP-FLOP) interconectados internamente y que pueden ser configurados con "cualquier" función lógica, ya sea secuencial o combinacional [1,3].

En la Figura 10, se observa un modelo de la estructura interna más común de estos dispositivos. Teniendo en cuenta en el marco teórico de la clase que acompaña al manual, no todos los tipos de estos dispositivos han permanecido en uso hasta la actualidad, sin embargo, se muestran al alumno por dos razones: la primera es que son parte del objetivo de la asignatura el mostrar las distintas escalas de integración de elementos menos complejos para formar sistemas digitales más avanzados hasta llegar a un microcontrolador, y la segunda razón es que si existe un tipo de PLD en la actualidad, el cual es la matriz de puertas lógicas programable en campo o FPGA (del inglés *field-programmable gate array*), no obstante es un tema de estudio a profundidad que escapa de los objetivos del manual y de la clase.

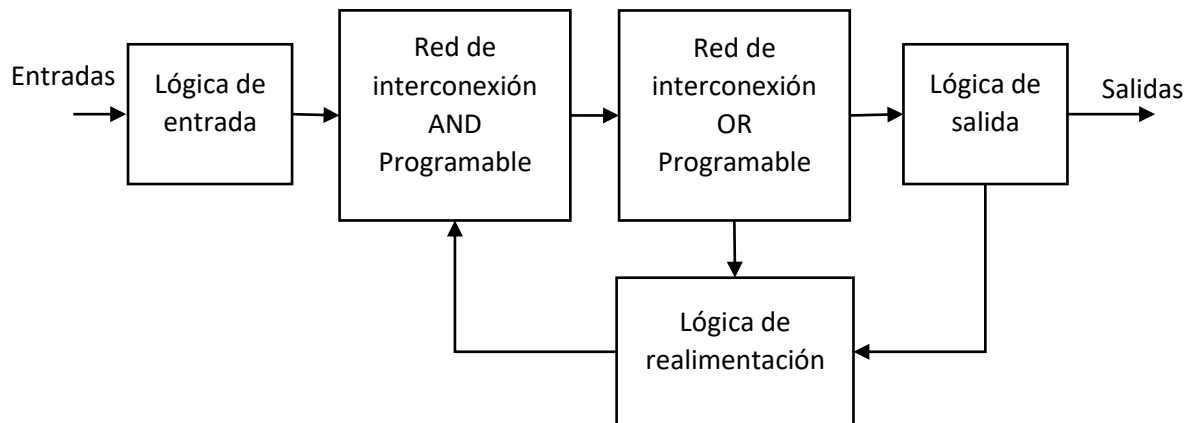


Figura 10 Arquitectura básica de un dispositivo programable por el usuario

Han existido distintos tipos de dispositivos lógicos programables, entre los cuales está una denominada Matriz de Lógica Genérica o GAL (*Generic Array Logic*) un ejemplo que se usará en esta práctica y se recomienda el uso de simuladores en caso de no contar con el hardware requerido.

6.2. Objetivos

- Conocer el funcionamiento de un dispositivo programable integrado por compuertas lógicas y flip-flops interconectadas.
- El alumno comprenderá el uso de una GAL mediante el uso de hardware o software especializado.

6.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Una protoboard
- Un botón pulsador
- Dos resistencias de 1 K Ω
- Un dispositivo lógico programable GAL (G22V10)
- Un motor a pasos
- Un circuito integrado de potencia para el motor (L293D)
- Programa para diseñar el archivo que se grabará en el GAL

Equipo

- Fuente de poder programable
- Generador de funciones

6.4. Descripción y desarrollo de la práctica

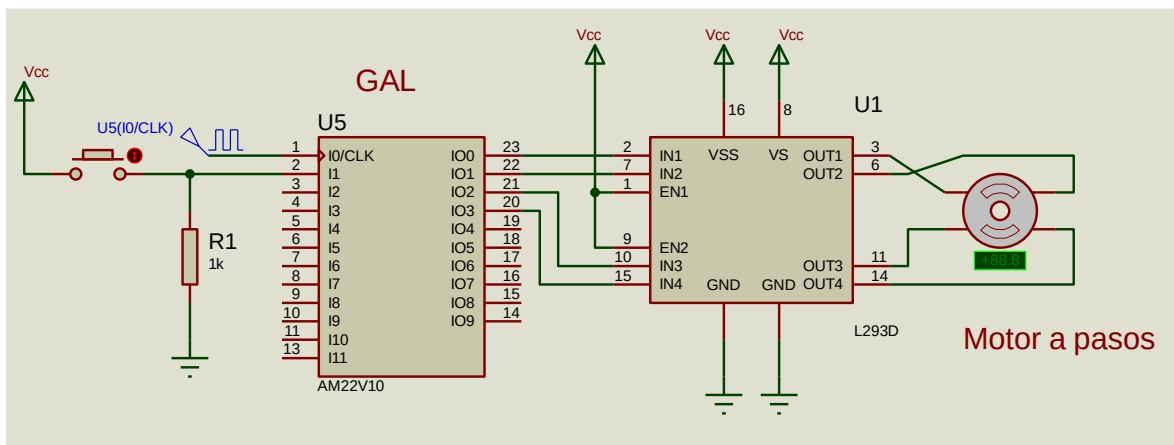


Figura 11. Diagrama de conexiones para probar la GAL

Una vez que se tenga el circuito, se genera un programa para establecer las funciones booleanas (en este caso serán secuenciales) y a continuación, se

muestra las líneas de programa que coordinan el movimiento del motor a pasos, previamente diseñadas con ayuda del diseño con funciones con lógica secuencial, en particular con flip-flop tipo D.

Tabla 2. Programa sugerido.

```
Device      G22V10 ;

Definición de pines de entrada
PIN 2 = X;
Definición de pines de salida
PIN 23 = A;
PIN 22 = B;
PIN 21 = AN;
PIN 20 = BN;
Definición de funciones
A.D = !(X$B);
AN.D = (X$B);
B.D = (X$A);
BN.D = !(X$A);
```

- I. Una vez que se tenga el circuito y el programa dentro de la GAL, se conecta a la fuente de poder configurada a 5 V.
- II. Se enciende la fuente de poder y se observa el movimiento del motor a pasos.
- III. Con ayuda del botón (entrada X), al ser presionado, el sentido de giro se invierte.
- IV. Se comentan los resultados en el grupo.

6.5. Bibliografía recomendada

- Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

7. Práctica 7. Microcontroladores: Registros

7.1. Introducción

Los microcontroladores son dispositivos programables que contienen en su arquitectura los tres componentes básicos de un microcomputador: el procesador, la memoria y una interfaz de entradas y salidas. Durante estas prácticas se irán usando los distintos dispositivos o periféricos característicos de los micros [2,3]. En este manual se usará *lenguaje ensamblador* y se usan ejemplos de la programación de un PIC de gama media (PIC16F887). Sin embargo, se pueden usar equivalencias.

En el lenguaje ensamblador, la manipulación de registros requiere instrucciones como son, por ejemplo, MOVF y MOVWF. Por un lado, MOVF, permite transferir el valor de un registro hacia otro registro, por ejemplo, se utiliza para asignar valores al registro de trabajo **W**, mientras que la función MOVWF asigna el contenido de **W** a otro registro donde se desee guardar [2,3].

7.2. Objetivos

- Comprender las instrucciones en el lenguaje ensamblador referidas a la manipulación de valores de algunos registros.
- El alumno implementará un programa para desplegar el valor binario de un puerto de entrada en otro de salida.

7.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material sugerido

- Un PIC16F887
- Un botón pulsador
- Nueve resistencias de 1 k Ω
- Ocho resistencias de 330 Ω
- Un *dipswitch* de 8 palancas
- Una barra de 10 LED
- Un cristal de cuarzo de 20 MHz
- Dos capacitores 22 pF
- Una protoboard

Equipo

- Fuente de poder

7.4. Descripción y desarrollo de la práctica

Puede emplearse el siguiente diagrama de la Figura 12 que contiene un PIC16F887, pero es posible usar otro microcontrolador que tenga las capacidades equivalentes de puertos de entrada y salida.

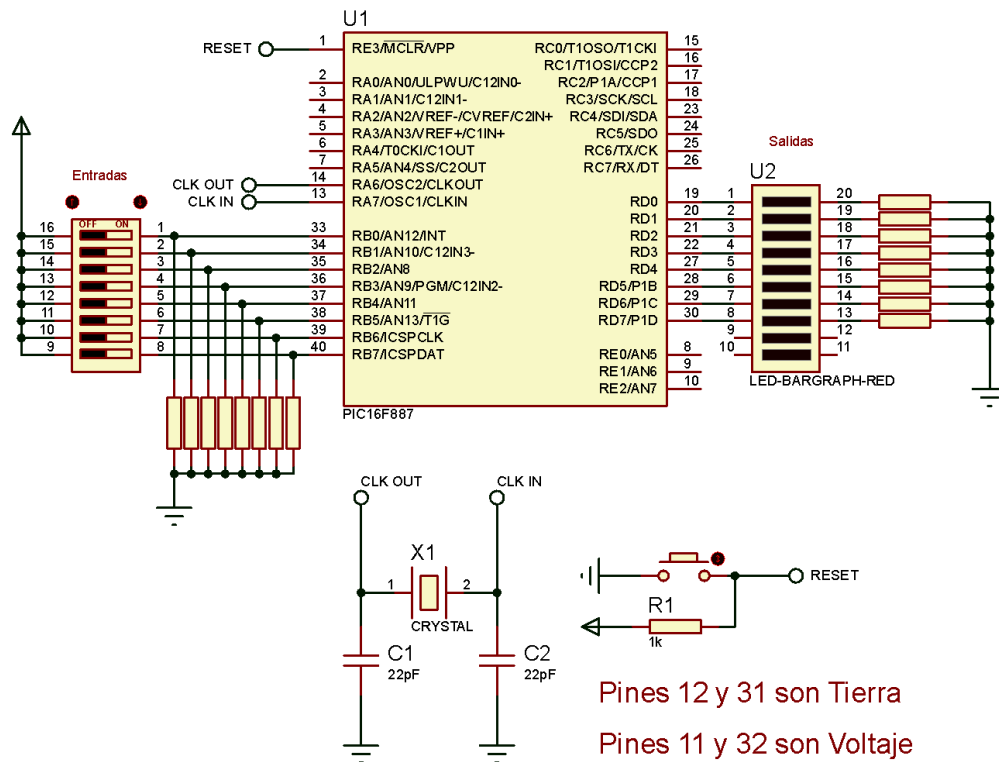


Figura 12. Esquemático de un microcontrolador y conexiones requeridas.

Mediante un dipswitch de 8 interruptores se conectará a los pines correspondientes a un puerto configurado como de entrada (p.ej. Puerto B del PIC16F887) y una barra de 8 LED será conectada a los pines de un puerto de salida del mismo dispositivo (p. ej. Puerto D del PIC16F887) con el fin de desplegar el estado de los binario del puerto C.

A continuación, se muestra el extracto de un código requerido para transferir el valor de un puerto de entrada a uno de salida, usando las instrucciones sugeridas.

Tabla 3. Programa en lenguaje ensamblador sugerido.

Etiquetas	Instrucción	Argumento	Comentario de programador
LOOP	MOVF	PORTC, W	Se le asigna el valor del registro del Puerto C al registro de trabajo o W
	MOVWF	PORTD	El valor del acumulador se le asigna al registro del Puerto D
	GOTO	LOOP	

- I. Se alambra un circuito con las características previamente mencionadas y se puede usar el de la Figura 12.
- II. Se realiza la programación en lenguaje de ensamblador para desplegar el valor binario de un puerto de entrada en otro de salida. En la Tabla 3 muestra las instrucciones principales con las que se refleja un puerto de entrada en uno de salida. Considerar la definición de puertos de entrada y salida en el lenguaje elegido.
- III. Se verifica el funcionamiento del circuito,
- IV. Se hacen las correcciones necesarias, en caso de requerirlas y se comenta en el grupo.

7.5. Bibliografía recomendada

- Fernando Valdés Pérez. Microcontroladores, Fundamentos y aplicaciones PIC, Alfaomega, México, 2007.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

8. Práctica 8. Microcontroladores: Operaciones Aritméticas

8.1. Introducción

Los microcontroladores son dispositivos programables que contienen en su arquitectura los tres componentes básicos de un microcomputador: el procesador, la memoria y una interfaz de entradas y salidas [2]. Durante estas prácticas se irán usando los distintos dispositivos o periféricos característicos de los micros. En este manual se usará *lenguaje ensamblador* y se usan ejemplos de la programación de un PIC de gama media (PIC16F887). Sin embargo, se pueden usar equivalencias.

La unidad lógico-aritmética, también conocida como ALU (siglas en inglés de *arithmetic logic unit*), es un circuito digital integrado al microcontrolador que realiza operaciones aritméticas básicas (p.ej. suma, resta) y operaciones lógicas (p.ej. AND, OR, XOR) entre los valores de sus entradas argumentos (uno o dos) [2].

En esta práctica se tienen dos puertos como entradas y uno de salida para desplegar el resultado de la operación elegida, por ejemplo, se tienen varias opciones elementales que son suma y resta, aunque se puede usar también las operaciones lógicas de AND (multiplicación lógica), OR (suma lógica) y XOR (suma exclusiva lógica).

8.2. Objetivo

- Aplicar las instrucciones que realizan las operaciones aritméticas y lógicas usando los valores de los registros en un microcontrolador.
- El alumno empleará puertos de entrada y de salida para interactuar con el proceso de las operaciones que se realizan en el interior del microcontrolador.

8.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Un PIC16F887
- Un botón pulsador
- 17 resistencias de 1 k Ω

- Ocho resistencias de 330 Ω
- Dos *dipswitch* de 8 interruptores
- Una barra de 8 LED
- Un cristal 20 MHz
- Dos capacitores 22 pF
- Una protoboard
- Un metro de cable de calibre 22

Equipo

- Fuente de poder

8.4. Descripción y desarrollo de la práctica

Puede emplearse el siguiente diagrama de la Figura 13 que contiene un PIC16F887, pero es posible usar otro microcontrolador que tenga las capacidades equivalentes de puertos de entrada y salida.

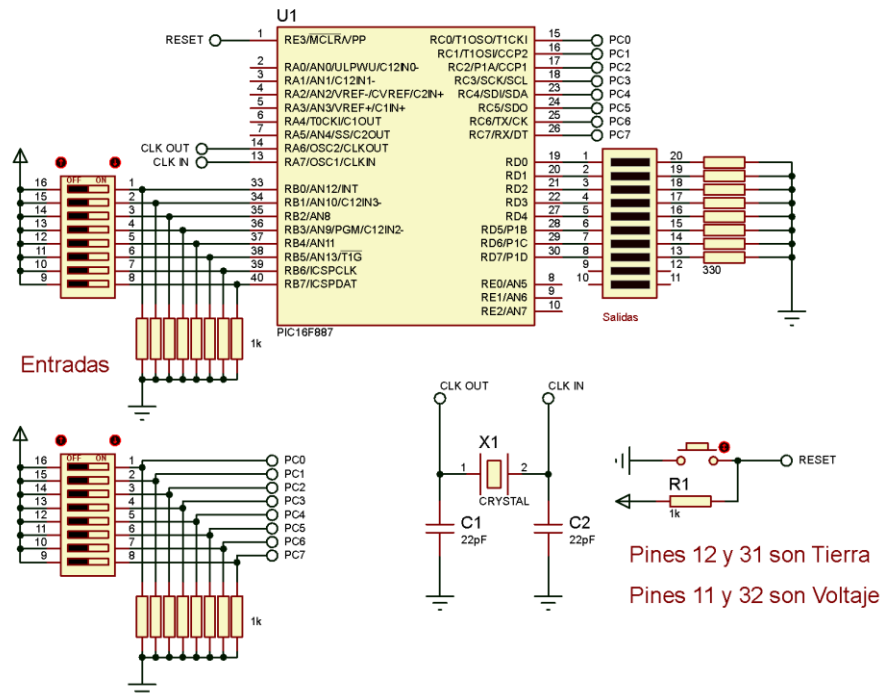


Figura 13. Esquemático de un microcontrolador y conexiones requeridas.

Mediante el uso de los puertos del microcontrolador se puede hacer la operación de "suma" de valores en número binario del puerto B y el puerto C para desplegar el resultado de la operación en el puerto D. Para ello se colocarán *dipswitch* de 8 interruptores en los puertos B y C para que el usuario ingrese el valor de 8 bits. El resultado binario de 8 bits se despliega en el puerto D, el cual tendrá conectada

una barra de LED. Cabe recordar que la suma de ambos puertos no puede ser mayor a 255 debido a que existiría un desborde de datos y regresaría a 0.

Las instrucciones en ensamblador que contienen el prefijo ADD (de suma en inglés) son las que se pueden usar para realizar la operación entre el acumulador (ADDWF) o con una literal (ADDLW). Eso mismo ocurre con la resta y las operaciones lógicas. En esta práctica se usará la suma y se puede agregar o cambiar las instrucciones para verificar otras operaciones que realiza la ALU [2].

Tabla 4. Programa en lenguaje ensamblador sugerido.

Etiquetas	Instrucción	Argumento	Comentario de programador
LOOP	MOVF	PORTB, W	Se le asigna el valor del registro del Puerto B al registro de trabajo o W
	ADDWF	PORTC, W	Se suma el contenido de W (valor del puerto B, cargado con la instrucción previa) y se guarda en W
	MOVWF	PORTD	El valor del acumulador se le asigna al registro del Puerto D, en este momento $W = \text{PUERTO B} + \text{PUERTO C}$
	GOTO	LOOP	

- I. Se alambra un circuito con las características previamente mencionadas y se puede usar el de la Figura 13.
- II. Se realiza la programación en lenguaje de ensamblador para desplegar el resultado de la suma en valor binario de un puerto de entrada más el valor de otro puerto entrada en otro de salida. En la Tabla 4 muestra las instrucciones principales con las que se refleja un puerto de entrada en uno de salida. Considerar la definición de puertos de entrada y salida en el lenguaje elegido.
- III. Se prueba el circuito cambiando lo valores con ayuda de los *dipswitch*. Cabe recordar que la suma de ambos puertos no puede ser mayor a 255 debido a que existiría un desborde de datos y en el puerto de 8 bits no se vería la información completa.
- IV. Se hacen las correcciones necesarias, en caso de requerirlas y se comenta en el grupo.
- V. Se cambian las instrucciones para observar la resta y las operaciones lógicas.
- VI. No se requieren cambios en el circuito, solo la programación.

8.5. Bibliografía recomendada

- Fernando Valdés Pérez. Microcontroladores, Fundamentos y aplicaciones PIC, Alfaomega, México, 2007.

- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

9. Práctica 9. Microcontroladores: Temporizadores

9.1. Introducción

Los microcontroladores son dispositivos programables que contienen en su arquitectura los tres componentes básicos de un microcomputador: el procesador, la memoria y una interfaz de entradas y salidas [2]. Durante estas prácticas se irán usando los distintos dispositivos o periféricos característicos de los micros. En este manual se usará *lenguaje ensamblador* y se usan ejemplos de la programación de un PIC de gama media (PIC16F887). Sin embargo, se pueden usar equivalencias.

En los microcontroladores, uno de los periféricos más comunes es el temporizador. Como su nombre lo indica es un auxiliar en la medición de tiempo o si es requerido un retraso entre un proceso y otro. Se pueden encontrar uno o más temporizadores dependiendo del tipo de microcontrolador o de fabricante, pero siempre existirán [2,3]. En esta práctica se usará el denominado *Timer 0* del PIC16F887, para lo cual se emplean bits de registros y registros completos para su funcionamiento, no obstante, se pueden consultar las particularidades de los temporizadores en las hojas de especificaciones de cada microcontrolador.

9.2. Objetivos

- Construir una función de retardo con el uso de temporizadores.
- El alumno aprenderá a usar un periférico de temporización (*Timer 0*) del PIC16F887.

9.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Un PIC16F887
- 17 resistencias de 1 k Ω
- Ocho resistencias de 330 Ω
- Dos dipswitch de 8 interruptores
- Una barra de 8 LED's
- Un cristal 20 MHz
- 17 botones
- Dos capacitores 22 pF

- ## Equipo

- Página 26

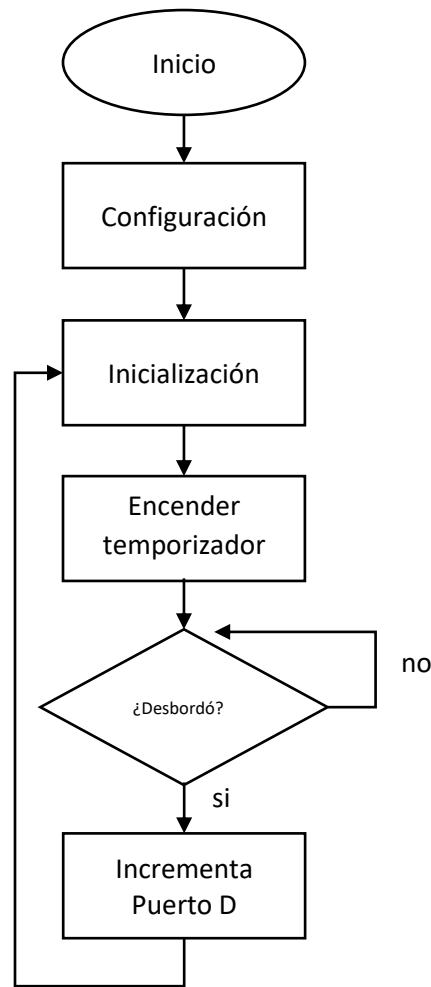


Figura 15. Diagrama de flujo para el temporizador.

A continuación, la Tabla 4 muestra un extracto del código para lograr el objetivo de la actividad. Se han colocado instrucciones, registros y bits requeridos para el funcionamiento del temporizador y los bucles de espera.

Tabla 5. Programa en lenguaje ensamblador sugerido.

Etiquetas	Instrucción	Argumento	Comentario de programador
Config	MOVLW	b'00000111'	Coloca un número al acumulador W
	MOVWF	OPTION_REG	El valor en acumulador w se envía al registro "OPTION_REG" el cual está encargado de controlar el comportamiento del temporizador
Inicial	MOVLW	0xFF	Coloca un número al acumulador W
	MOVWF	TMR0	El valor en acumulador w se envía al registro TMR0 como valor inicial al conteo del temporizador
	BCF	INTCON, T0IF	Al apagar el bit en este registro, se habilita el temporizador

Bucle	BTFS	INTCON, T0IF	Esta instrucción pregunta si el temporizador ya terminó de contar, hay dos posibles respuestas:
	GOTO	Bucle	No ha terminado. El programa se dirige hacia la etiqueta de "Bucle"
	INCF	PORTD, F	Si, ya terminó. En este momento se incrementa en uno al valor del puerto D de salida y se verá reflejado en los LED
	GOTO	Inicial	Vuelve a empezar

- I. Se alambra un circuito de la Figura 14.
- II. Se realiza la programación en lenguaje de ensamblador o equivalente. Se sugiere seguir el diagrama de flujo de la Figura 15.
- III. Se ha colocado un programa en la tabla 5, pero puede cambiar si el modelo de PIC cambia o si se usa otro microcontrolador.
- IV. Se prueba el circuito.
- V. Se hacen las correcciones necesarias, en caso de requerirlas
- VI. Se apaga la fuente.

9.5. Bibliografía recomendada

- Fernando Valdés Pérez. Microcontroladores, Fundamentos y aplicaciones PIC, Alfaomega, México, 2007.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

10. Práctica 10. Microcontroladores: Convertidor analógico-digital

10.1. Introducción

Los microcontroladores son dispositivos programables que contienen en su arquitectura los tres componentes básicos de un microcomputador: el procesador, la memoria y una interfaz de entradas y salidas [2]. Durante estas prácticas se irán usando los distintos dispositivos o periféricos característicos de los micros. En este manual se usará *lenguaje ensamblador* y se usan ejemplos de la programación de un PIC de gama media (PIC16F887). Sin embargo, se pueden usar equivalencias.

El convertidor analógico digital (ADC) es otro periférico de los microcontroladores muy común y útil para captar información del medio externo. Si bien existen los circuitos integrados que realizan la misma función, tenerlo dentro de un microcontrolador abre la posibilidad de manejar la información tan rápido sea posible y se disminuye el hardware adicional. Además, mediante el software y un multiplexor, es posible tener alrededor de catorce canales de lectura analógica (al menos en el PIC propuesto). Esto hace que con el ADC del microcontrolador se puede usar tanto para la captación de información como la toma de decisiones o control en un sistema digital [2,3].

10.2. Objetivos

- Usar un convertidor analógico digital de un microcontrolador.
- El alumno convertirá el voltaje de entrada y lo verá reflejado en una barra de luces LED.

10.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Un PIC16F887
- Una resistencia de 1k
- Un botón pulsador
- Una barra de 8 LED's
- Un cristal 20 MHz
- Dos capacitores 22 pF

- Una resistencia variable (potenciómetro) 10 K
- Una protoboard

Equipo

- Fuente de poder

10.4. Descripción y desarrollo de la práctica

Para esta práctica se sugiere implementar en un PIC16F887 la lectura de una entrada analógica usando un convertidor analógico digital, se usará la señal generada por un potenciómetro (configurado como divisor de voltaje), en cual estará conectado al pin RA0 como se observa en la Figura 16. Posteriormente el valor binario de la lectura de se despliega en el puerto D. Cabe recordar que el resultado de 10 bits se observará recortado a 8 bits, debido a la naturaleza del programa, pero se puede aumentar mediante software.

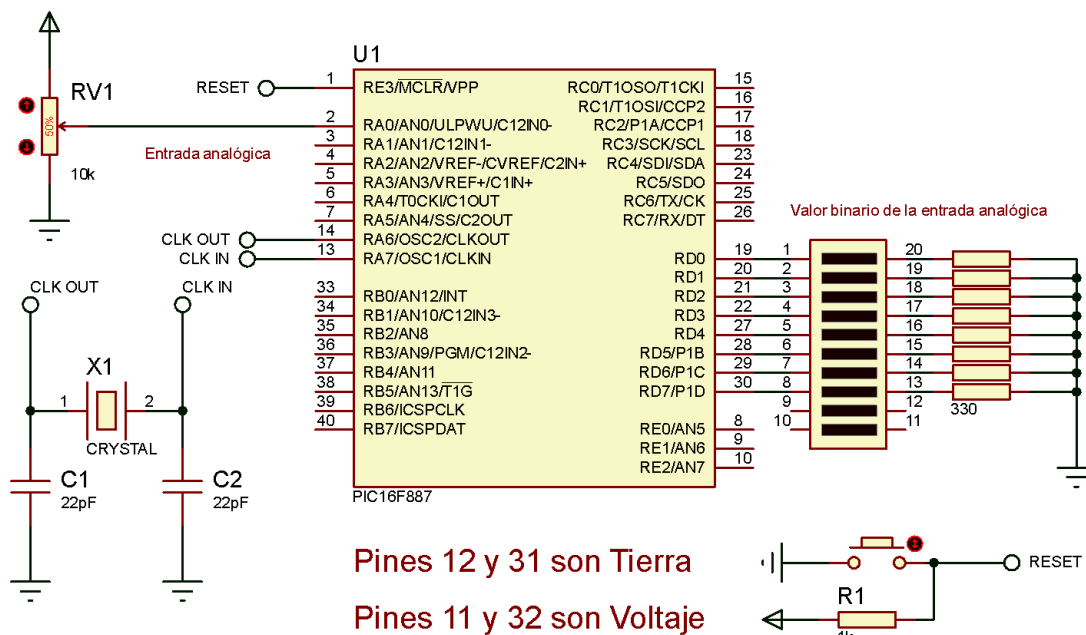


Figura 16. Esquemático de un microcontrolador y conexiones requeridas.

Para su implementación se requiere un procedimiento enumerado como sigue:

1. Configurar puertos de entrada y de salida
2. Configurar el módulo ADC
 - a) Seleccionar el reloj de conversión

- b) Configurar el voltaje de referencia
- c) Seleccionar el canal de entrada
- d) Seleccionar el formato de resultado
- e) Encender el módulo ADC
3. Configurar la interrupción (opcional)
4. Esperar el tiempo de adquisición (generar un retraso de tiempo con temporizadores o equivalente)
5. Iniciar la conversión con el bit (Se usan bits de registros)
6. Esperar un tiempo de conversión (bucle de espera)
7. Leer el resultado (otras acciones son: usar, comparar, desplegar, etc.)
8. Limpiar banderas (Si se usó interrupción)

En el siguiente código (Tabla 9) se incluyen los pasos anteriores, excepto el uso de la interrupción. Además, se identifica la configuración y líneas de código sugeridas.

Tabla 6. Programa en lenguaje ensamblador sugerido.

Etiquetas	Instrucción	Argumento	Comentario de programador
Config	MOVLW	b'00000000'	Se inicializa el ADC y puerto analógico canal cero
	MOVWF	ADCON1	
	MOVLW	b'10000001'	
	MOVWF	ADCON0	
LOOP	CALL	RETRASO	Se llama una subrutina de "retraso"
	BSF	ADCON0, GO_DONE	Inicia la adquisición
	BTFSZ	ADCON0, GO_DONE	Revisa si la lectura/ adquisición se completó
	GOTO	\$-1	Regresa a la línea anterior si no se completó
	MOVF	ADRESH, w	
	MOVWF	PORTD	se despliegan 8 bits de la lectura en el puerto d
	GOTO	LOOP	
RETRASO	DECFSZ	Delay1, F	Subrutina de retraso
	GOTO	RETRASO	
	DECFSZ	Delay2, F	
	GOTO	RETRASO	
	RETURN		

- I. Se alambra un circuito con las características requeridas, se sugiere el de la Figura 16
- II. Se realiza la programación, se sugiere tomar como base el que está en la Tabla 6.

- III. Se prueba el circuito, moviendo el potenciómetro, con lo que variará la secuencia de bits encendidos en la barra de LED
- IV. Se hacen las correcciones necesarias, en caso de requerirlas

10.5. Bibliografía recomendada

- Fernando Valdés Pérez. Microcontroladores, Fundamentos y aplicaciones PIC, Alfaomega, México, 2007.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

11. Práctica 11. Microcontroladores: Interrupciones

11.1. Introducción

Los microcontroladores son dispositivos programables que contienen en su arquitectura los tres componentes básicos de un microcomputador: el procesador, la memoria y una interfaz de entradas y salidas [2]. Durante estas prácticas se irán usando los distintos dispositivos o periféricos característicos de los micros. En este manual se usará *lenguaje ensamblador* y se usan ejemplos de la programación de un PIC de gama media (PIC16F887). Sin embargo, se pueden usar equivalencias.

En todos los microcontroladores existen las “interrupciones” que pueden explicarse como *desvíos en el flujo de control del programa* que son originados asincrónicamente por diversos eventos que no dependen del programador, es decir, pueden ocurrir en cualquier momento debido a un evento.

Las interrupciones ocurren por eventos externos (p.ej. interrupción por flanco en una entrada, cambio de un puerto o recepción de un dato) o eventos internos (p. ej. desbordamiento de un temporizador, terminación del conversor análogo a digital) [2,3].

En esta práctica se usará la “Interrupción externa” localizada físicamente en un pin del puerto B (*RB0/INT*).

11.2. Objetivos

- Usar una de las fuentes de interrupción del microcontrolador.
- Implementar un circuito que aproveche la interrupción externa.
- El alumno observará el incremento o decremento en el valor binario en un puerto de salida digital cuando se aplica la interrupción externa.

11.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Un PIC16F887
- Tres resistencias de 1k Ω
- Tres botones pulsadores

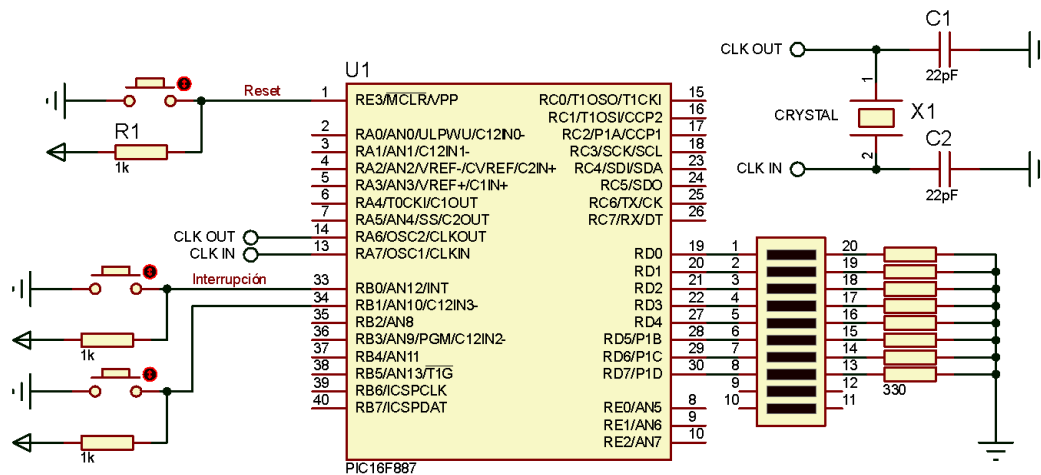
- Un cristal 20 MHz
- Dos capacitores de 22pF
- Una protoboard
- Un metro de cable de calibre 22

Equipo

- Fuente de poder

11.4. Descripción y desarrollo de la práctica

Se propone el siguiente circuito para la realización de la práctica, pero se recuerda que puede usarse otros microcontroladores.



Pines 12 y 31 son Tierra

Pines 11 y 32 son Voltaje

Figura 17. Esquemático de un microcontrolador y conexiones requeridas.

En la siguiente Tabla 7 se propone un programa que emplea la interrupción externa del PIC16F887, y para activarla se usa un botón de entrada en el pin cero del puerto B. También se usa un botón en el pin uno del mismo puerto para determinar si el valor binario del puerto D incrementa (si el botón está encendido) o si se decrementa (si el botón está apagado). En el siguiente programa de la Tabla 7 se identifica la configuración y líneas de código sugeridas.

Tabla 7. Programa en lenguaje ensamblador sugerido.

Etiquetas	Instrucción	Argumento	Comentario de programador
Config	MOVLW	b'11000000'	Se configura la interrupción externa y los permisos para habilitar las interrupciones en general
	MOVWF	OPTION_REG	
	MOVLW	B'11010000'	

	MOVWF	INTCON	
Bucle			Bucle infinito
	GOTO	Bucle	Aquí es donde espera la interrupción el programa
IntExt	BTFSS	PORTB, 1	En esta línea empieza la interrupción
	GOTO	incrementa	Se toma la decisión si el programa incrementa o decrementa el valor del puerto D
	GOTO	decrementa	
incrementa	INCF	PORTD, F	INCREMENTA el registro del puerto D
	BCF	INTCON, INTF	Limpiar bandera de interrupción
	RETFIE		Salir de interrupción
decrementa	DECF	PORTD, F	DECREMENTA el registro del puerto D
	BCF	INTCON, INTF	Limpiar bandera de interrupción
	RETFIE		Salir de interrupción

- I. Se alambra un circuito con las características previamente mencionadas
- II. Se realiza la programación en lenguaje de ensamblador.
- III. Se prueba el circuito.
- IV. Se hacen las correcciones necesarias, en caso de requerirlas.

11.5. Bibliografía recomendada

- Fernando Valdés Pérez. Microcontroladores, Fundamentos y aplicaciones PIC, Alfaomega, México, 2007.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

12. Práctica 12. Microcontroladores: Modulación de ancho de pulso

12.1. Introducción

Los microcontroladores son dispositivos programables que contienen en su arquitectura los tres componentes básicos de un microcomputador: el procesador, la memoria y una interfaz de entradas y salidas [2]. Durante estas prácticas se irán usando los distintos dispositivos o periféricos característicos de los micros. En este manual se usará *lenguaje ensamblador* y se usan ejemplos de la programación de un PIC de gama media (PIC16F887). Sin embargo, se pueden usar equivalencias.

La modulación por ancho de pulso (siglas en inglés de *pulse-width modulation*, PWM), es una herramienta importante en la electrónica digital, con el PWM es posible variar la intensidad lumínica de un led, la velocidad de un motor, o la posición angular de un servomotor, solo por mencionar los ejemplos más comunes. Consta de una señal digital cuadrada cuya característica principal es que el usuario le cambia el tiempo de prendido y apagado que tiene la señal, sin modificar su frecuencia, y a esto se le denominará el ciclo de trabajo (*duty*) expresado en porcentaje, por ejemplo, si el tiempo de prendido de la señal digital es del 75% del periodo, el 25 % restante estará apagado [2,3].

Cada microcontrolador tiene un modo de programación y de operación alrededor de este periférico que normalmente está asociado a pines y registros específicos dentro del microcontrolador en cuestión.

12.2. Objetivo

- Diseñar e implementar la modulación por ancho de pulso con un PIC, conocer su funcionamiento y las entradas que lo comprenden.

12.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Un PIC16F887
- Una resistencia variable de 10 k Ω

- ## Equipo

- Página 37

Tabla 8. Programa en lenguaje ensamblador sugerido.

Etiquetas	Instrucción	Argumento	Comentario de programador
Config	MOVLW	B'01111100'	Configuración del periodo del PWM
	MOVWF	PR2	
	BCF	TRISC,2	PC2 es salida
	CLRF	TRISD	PD es salida
	MOVLW	0xFF	PA entrada
	MOVWF	TRISA	
	MOVLW	B'00000000'	Configura ADC: resultado a la izquierda, Vref+ =Vcc, Vref- = GND
	MOVWF	ADCON1	Fosc/32; Canal analógico 0, ADC habilitado
	MOVLW	b'10000001'	
	MOVWF	ADCON0	
	MOVLW	0x07	Postescala 1/16; timer2 ON; Preescala 16
	MOVWF	T2CON	
	MOVLW	B'00011100'	Salida simple de PWM
	MOVWF	CCP1CON	
Bucle	CALL	DELAY	Subrutina de retraso
	BSF	ADCON0, GO_DONE	Inicio de conversión
	BTFSC	ADCON0, GO_DONE	
	GOTO	\$_1	
	MOVF	ADRESH, w	
	MOVWF	CCPR1L	El resultado de ADC se asigna al PWM
	GOTO	Bucle	
DELAY	MOVLW	.2	Subrutina de retraso
	MOVWF	Delay1	
	MOVLW	.2	
	MOVWF	Delay2	
DelayLoop	DECFSZ	Delay1, f	
	GOTO	DelayLoop	
	DECFSZ	Delay2, f	
	GOTO	DelayLoop	
	RETURN		

- I. Se alambra un circuito con las características previamente mencionadas (Figura 18).
- II. Se realiza la programación en lenguaje de ensamblador (Tabla 8)
- III. Se prueba el circuito.
- IV. Se hacen las correcciones necesarias, en caso de requerirlas

12.5. ***Bibliografía recomendada***

- Fernando Valdés Pérez. Microcontroladores, Fundamentos y aplicaciones PIC, Alfaomega, México, 2007.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

13. Práctica 13. Microcontroladores: Comunicación serial

13.1. Introducción

Los microcontroladores son dispositivos programables que contienen en su arquitectura los tres componentes básicos de un microcomputador: el procesador, la memoria y una interfaz de entradas y salidas [2]. Durante estas prácticas se irán usando los distintos dispositivos o periféricos característicos de los micros. En este manual se usará *lenguaje ensamblador* y se usan ejemplos de la programación de un PIC de gama media (PIC16F887). Sin embargo, se pueden usar equivalencias.

La comunicación serial en los microcontroladores es un periférico útil para interconectar dispositivos y que se pueda compartir información e incluso compartir funciones de procesamiento, por ejemplo, si se usa una computadora, una parte del programa de control estará en la computadora y la otra estará en el microcontrolador [2].

La comunicación serial utiliza tres líneas: una para recibir los datos (**Rx**), otra para transmitir los datos (**Tx**) y la línea común de tierra (**GND**). En el caso específico para el PIC16F887 las líneas TX y RX corresponden a los pines 25 y 26 (Puerto C pin 6 y pin 7 respectivamente). Para cualquier otro microcontrolador, es necesario consultar estos pines en su hoja de especificaciones.

13.2. Objetivo

- Implementar una comunicación serial con el periférico USART del microcontrolador.

13.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Un PIC16F887
- Un cristal 20 MHz
- Dos capacitores 22 nF
- Dos resistencias de 1k Ω

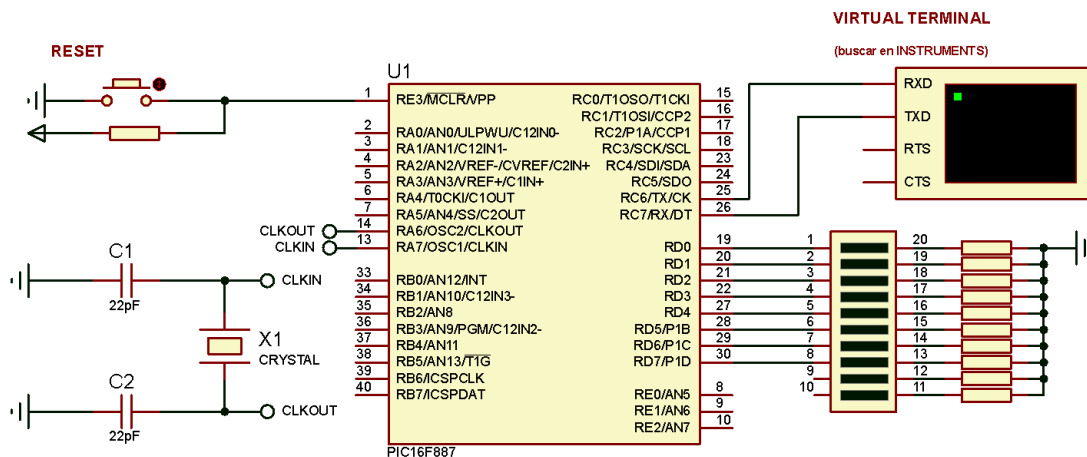
- Un botón pulsador
- Un convertidor serial a USB
- Una protoboard
- Un metro de cable de calibre 22

Equipo

- Fuente de poder
- Computadora

13.4. Descripción y desarrollo de la práctica

Se propone el siguiente circuito, donde el microcontrolador está usando una interfaz de comunicación serial a USB (de la computadora). En la Figura 19 señala una Terminal Virtual que se emplea durante la simulación, sin embargo, en la realidad se tienen dispositivos comerciales muy comunes que hacen la función de conversión del protocolo de comunicación serial a un puerto de la computadora.



Pines 12 y 31 son Tierra

Pines 11 y 32 son Voltaje

Figura 19. Esquemático de un microcontrolador y conexiones requeridas.

El programa propuesto realiza dos funciones a la vez, la primera es recibir desde la computadora (puerto USB) un carácter codificado en 8 bits, para esto se ocupa el teclado y una interfaz para envío de información desde la computadora. Posteriormente, el programa guarda esa palabra de 8 bits en el registro de recepción (Rx) y la transmite con una subrutina que envía la misma palabra binaria

de 8 bits usando un registro de transmisión (Tx). En el siguiente programa de la Tabla 9 se identifica la configuración y líneas de código sugeridas.

Tabla 9. Programa en lenguaje ensamblador sugerido.

Etiquetas	Instrucción	Argumento	Comentario de programador
#DEFINE	FOSC	.20000000	Definición de variables para la transmisión
#DEFINE	BAUD	.19200	
#DEFINE	EQUIS	FOSC/(.16*BAUD)-1	
Principal y configuración			
Config	CLRF	TRISD	PD es salida
	MOVLW	B'00100100'	Configura la transmisión de 8 bits, permite la transmisión recepción en modo asíncrono, opera a alta velocidad
	MOVWF	TXSTA	
	BCF	TRISC,6	Habilita la transmisión de datos
	BSF	TRISC,7	Habilita la recepción de datos
	MOVLW	EQUIS	Velocidad de transmisión
	MOVWF	SPBRG	Velocidad de transmisión Baud Rate=9600
	BSF	PIE1, RCIE	Permiso de interrupción por recepción de datos
	MOVLW	B'10010000'	Permiso para utilización de puerto serial
	MOVWF	RCSTA	
	BSF	INTCON, PEIE	Permiso de interrupción por periférico
	BSF	INTCON, GIE	Permiso global de interrupciones
	BCF	PIR1, RCIF	Limpia la bandera de interrupción por recepción
	BCF	PIR1, TXIF	Limpia la bandera de interrupción por transmisión
Loop	NOP		
	GOTO	Loop	Espera
Interrupción			
INTER	MOVF	RCREG, W	W=RCREG(Registro de recepción de datos)
	MOVWF	PORTD	
	CALL	TRANS	Llama a la subrutina de transmisión
	BCF	PIR1, RCIF	Pone en cero la bandera por permiso de recepción de datos
	RETFIE		Regresa de la interrupción
Subrutina			
TRANS	MOVWF	TXREG	Carga al registro de transmisión lo que hay en el registro de recepción (o envía)
	BTFS	PIR1, TXIF	¿Está activada la bandera de transmisión?
	GOTO	\$_1	No, pregunta nuevamente
	BCF	PIR1, TXIF	Sí, limpia la bandera de transmisión (se ha enviado)
	RETURN		Regresa del subprograma

- I. Se alambra un circuito con las características previamente mostradas en la Figura 19.
- II. Se realiza la programación en lenguaje de ensamblador, se puede tomar como base las líneas de programación establecidas en la Tabla 9.
- III. Se prueba el circuito, usando el teclado de la computadora y una terminal de envío y recepción de datos por el puerto USB correspondiente.

IV. Se hacen las correcciones necesarias, en caso de requerirlas

13.5. Bibliografía recomendada

- Fernando Valdés Pérez. Microcontroladores, Fundamentos y aplicaciones PIC, Alfaomega, México, 2007.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

14. Proyecto integrador 1. Circuitos Secuenciales

14.1. Introducción

Como parte del proyecto se busca integrar todos los temas vistos previamente en un circuito simulando un semáforo de tres colores verde, amarillo y rojo, en el cual dependiendo del estado de un botón será la secuencia que se ejecute.

Cuando el botón se encuentre inactivo, la secuencia de los LED's será verde durante cuatro segundos, amarillo por dos segundos y rojo por dos segundos, mientras que, cuando se active el botón la secuencia cambiará y empezará rojo durante dos segundos, amarillo por dos segundos y verde durante cuatro segundos.

14.2. Objetivos

- Integrar los conocimientos de circuitos combinacionales y secuenciales
- El alumno diseñará un circuito digital desde el planteamiento del problema hasta la realización del prototipo electrónico real.

14.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Tres resistencias de 330Ω
- Una resistencia de $1k\Omega$
- Un botón
- Dos circuitos integrados 74LS74
- Un circuito integrado 74LS48
- Una compuerta lógica "NOT" (74LS04)
- Tres compuertas lógicas "AND" (74LS08)
- Una compuerta lógica "OR" (74LS32)
- Una compuerta lógica "NOR" (74LS86)
- Tres LED's (uno verde, uno amarillo y uno rojo)
- Un display de 7 segmentos de cátodo común
- Dos protoboard (tableta de pruebas)
- Un metro de cable de calibre 22

Equipo

- Un multímetro
- Una pinza de corte diagonal
- Una pinza de punta

14.4. Descripción y desarrollo de la práctica

Para lograr la secuencia deseada, primero se realizará una tabla "de estado actual y siguiente" la cual nos ayudará a visualizar la secuencia a la que se quiere llegar y con ayuda de esta se podrá realizar las diversas tablas de verdad.

Tabla 10. Representación de los estados del sistema para el diseño de las funciones booleanas.

X	Estado actual			Estado siguiente			Rojo	Amarillo	Verde
0	0	0	0	0	0	1			1
0	0	0	1	0	1	0			1
0	0	1	0	0	1	1			1
0	0	1	1	1	0	0			1
0	1	0	0	1	0	1		1	
0	1	0	1	1	1	0		1	
0	1	1	0	1	1	1	1		
0	1	1	1	0	0	0	1		
1	0	0	0	0	0	1	1		
1	0	0	1	0	1	0	1		
1	0	1	0	0	1	1		1	
1	0	1	1	1	0	0		1	
1	1	0	0	1	0	1			1
1	1	0	1	1	1	0			1
1	1	1	0	1	1	1			1
1	1	1	1	0	0	0			1

A continuación, se presentan las tablas de verdad para poder sacar las funciones booleanas necesarias para lograr la secuencia del semáforo deseada:

Tabla 11. Mapa de Karnaugh para DC

	DC			
$XC \setminus BA$	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{X}\bar{C}$			1	
$\bar{X}C$	1	1		1
$\bar{X}C$	1	1		1
$X\bar{C}$			1	
$DC = C\bar{B} + \bar{C}BA + CB\bar{A}$				
$DC = AB + C$				

Tabla 12. Mapa de Karnaugh para DB

	DB			
$XC \setminus BA$	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{X}\bar{C}$		1		1
$\bar{X}C$		1		1
$\bar{X}C$		1		1
$X\bar{C}$		1		1
$DB = \bar{B}A + B\bar{A}$				
$DB = B \otimes A$				

Tabla 13. Mapa de Karnaugh para DA

	DA			
$XC \setminus BA$	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{X}\bar{C}$	1			1
$\bar{X}C$	1			1
$\bar{X}C$	1			1
$X\bar{C}$	1			1
$DA = \bar{B}\bar{A} + B\bar{A}$				
$DA = \bar{A}$				

Tabla 14. Mapa de Karnaugh para Rojo

	D Rojo			
$XC \setminus BA$	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{X}\bar{C}$				
$\bar{X}C$			1	1
$\bar{X}C$				
$X\bar{C}$	1	1		
$D \text{ Rojo} = X\bar{C}\bar{B} + \bar{X}CB$				

Tabla 14. Mapa de Karnaugh para Amarillo

	D Amarillo			
$XC \setminus BA$	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{X}\bar{C}$				
$\bar{X}C$	1	1		
$\bar{X}C$			1	1
$X\bar{C}$				
$D \text{ Amarillo} = \bar{X}C\bar{B} + XCB$				
$D \text{ Amarillo} = (X \odot B)C$				

Tabla 15. Mapa de Karnaugh para Verde

	D Verde			
$XC \setminus BA$	$\bar{B}\bar{A}$	$\bar{B}A$	BA	$B\bar{A}$
$\bar{X}\bar{C}$	1	1	1	1
$\bar{X}C$				
$\bar{X}C$	1	1	1	1
$X\bar{C}$				
$D \text{ Verde} = \bar{X}\bar{C} + X\bar{C}$				
$D \text{ Verde} = (X \odot C)$				

Una vez que ya se definieron las funciones a realizar se realiza el alambrado del circuito.

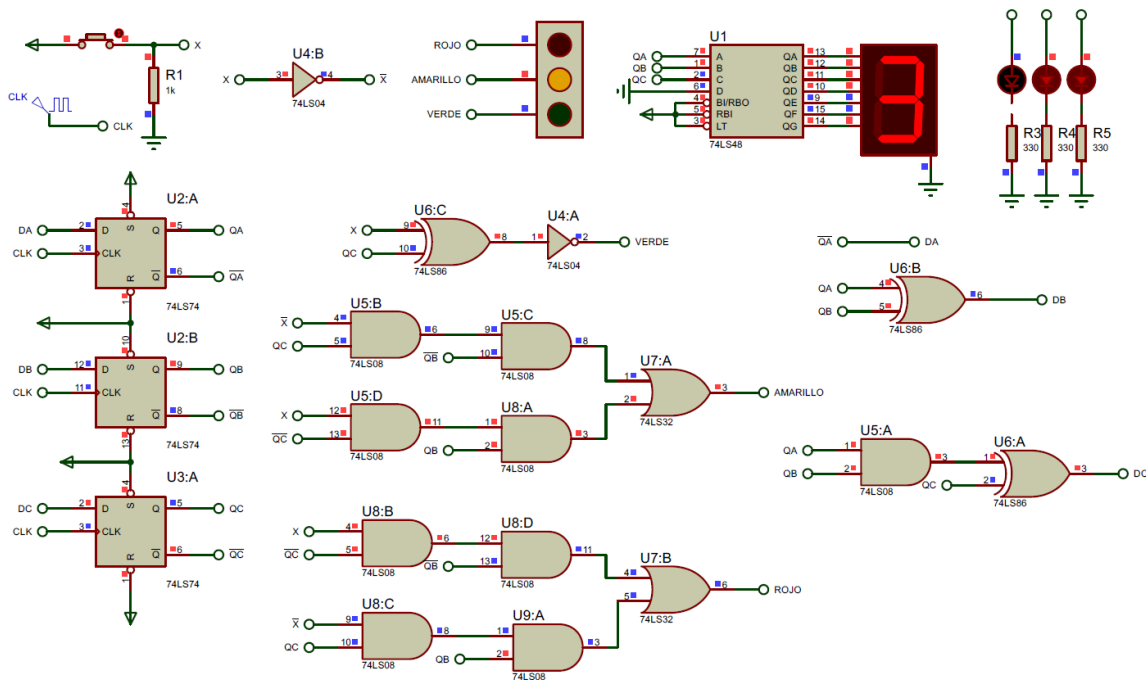


Figura 20: Esquema del circuito para alambrear

- I. Se realiza una tabla de estado actual y estado siguiente para la cual se cumpla la secuencia del semáforo
- II. Con base a las tablas de verdad de las salidas para poder obtener las funciones booleanas.
- III. Realizar una simulación en computadora para poder comprobar el correcto funcionamiento de las funciones booleanas
- IV. Una vez que se comprobaron las funciones en la simulación, alambrear el circuito
- V. Encender la fuente de poder y regularla a 5V
- VI. Comprobar el funcionamiento de la secuencia con ayuda del display y de los LED's, así como el cambio de esta al presionar el botón
- VII. Apagar la fuente de poder

14.5. Bibliografía recomendada

- Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

15. Proyecto 2. Control de servomotores y señales analógicas

15.1. Introducción

Los servomotores son actuadores que ayudan para el movimiento de articulaciones u otros mecanismos en proyectos de órtesis y prótesis. Los servomotores tienen un control basado en Modulación de ancho de pulso (Práctica 12) Dependiendo de los sensores y del tipo de proyecto, se pueden encontrar variaciones del sistema propuesto en este proyecto. Para lograr la integración de conocimientos vistos en este manual, se eligieron sensores de flexión que serán leídos con los canales analógicos (Práctica 10) de algún microcontrolador. Posteriormente se moverá el ángulo de los servomotores de acuerdo con el ángulo leído en los sensores de flexión. Adicionalmente se usará la comunicación serial (Práctica 13) para transmitir información relevante a la computadora.

15.2. Objetivos

- Integrar el uso de sensores, actuadores y microcontrolador para generar un sistema con movimiento controlado.
- El alumno realiza un sistema para mover servomotores con señales de entrada analógicas.

15.3. Recursos de espacio, materiales y equipo

Espacio

- Laboratorio de electrónica con mesas con superficies dieléctricas

Material

- Un microcontrolador (se sugiere una tarjeta de desarrollo que incluya los elementos como el reloj de cuarzo, capacitores, reguladores y comunicación serial integrada)
- Tres sensores de flexión
- Tres servomotores
- Una protoboard
- Un metro de cable de calibre 22

Equipo

- Fuente de poder

15.4. Descripción y desarrollo de la práctica

Se propone el siguiente sistema.

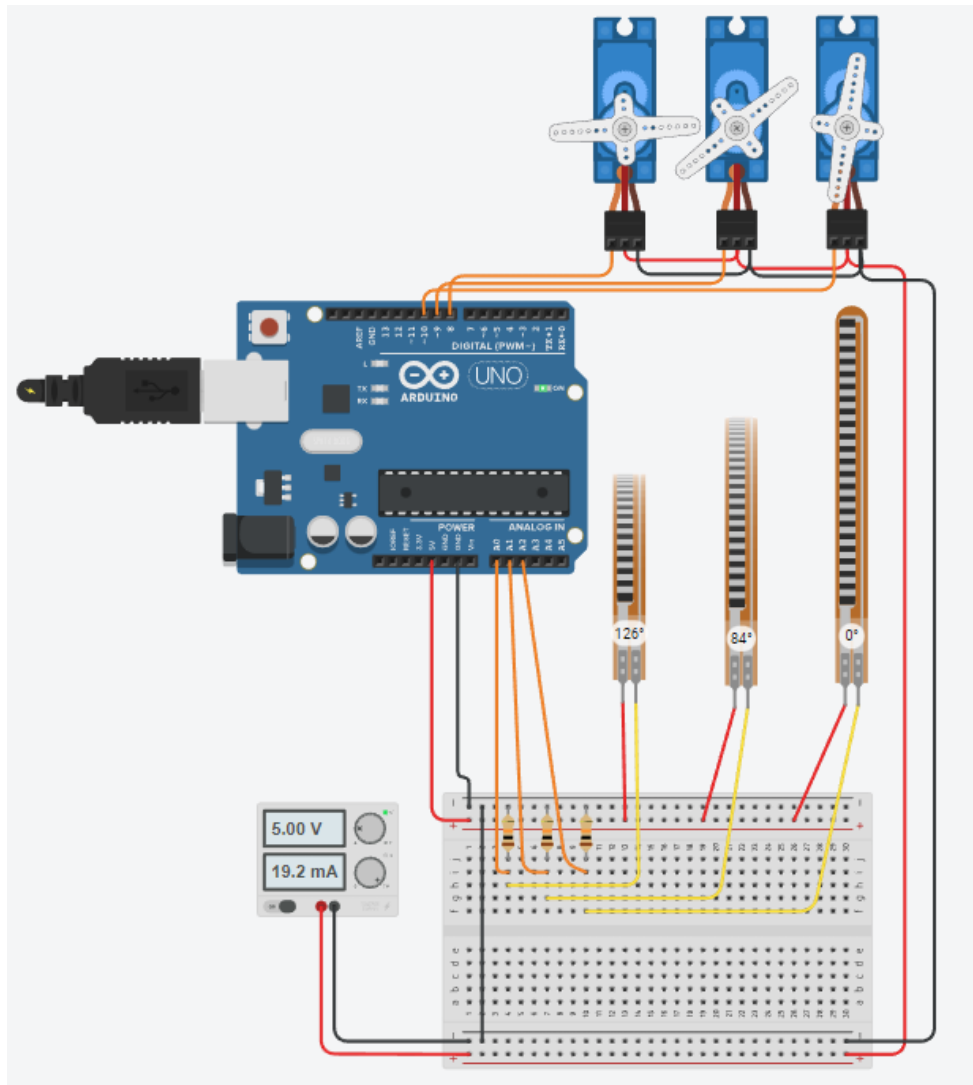


Figura 21. Esquema del circuito propuesto es de elaboración propia y fue dibujado en *Tinkercad* de Autodesk, cuyo uso es libre para la comunidad UNAM.

Se observan tres distintos ángulos en los sensores de flexión que modifican el ángulo de salida de cada servomotor ubicado arriba del sensor que lo modifica. Adicionalmente se ha propuesto que los valores obtenidos del sensor y que serán reflejados en los servomotores, se puedan transmitir a la computadora por el puerto serie.

Tabla 16. Programa propuesto en la plataforma de Arduino.

```

#include <Servo.h>
Servo servo0; Servo servo1; Servo servo2;
int sensor0 = 0; int sensor1 = 0; int sensor2 = 0;
void setup()
{ //configuración
  pinMode(A0, INPUT); pinMode(A1, INPUT); pinMode(A2, INPUT);
  servo0.attach(8); servo1.attach(9); servo2.attach(10);
  Serial.begin(9600);
}
// programa principal
void loop() {
  // lectura
  sensor0 = analogRead(A0);
  sensor1 = analogRead(A1);
  sensor2 = analogRead(A2);
  // ajustar los valores de entrada a los ángulos del servomotor
  sensor0 = map(sensor0, 256, 59, 0, 180);
  sensor1 = map(sensor1, 256, 59, 0, 180);
  sensor2 = map(sensor2, 256, 59, 0, 180);
  // enviar los ángulos de salida a los servomotores
  servo0.write(sensor0);
  servo1.write(sensor1);
  servo2.write(sensor2);
  // transmisión de los valores al puerto serie de la computadora
  Serial.print(sensor0);
  Serial.print(",");
  Serial.print(sensor1);
  Serial.print(",");
  Serial.println(sensor2);

  delay(10);
}

```

- I. La propuesta de los dispositivos a usar se encuentra en la Figura 21.
- II. Se realiza la programación del microcontrolador. Se ha propuesto una secuencia de control que se puede encontrar en la Tabla 16.
- III. Se prueba el proyecto físicamente.

15.5. Bibliografía recomendada

- Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.
- William Bolton. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.

16. Referencias del manual

- [1] Floyd, Thomas L., Fundamentos de sistemas digitales, 9ª edición, Pearson, España, 2006.
- [2] Valdés Pérez, Fernando. Microcontroladores, Fundamentos y aplicaciones PIC, Alfaomega, México, 2007.
- [3] Bolton, William. Mecatrónica: sistemas de control electrónico en la ingeniería mecánica y eléctrica, 6 Ed. Alfaomega, México, 2017.