

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

PROBADOR DE COMPONENTES ELECTRONICOS

KATHERINE ANDREA ROJAS ALZATE

INGENIERIA ELECTROMECHANICA

Asesor:

MSc. Carlos Mario Londoño Parra

INSTITUTO TECNOLÓGICO METROPOLITANO

2017

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

Se desea desarrollar un sistema para controlar dispositivos electrónicos, que consta de una tarjeta electrónica con un circuito embebido que interactúa con señales eléctricas a través de entradas y salidas, de manera que permita evaluar los diferentes dispositivos electrónicos.

Además, su manual de instrucciones para el manejo del sistema o módulo y la clasificación de las unidades que se pueden chequear con esta tarjeta y especificar la conexión de las mismas.

El presente trabajo es el resultado de una inquietud, al observar que éstos acudían a los laboratorios y presentaban constantemente inconvenientes en sus montajes, pues descubrían que los componentes electrónicos presentaban averías, además, de que muchos no conocían la forma correcta e conectar dichos componentes.

Con el propósito de darle al estudiante mayor oportunidad de trabajo. Habiendo realizado el trabajo con mucha dedicación y con un único ideal, lograr la meta que me he propuesto.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

Quiero agradecer a Dios sobre todas las cosas, a mis padres Wilson Rojas (QED) y Adriana Álzate, quienes siempre me apoyaron y creyeron en mí; a ellos debo mi carrera, mis valores, el apoyo moral y económico que siempre me brindaron.

Agradezco a mi asesor y profesor Carlos Mario Londoño, quien creyó en mí y en este proyecto y a pesar de las circunstancias personales se dieron las cosas de la mejor manera para llevar a buen término esta meta, sin su ayuda esto no habría sido posible.

Agradezco también al respetado profesor Dr. Carlos Alberto Acevedo, quien siempre estuvo conmigo apoyándome y motivándome para que sacara este logro profesional y obtuviera mi título como ingeniera Electromecánica.

Agradezco a mi excelente amigo y próximamente colega Víctor Mario Oquendo Correa, quien desde un principio fue el que impulso para materializar esta idea y así subsanar un requerimiento del área de laboratorios, por ende, siempre quiso que yo dejara un buen legado, además de brindarme la posibilidad de trabajar de su mano con otros profesionales del área; siempre creyó en mí y hasta el último día me motivo y me ayudo en este proceso.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

ITM Instituto Tecnológico Metropolitano

V Voltios

A Amperios

I Corriente

CPU Unidad Central de Procesamiento.

PC Personal Computer.

UC Unidad de Control.

ALU Unidad Aritmético Lógica.

CM “coprocesador matemático” unidad de cálculo en coma flotante.

μC, UC o MCU Microcontrolador.

E/S Entradas/Salidas

PICmicro Peripheral Interface Controller / controlador de interfaz periférico.

RAM Random Access Memory / Memoria de acceso aleatorio.

ROM Read Only Memory / Memoria de solo lectura.

OTP One Time Programmable.

EPROM Erasable Programmable Read-Only Memory / ROM programable borrrable.

EEPROM o E²PROM Electrically Erasable Programmable Read-Only Memory / ROM programable y borrrable eléctricamente.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

D/A Digital/Analogo

LED light-emitting diode / diodo emisor de Luz

RC Resistencia tipo Condensador

LCD Liquid Crystal Display / Pantalla de cristal líquido

CMOS Metal Oxid Semiconductor Complement

JFET Junction Field Effect Transistor

MOSFET Metal-Oxide-Semiconductor FET

SCR Silicon Controlled Rectifier

TRIAC Triodo para Corriente Alterna

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

Asesor:	1
1. INTRODUCCIÓN	8
2. MARCO TEÓRICO	10
2.1. CAPITULO 1: COMPUERTAS LOGICAS	14
2.2. CAPITULO 2: TRANSISTORES	21
2.3. CAPITULO 3: DISPLAY 7 SEGMENTOS DE ANODO COMUN Y CATODO COMUN ..	29
2.4. CAPITULO 4: TEMPORIZADOR 555	32
3. METODOLOGÍA	34
4. RESULTADOS	38
5. CONCLUSIONES	41
6. RECOMENDACIONES Y TRABAJO FUTURO	43
REFERENCIAS	45
APÉNDICE	49
APENDICE A GASTOS.....	49
APENDICE B MIRCOCNTROLADORES, TEORIA Y PROGRAMA.....	51
APENDICE C MANUAL DE USO	67

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

INDICE DE TABLAS

Tabla 1. Microcontrolador (Aprendiendo Arduino , 2015)	10
Tabla 2 Estructura de un Microcontrolador (Microchip Technology, Inc., 2015) (MikroElektronika, 2017)	13
Tabla 3 Diferencias entre Compuertas TTL y C-MOS (Medina Salazar, 2015)	15
Tabla 4 Compuerta AND (Molina Marticorena J. , Compuertas Logicas, 2008).....	17
Tabla 5 Compuerta OR (Molina Marticorena J. , Compuertas Logicas, 2008)	17
Tabla 6. Compuerta NOT (Molina Marticorena J. L., 2008)	18
Tabla 7. Compuerta NAND (Molina Marticorena J. L., Compuertas Logicas, 2008)	18
Tabla 8. Compuerta NOR (Molina Marticorena J. L., Compuertas Logicas, 2008).....	19
Tabla 9. Compuerta XOR (Manuales electrónicos 4all, 2013).....	19
Tabla 10. Compuerta XNOR (Fernandez Perez, 2015)	20
Tabla 11. Tipos de Transistores (Jose , s.f.).....	21
Tabla 12. Transistor de Unión Bipolar o BJT. (www.neoteo.com, 2009)	22
Tabla 13. Transistor BJT vs Efecto Campo o FET.	24
Tabla 14. Transistor JFET Junction Field Effect Transistor (CircuitsToday, 2013)	25
Tabla 15. Transistor Mosfet (Oscar Lian, 2013)	25
Tabla 16. Fototransistor (Electronica PT, s.f.)	26
Tabla 17. SCR - Silicon Controlled Rectifier (Angulo & Barreto, 2013).....	27
Tabla 18. Transistor TRIAC (Poole, n.d.).....	28
Tabla 19. Display 7 Segmentos AC y CC (Electronic Tools, s.f.).....	29
Tabla 20. Pines, Números y Letras del Display (Electronic Tools, s.f.)	30
Tabla 21. Temporizador 555 (es.wikipedia.org, 2005).....	32
Tabla 22. Conexión Compuerta AND (Lorenzo Flores, 2015).....	35
Tabla 23. Conexión Transistor NPN y PNP (Comunidad Internacional de electronicos, 2012).....	36
Tabla 24. Visualización Display 7 Segmentos (Area tecnologia, s.f.).....	37
Tabla 25. Timer 555 (ElectroSome, 2014)	37

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

Debido a las necesidades encontradas en los laboratorios de circuitos eléctricos, electrónica y mecatrónica del ITM es aquí donde surge la idea de desarrollar un sistema de uso general y flexible en diferentes aplicaciones y componentes electrónicos, considerando un mayor potencial de ser eficientes, económicos y hacer que sea compatible con nuevos sistemas y nuevas tecnologías emergentes.

Se ha puesto énfasis en crear un módulo que utilice elementos sustanciales y fundamentales, así como se le ha dado importancia en generalizar los componentes más usados de tal manera que puedan adaptar a las necesidades del estudiante, así podrán utilizar componentes estándares y utilizados mundialmente.

Este sistema consta de un módulo con una tarjeta y un circuito embebido que incluye un microcontrolador y un software, para simular y chequear circuitos integrados (Compuertas TTL y CMOS, Transistores BJT, Mosfet, Jfet, Display AC y CC, Temporizador 555, Triac, SCR).

Este circuito embebido, permitirá evaluar los componentes más utilizados en los laboratorios de circuitos. La unidad cuenta con un sistema de visualización LED, se podrá interactuar por medio de una pantalla LCD digital de la cual se elegirán los componentes del menú y podrá indicar el estado de los elementos a chequear con la ayuda de un microcontrolador y un código fuente detectará qué tipo de componente se desea medir, así también se garantizará que, en efecto el estado de los componentes sea operativo.

Este módulo de chequeo, cuenta con aplicaciones variadas, por eso limita el uso del mismo a personas con un grado mayor de experiencia. Es por esto que este producto se recomienda para ser usado por personal con conocimientos en electrónica, para que puedan modelar y probar sus propios componentes, de tal manera que pudiera llegar a

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

convertirse en una herramienta para llevar a cabo el desarrollo del producto del estudiante, no obstante, se entrega un manual de usuario donde se encuentran las instrucciones de conexión y funcionamiento de cada componente. Por ende, el sistema tendrá sus limitaciones.

El manual de operación está estructurado por capítulos, el primer capítulo se habla de compuertas, familias, conexiones y características. El segundo capítulo se habla de transistores también exponiendo las diferentes referencias que hay en el mercado y sus respectivas funciones. En el tercero se habla de los display de 7 segmentos, sus características y funcionalidades. En el cuarto se habla acerca de temporizadores 555;

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEÓRICO

Microcontrolador

También conocido como μC , MC, MCU, básicamente es una computadora pequeña en un circuito integrado. Está compuesto por varios bloques funcionales para cumplir una o varias tareas específicas. Estos bloques son iguales a los de una computadora personal: procesador, memoria, entradas y salidas, etc. Es capaz de ejecutar instrucciones almacenadas en su memoria, están diseñados para aplicaciones embebidas, principalmente para automatizar el control de dispositivos como el motor de un automóvil, dispositivos médicos, controles remotos, juguetes, etc. Por su reducido tamaño, reducido consumo y bajo costo es la opción más adecuada para control digital. (Elsivier, 1ra edición 2007)

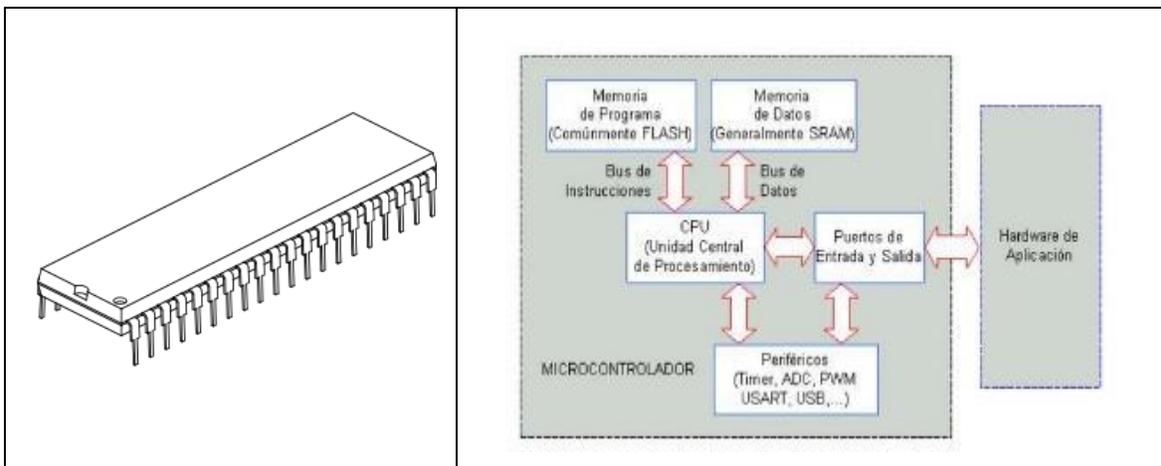


Tabla 1. Microcontrolador (Aprendiendo Arduino , 2015)

Memorias y puertos

La memoria en los microcontroladores debe estar ubicada dentro del mismo encapsulado, esto es así la mayoría de las veces, porque la idea fundamental es mantener el grueso de los circuitos del sistema dentro de un solo integrado.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En los microcontroladores la memoria no es abundante, aquí no encontrará Gigabytes de memoria como en las computadoras personales. Típicamente la memoria de programas no excederá de 16 K-localizaciones de memoria no volátil (flash o eprom) para contener los programas. (Boylestad R. L., Introducción al análisis de circuitos, 2011, 12a ed.)

- **Memoria RAM**

La memoria RAM está destinada al almacenamiento de información temporal que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. En el espacio de direcciones de memoria RAM se ubican además los registros de trabajo del procesador y los de configuración y trabajo de los distintos periféricos del microcontrolador. (ITPN)

- **Memoria ROM de máscara**

Conocida también como ROM (acrónimo en inglés de read-only memory), es un medio de almacenamiento utilizado en ordenadores y dispositivos electrónicos, que permite solamente la lectura de la información y no su escritura, independientemente de la presencia o no de una fuente de energía. Cabe recordar que esta es una memoria de acceso secuencial; Los datos almacenados en la ROM no se pueden modificar, o al menos no de manera rápida o fácil, las ROM más modernas, como EPROM y Flash EEPROM, efectivamente se pueden borrar y volver a programar varias veces, aun siendo descritos como "memoria de sólo lectura" (ROM). La razón de que se las continúe llamando así es que el proceso de reprogramación en general es poco frecuente, relativamente.

- **Memoria OTP**

One Time Programmable Este tipo de memoria, también es conocida como PROM o simplemente ROM.

Los microcontroladores con memoria OTP se pueden programar una sola vez, con algún tipo de programador

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- **Memoria EPROM**

Erasable Programmable Read Only Memory Los microcontroladores con este tipo de memoria son muy fáciles de identificar porque su encapsulado es de cerámica y llevan encima una ventanita de vidrio desde la cual puede verse la oblea de silicio del microcontrolador. Es una memoria reprogramable, pero antes se debe borrar, y para ello hay que exponerla a una fuente de luz ultravioleta.

- **Memoria EEPROM**

Electrical Erasable Programmable Read Only Memory. Fueron el sustituto natural de las memorias EPROM, la diferencia fundamental es que pueden ser borradas eléctricamente.

- **Memoria FLASH**

En el campo de las memorias reprogramables para microcontroladores, son el último avance tecnológico en uso a gran escala, y han sustituido a los microcontroladores con memoria EEPROM. Permite la lectura y escritura de múltiples posiciones de memoria en la misma operación. Gracias a ello, la tecnología flash, siempre mediante impulsos eléctricos, permite velocidades de funcionamiento muy superiores frente a la tecnología EEPROM primigenia, que sólo permitía actuar sobre una única celda de memoria en cada operación de programación. Se trata de la tecnología empleada en los dispositivos denominados memoria USB. (Salinas Delgado & Zúñiga Huertas, 2016)

PUERTOS

- También conocidas como entradas/salidas de propósito general.
- Generalmente agrupadas en puertos de 8 bits de longitud, permiten leer datos del exterior o escribir en ellos desde el interior del microcontrolador.

- Algunos puertos de E/S tienen características especiales que le permiten manejar salidas con determinados requerimientos de corriente, o incorporan mecanismos especiales de interrupción para el procesador.

El destino habitual es el trabajo con dispositivos simples como relés, LED, LCD, teclados matriciales, o cualquier otra cosa que se le ocurra al programador.

Como los microcontroladores no pueden tener infinitos pines, las E/S de propósito general comparten los pines con otros periféricos. Para usar un pin con cualquiera de las características a él asignadas debemos configurarlo mediante los registros destinados a ellos. (Rodolfo, Hernández Vázquez, & Resendiz Aguilar, 2010)

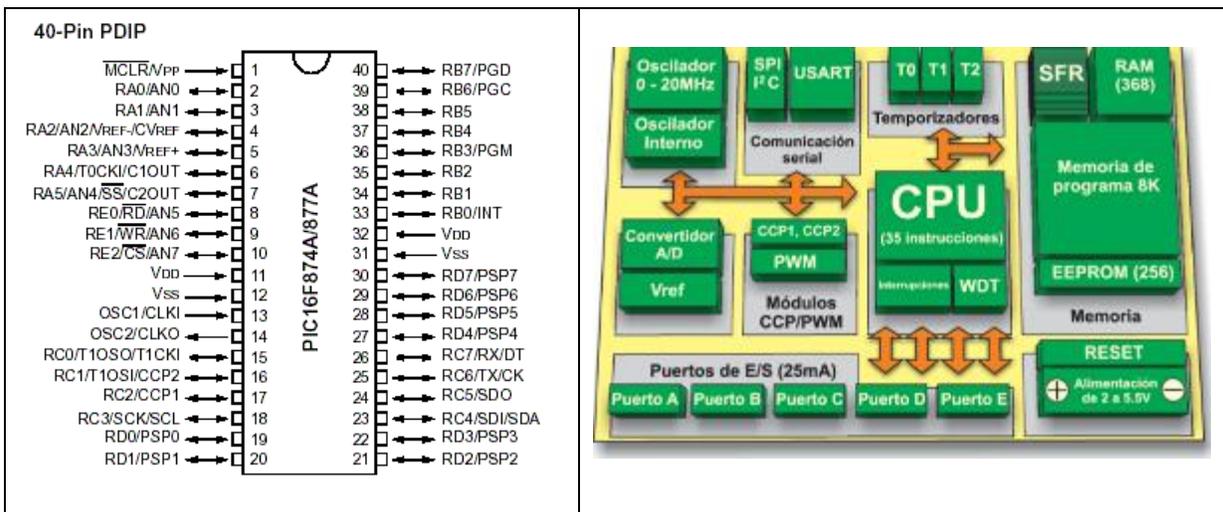


Tabla 2 Estructura de un Microcontrolador (Microchip Technology, Inc., 2015) (MikroElektronika, 2017)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.1. CAPITULO 1: COMPUERTAS LOGICAS

Existen varias familias de Circuitos integrados, pero para este proyecto solo se han estudiado las más comunes, TTL y las CMOS: estos Integrados se caracterizan por el número que corresponde a cada familia según su composición. Por ejemplo:

- Los TTL se corresponden con la serie 5400, 7400, 74LSXX, 74HCXX, 74HCTXX etc.
- Los C-MOS y MOS se corresponde con la serie CD4000, CD4500, MC14000, 54C00 o 74C00.

Los circuitos C-MOS son más lentos que los TTL, pero ocupan menos espacio; por eso su uso en algunos u otros equipos. Es importante buscar la hoja de datos o datasheet del integrado en cuestión, distribuido de forma gratuita por cada fabricante y disponible en Internet. (M, 2007)

Las compuertas más utilizadas en la familia TTL son:

- NOT – 74LS04
- AND – 74LS08
- NAND – 74LS00
- OR – 74LS32
- NOR – 74LS02
- XOR – 74LS86
- XNOR – 74LS266

Las compuertas más utilizadas en la familia C-MOS son:

- NOT – CD4069
- AND – CD4081
- NAND – CD4011/4093
- OR – CD4071
- NOR – CD4001

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- XOR – CD4070/4030
- XNOR – CD4077

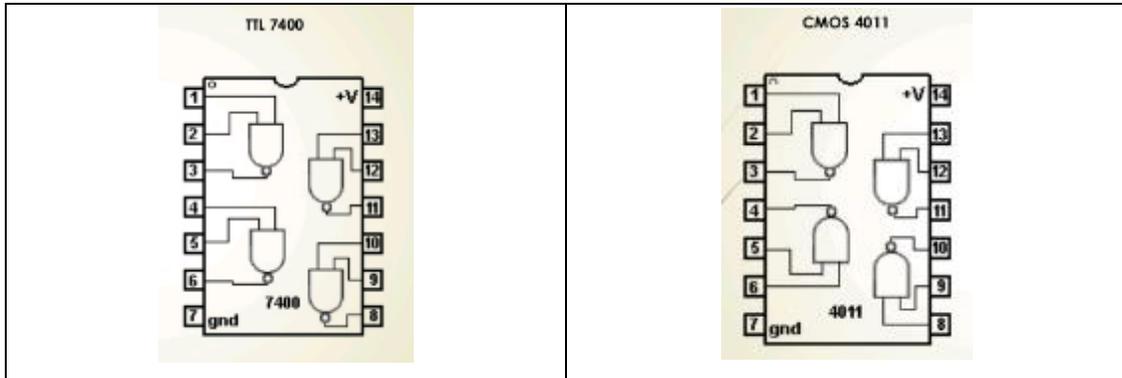


Tabla 3 Diferencias entre Compuertas TTL y C-MOS (Medina Salazar, 2015)

COMPUERTAS LOGICAS EN FORMA BINARIA

Las computadoras digitales utilizan el sistema de números binarios, que tiene dos dígitos 0 y 1. Un dígito binario se denomina un bit. La información está representada en las computadoras digitales en grupos de bits. Utilizando diversas técnicas de codificación los grupos de bits pueden hacerse que representen no solamente números binarios sino también otros símbolos discretos cualesquiera, tales como dígitos decimales o letras de alfabeto. Utilizando arreglos binarios y diversas técnicas de codificación, los dígitos binarios o grupos de bits pueden utilizarse para desarrollar conjuntos completos de instrucciones para realizar diversos tipos de cálculos.

La información binaria se representa en un sistema digital por cantidades físicas denominadas señales, Las señales eléctricas tales como voltajes existen a través del sistema digital en cualquiera de dos valores reconocibles y representan una variable binaria igual a 1 o 0. Por ejemplo, un sistema digital particular puede emplear una señal de 3 volts para representar el binario "1" y 0.5 volts para el binario "0".

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

La lógica binaria tiene que ver con variables binarias y con operaciones que toman un sentido lógico. La manipulación de información binaria se hace por circuitos lógicos que se denominan Compuertas.

Las compuertas son bloques del hardware que producen señales en binario 1 o 0 cuando se satisfacen los requisitos de entrada lógica. Las diversas compuertas lógicas se encuentran comúnmente en sistemas de computadoras digitales. Cada compuerta tiene un símbolo gráfico diferente y su operación puede describirse por medio de una función algebraica. Las relaciones entrada - salida de las variables binarias para cada compuerta pueden representarse en forma tabular en una tabla de verdad. (Molina Marticorena J. L., 2008)

Compuerta AND

Cada compuerta tiene dos variables de entrada designadas por A y B y una salida binaria designada por X.

La compuerta AND produce la multiplicación lógica AND: esto es: la salida es 1 si la entrada A y la entrada B están ambas en el binario 1: de otra manera, la salida es 0.

Estas condiciones también son especificadas en la tabla de verdad para la compuerta AND. La tabla muestra que la salida x es 1 solamente cuando ambas entradas A y B están en 1.

El símbolo de operación algebraico de la función AND es el mismo que el símbolo de la multiplicación de la aritmética ordinaria (*).

Las compuertas AND pueden tener más de dos entradas y por definición, la salida es 1 si todas las entradas son 1. (Molina Marticorena J. L., 2008)

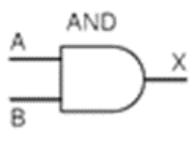
		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

Tabla 4 Compuerta AND (Molina Marticorena J. , Compuertas Logicas, 2008)

Compuerta OR

La compuerta OR produce la función sumadora, esto es, la salida es 1 si la entrada A o la entrada B o ambas entradas son 1; de otra manera, la salida es 0.

El símbolo algebraico de la función OR (+), es igual a la operación de aritmética de suma.

Las compuertas OR pueden tener más de dos entradas y por definición la salida es 1 si cualquier entrada es 1. (Molina Marticorena J. L., 2008)

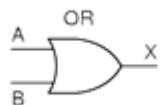
		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X															
0	0	0															
0	1	1															
1	0	1															
1	1	1															

Tabla 5 Compuerta OR (Molina Marticorena J. , Compuertas Logicas, 2008)

Compuerta NOT

El circuito NOT es un inversor que invierte el nivel lógico de una señal binaria. Produce el NOT, o función complementaria. El símbolo algebraico utilizado para el complemento es una barra sobre el símbolo de la variable binaria.

Si la variable binaria posee un valor 0, la compuerta NOT cambia su estado al valor 1 y viceversa.

El círculo pequeño en la salida de un símbolo gráfico de un inversor designa un inversor lógico. Es decir cambia los valores binarios 1 a 0 y viceversa. (Molina Marticorena J. L., 2008)

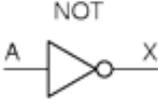
		<table border="1"> <thead> <tr> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	X	0	1	1	0
A	X							
0	1							
1	0							

Tabla 6. Compuerta NOT (Molina Marticorena J. L., Compuertas Logicas, 2008)

Compuerta NAND

Es el complemento de la función AND, como se indica por el símbolo gráfico, que consiste en una compuerta AND seguida por un pequeño círculo (quiere decir que invierte la señal).

La designación NAND se deriva de la abreviación NOT - AND. Una designación más adecuada habría sido AND invertido puesto que es la función AND la que se ha invertido.

Las compuertas NAND pueden tener más de dos entradas, y la salida es siempre el complemento de la función AND y NOT. (Molina Marticorena J. L., 2008)

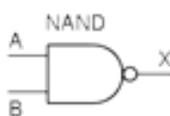
		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X															
0	0	1															
0	1	1															
1	0	1															
1	1	0															

Tabla 7. Compuerta NAND (Molina Marticorena J. L., Compuertas Logicas, 2008)

Compuerta NOR

La compuerta NOR es el complemento de la compuerta OR y utiliza el símbolo de la compuerta OR seguido de un círculo pequeño (quiere decir que invierte la señal). La

designación NOR se deriva de la abreviación NOT - OR. Una designación más adecuada habría sido OR invertido puesto que es la función OR la que se ha invertido.

Las compuertas NOR pueden tener más de dos entradas, y la salida es siempre el complemento de la función OR y NOT. (Molina Marticorena J. L., 2008)

		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0
A	B	X															
0	0	1															
0	1	0															
1	0	0															
1	1	0															

Tabla 8. Compuerta NOR (Molina Marticorena J. L., Compuertas Logicas, 2008)

Compuerta OR-EX o XOR o “OR exclusiva”

La OR Exclusiva tiene dos entradas, y lo que hará con ellas será una suma lógica entre “A” por “B” invertida y “A” invertida por “B”. Su tabla de verdad indica que la salida será 1 solo si una de las entradas lo es, pero la segunda no lo es, si lo son las dos al mismo tiempo el resultado será un 0. (Molina Marticorena J. L., 2008)

		<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>XOR</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	a	b	XOR	0	0	0	0	1	1	1	0	1	1	1	0
a	b	XOR															
0	0	0															
0	1	1															
1	0	1															
1	1	0															

Tabla 9. Compuerta XOR (Manuales electrónicos 4all, 2013)

Compuerta XNOR

Una compuerta NOR-exclusiva o XNOR opera en forma exactamente opuesta a una compuerta XOR, entregando un 0 cuando una de sus entradas sea 0 y la otra sea 1, y un 1 cuando las dos entradas tienen el mismo estado. (Molina Marticorena J. L., 2008)

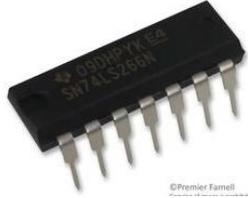
		<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>XNOR</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	b	XNOR	0	0	1	0	1	0	1	0	0	1	1	1
a	b	XNOR															
0	0	1															
0	1	0															
1	0	0															
1	1	1															

Tabla 10. Compuerta XNOR (Fernandez Perez, 2015)

2.2. CAPITULO 2: TRANSISTORES

Un transistor es un dispositivo que controla el flujo de una señal por medio de una segunda señal de mucha menor intensidad. La señal de control puede ser una señal de corriente o voltaje. El transistor es un dispositivo semiconductor, que presenta dos modos de funcionamiento: lineal y no lineal. Una de sus aplicaciones es la conmutación la cual se centra en la parte no lineal, que permite utilizar dos estados claramente diferenciado (corte y saturación; “1” lógico y “0” lógico), estos dispositivos también son fundamentales en la parte de amplificación de señales. (Boylestad & Nashelsky, Transistores de efecto campo, Polarización del FET, 1996, 6a Ed.)

En función de cuál sea el uso que quieras en tu circuito unos tipos de transistor son más eficientes que otros. Los tipos de transistor que se utilizan principalmente en la actualidad son:

- Transistor de unión bipolar o BJT.
- Transistor de efecto campo o FET.
- Fototransistores.
- JFET y Mosfet
- SCR

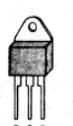
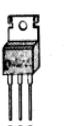
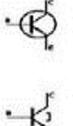
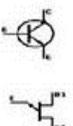
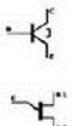
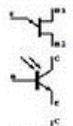
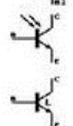
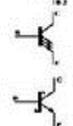
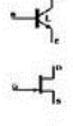
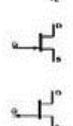
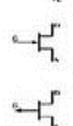
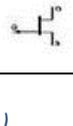
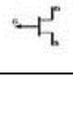
 2N3055 NPN	 2N5590 NPN	 PC100 NPN PC110 NPN	 2N6099 BDX53 NPN BDX54 NPN	 BD131 NPN BD132 NPN BD135 NPN BD136 PNP	 BD137 NPN BD138 PNP BD139 NPN BD140 PNP	 MC140 NPN MC150 PNP							
 BC547 NPN BC548 NPN BC549 NPN	 BSX20 NPN	 2N3866 NPN 2N3924 NPN	 SC107 NPN SC108 NPN SC109 NPN SF115 NPN	 SC157 PNP SC158 PNP SC159 PNP	 SC147 NPN SC148 NPN SC149 NPN	 AC125 PNP AC126 PNP AC127 NPN AC128 PNP							
 Transistor NPN	 Transistor PNP	 Transistor NPN con colector unido a la cubierta	 Transistor NPN túnel	 UJT-n Uniunión	 UJT-p Uniunión	 Fototransistor NPN	 Multiemisor NPN	 De avalancha NPN	 Transistor Schottky NPN	 Transistor JFET canal N	 Transistor JFET canal N	 Transistor JFET canal P	 Transistor JFET canal P

Tabla 11. Tipos de Transistores (Jose , s.f.)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

EL TRANSISTOR DE UNIÓN BIPOLAR O BJT.

Existen diversos tipos de transistores, entre ellos los TBJ o BJT Bipolar Junction Transistor, transistor bipolar de unión o juntura, en el que se permite que circule una corriente grande (entre el colector y el emisor) en función de una pequeña corriente que circula por la base.

En función de la corriente que esté circulando por la base el transistor se comportará de alguna de las siguientes formas:

Corte: circuito abierto (interruptor abierto) entre colector y emisor, si la corriente que circula por la base es nula.

Saturación: como circuito cerrando entre colector y emisor y con un aumento grande de corriente.

Activa: En un determinado rango de corrientes de base, la amplificación, el aumento de corriente, que se aprecia entre colector y emisor se puede regular.

Este tipo de dispositivo se gestiona mediante corriente y puede otorgar una gran amplificación, por lo que suelen ser preferibles estos tipos de transistor a la hora de utilizarlos como amplificadores (aunque también se pueden utilizar como interruptores, sobre todo cuando los cambios de estado corte-saturación no tienen que ser excesivamente rápidos, es decir, no se está trabajando a una frecuencia muy elevada).

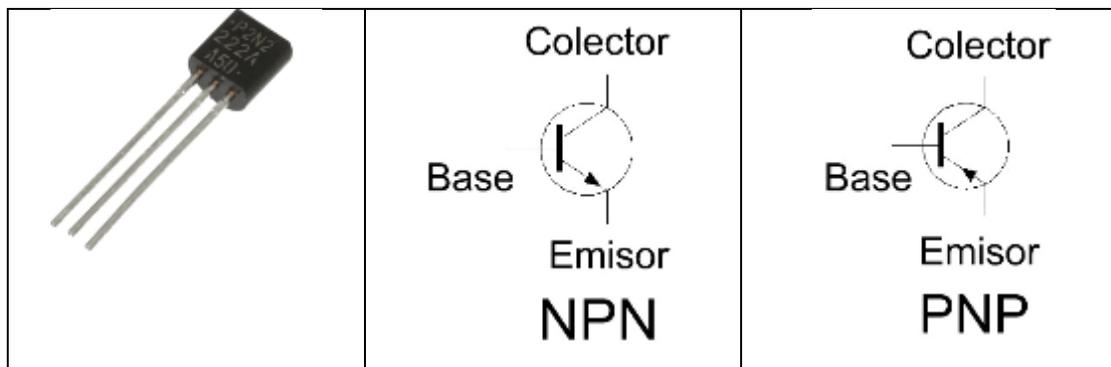


Tabla 12. Transistor de Unión Bipolar o BJT. (www.neoteo.com, 2009)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

(www.neoteo.com, 2009)

TRANSISTOR DE EFECTO CAMPO O FET.

El transistor de efecto de campo FET - Field Effect Transistor, es un dispositivo de tres terminales para aplicaciones diversas que se asemejan, en una gran proporción, a los transistores BJT (del inglés Bipolar Junction Transistor, transistor bipolar de unión). Aunque existen importantes diferencias entre los dos tipos de dispositivos, también es cierto que tienen muchas similitudes. (Boylestad & Nashelsky, Transistores de efecto campo, Polarización del FET, 1996, 6a Ed.)

Una de las características más importantes del FET es su gran impedancia de entrada. Con un nivel de uno, hasta varios cientos de megaohms, excede por mucho los niveles típicos de resistencia de entrada de las configuraciones con transistores BJT, lo cual es una característica muy importante para el diseño de sistemas de amplificación. Por otro lado, el transistor BJT tiene mayor sensibilidad ante cambios en la señal aplicada. En otras palabras, la variación en la corriente de salida, por lo general, es mucho mayor para el caso de los BJTs que para el caso de los FETs, ante el mismo cambio de voltaje aplicado. Por esta razón, las ganancias comunes de voltaje para los amplificadores de BJT son mucho mayores que para los de FET. En general, los FETs son más estables a la temperatura que los BJTs, lo cual los hace particularmente útiles en los circuitos integrados. Sin embargo, las características de construcción de algunos FETs los pueden hacer más delicado que los BJTs. (VIDEOSISTEMAS LTD. Videosistemas Corporation Int., 2011), (Scribd Inc., s.f.)

Los transistores de efecto de campo FET, normalmente tienen tres terminales denominados: puerta (Gate) similar a la base en los transistores bipolares que, controla el flujo de corriente entre los otros dos, la fuente (Surtidor) y el drenador (Drain). Una diferencia significativa frente a los transistores bipolares es que, la puerta no requiere del consumo de una intensidad como ocurre con los transistores bipolares que si bien es muy pequeña (depende de la ganancia), no se ha de despreciar.

Pese a que el concepto básico de los FET se conocía ya en 1930, estos dispositivos sólo empezaron a fabricarse comercialmente a partir de la década de los 60. Y a partir de los 80 los transistores de tipo MOSFET han alcanzado una enorme popularidad.

El transistor BJT es un dispositivo controlado por corriente, mientras que el transistor efecto de campo es un dispositivo controlado por voltaje. (Schiavon, 1997)

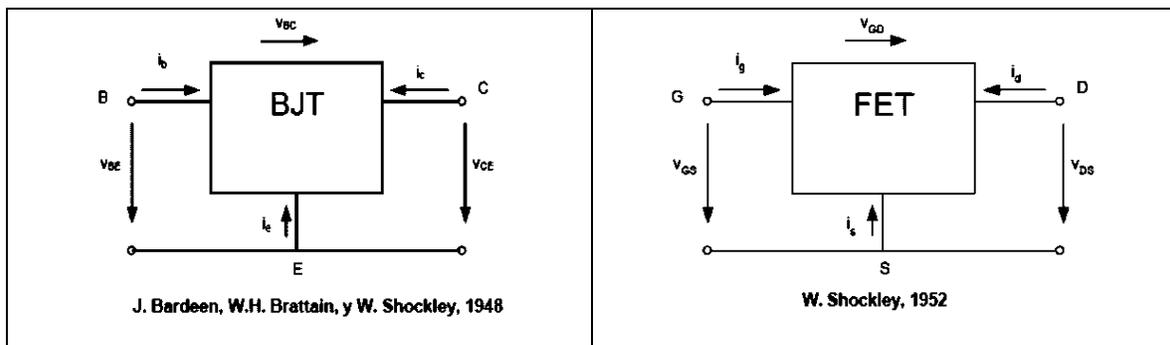


Tabla 13. Transistor BJT vs Efecto Campo o FET.

TRANSISTOR TIPO JFET Junction Field Effect Transistor

El JFET es un dispositivo de tres terminales, con una terminal capaz de controlar la corriente entre las otras dos, utiliza un campo eléctrico para controlar la conductividad de un canal en un semiconductor. (Villalba Madrid & Zamora Izquierdo, 2008)

La estructura física de un JFET (transistor de efecto campo de unión o juntura) consiste en un canal de semiconductor tipo n o p dependiendo del tipo de JFET, con contactos en cada extremo, llamados FUENTE (source) y DRENADOR O SUMIDERO (drain). A los lados del canal existen dos regiones de material semiconductor de diferente tipo al canal, conectados entre sí, formando el terminal de PUERTA o COMPUERTA (gate). (Schiavon, 1997). El símbolo grafico que se utiliza para representar al JFET de tipo n en los circuitos se muestra con la flecha de la compuerta apuntando hacia dentro (a la derecha) para indicar el carácter n del semiconductor. En el caso del JFET de tipo p, la flecha de la puerta tiene la dirección opuesta a la figura. (College of Engineering, University of Illinois, 2015)

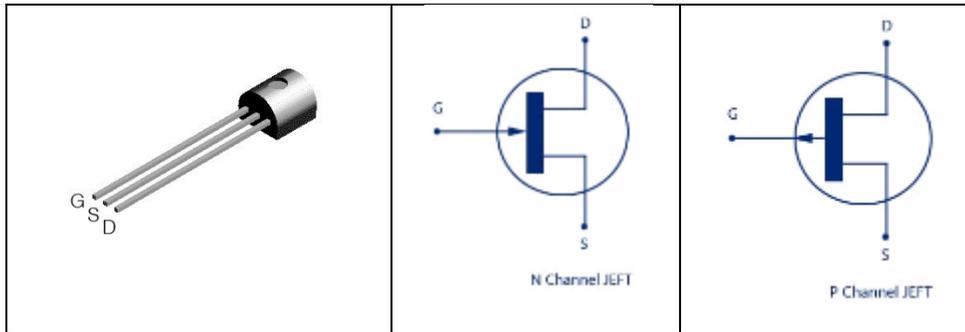


Tabla 14. Transistor JFET Junction Field Effect Transistor (CircuitsToday, 2013)

TRANSISTOR TIPO MOSFET Metal-Oxide-Semiconductor FET

Los transistores MOSFET o Metal-Oxido-Semiconductor (MOS) son dispositivos de efecto de campo que utilizan un campo eléctrico para crear una canal de conducción. Son dispositivos más importantes que los JFET ya que la mayor parte de los circuitos integrados digitales se construyen con la tecnología MOS

Existen dos tipos de transistores MOS: MOSFET de canal N o NMOS y MOSFET de canal P o PMOS. A su vez, estos transistores pueden ser de acumulación (enhancement) o depleción (depletion); en la actualidad los segundos están prácticamente en desuso, el símbolo del transistor de canal en el cual la flecha indica el sentido convencional de la corriente (drenador a fuente). En el MOSFET de canal p, la corriente tiene sentido opuesto y la flecha del dibujo lleva la dirección invertida. (Schiavon, 1997)

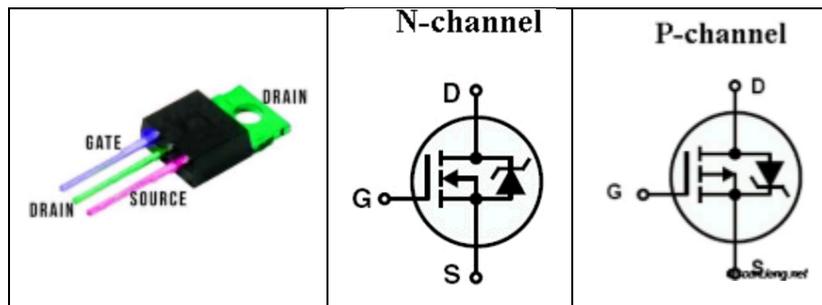


Tabla 15. Transistor Mosfet (Oscar Lian, 2013)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FOTOTRANSISTORES

Este componente funciona cuando la luz (visible o infrarroja) incide en él, cuando la luz alcanza la base de este elemento sucede que se generan portadores en ella, la carga de base generada por los portadores lleva al transistor al estado de conducción.

Un fototransistor común puede poseer 2 modos, el primer modo es el común y el segundo es el modo de iluminación

Modo común: En este modo el fototransistor puede funcionar tal y como si fuera un transistor normal

Modo de iluminación: Este es el modo en que se utiliza realmente uno de estos elementos, tal y como su nombre lo indica es cuando el fototransistor recibe luz y esto conlleva que se creen portadores que posteriormente vuelven conductivo al transistor.

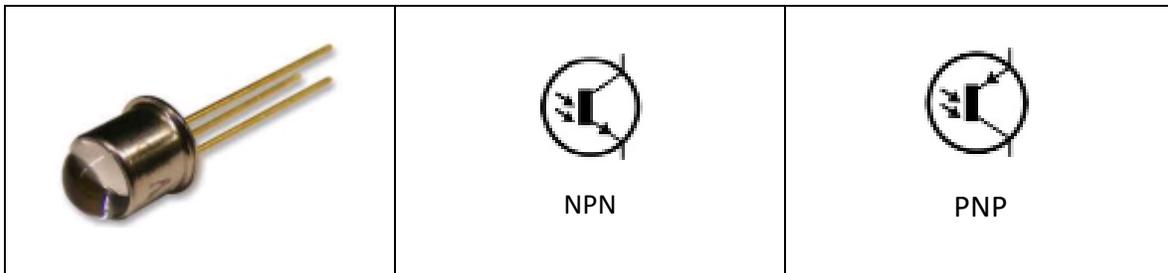


Tabla 16. Fototransistor (Electronica PT, s.f.)

TRANSISTOR TIPO SCR Silicon Controlled Rectifier

El rectificador controlado de silicio (en inglés SCR: Silicon Controlled Rectifier) es un tipo de tiristor formado por cuatro capas de material semiconductor con estructura PNPN o bien NPNP. El nombre proviene de la unión de Tiratrón (tyratron) y Transistor.

Un SCR posee tres conexiones: ánodo, cátodo y gate (puerta). La puerta es la encargada de controlar el paso de corriente entre el ánodo y el cátodo. Funciona básicamente como un

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

diodo rectificador controlado, permitiendo circular la corriente en un solo sentido. Mientras no se aplique ninguna tensión en la puerta del SCR no se inicia la conducción y en el instante en que se aplique dicha tensión, el tiristor comienza a conducir. Trabajando en corriente alterna el SCR se des-excita en cada alternancia o semiciclo. Trabajando en corriente continua, se necesita un circuito de bloqueo forzado, o bien interrumpir el circuito.

El pulso de conmutación ha de ser de una duración considerable, o bien, repetitivo si se está trabajando en corriente alterna. En este último caso, según se atrase o adelante el pulso de disparo, se controla el punto (o la fase) en el que la corriente pasa a la carga. Una vez arrancado, podemos anular la tensión de puerta y el tiristor continuará conduciendo hasta que la corriente de carga disminuya por debajo de la corriente de mantenimiento (en la práctica, cuando la onda senoidal cruza por cero)

Los SCR se utilizan en aplicaciones de electrónica de potencia, en el campo del control, especialmente control de motores, debido a que puede ser usado como interruptor de tipo electrónico.

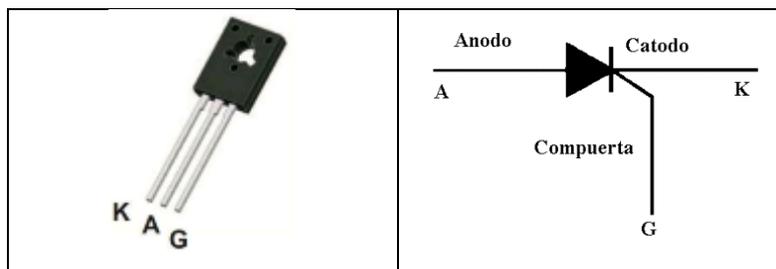


Tabla 17. SCR - Silicon Controlled Rectifier (Angulo & Barreto, 2013)

TRANSISTOR TIPO TRIAC - Triodo para Corriente Alterna

El triac es un dispositivo semiconductor de tres terminales que se usa para controlar el flujo de corriente promedio a una carga, con la particularidad de que conduce en ambos sentidos y puede ser bloqueado por inversión de la tensión o al disminuir la corriente por debajo del valor de mantenimiento. El triac puede ser disparado independientemente de la polarización de puerta, es decir, mediante una corriente de puerta positiva o negativa.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Un triac es un tipo de tiristor que puede conducir en ambas direcciones (corriente AC), ya que se puede considerar como si fueran dos SCR (Rectificador controlado de silicio) conectados en anti-paralelo, con una conexión de compuerta común. Un SCR es un dispositivo semiconductor que transfiere corriente sólo en los ciclos positivos de AC. Puesto que el triac es un dispositivo bidireccional, no es posible identificar sus terminales (ánodo y cátodo).

El triac tiene tres terminales denominadas MT1, MT2 y G; MT1 y MT2 son las terminales principales y G es la compuerta de control. Si la terminal MT2 es positiva con respecto a la terminal MT1, el triac se puede activar aplicando una señal de compuerta positiva entre MT1 y G. Caso contrario cuando la terminal MT2 es negativa con respecto a MT1, se activará con una señal de compuerta negativa. No es necesario que estén presentes ambas polaridades, ya que puede ser activado con una sola señal de compuerta (positiva o negativa). (Boylestad N. , 2003)

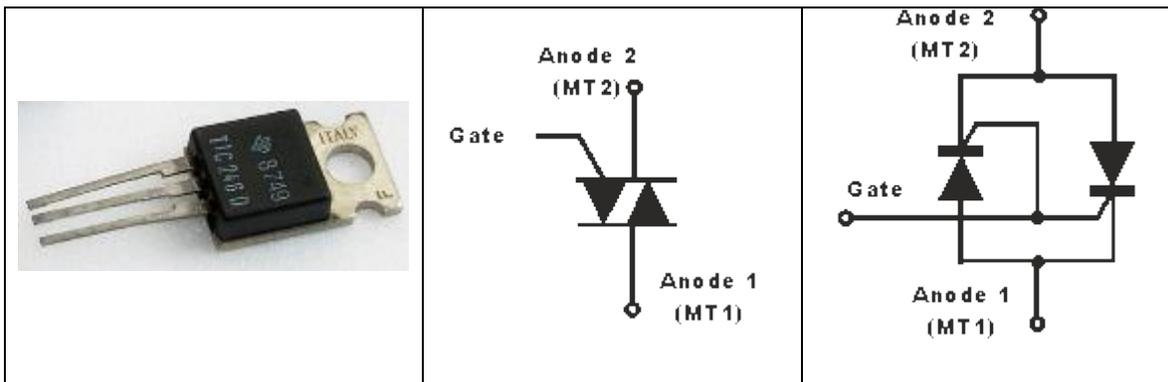


Tabla 18. Transistor TRIAC (Poole, n.d.)

2.3. CAPITULO 3: DISPLAY 7 SEGMENTOS DE ANODO COMUN Y CATODO COMUN

Los displays de 7 segmentos son por demás conocidos. Son muy usados en los equipos electrónicos desde hace años, y no ofrecen ningún tipo de dificultad. Se los utiliza, en general, en forma directa o multiplexada. En forma directa, es usado normalmente en chips que tienen incluidos los drivers para ello; como ser el ICL7107. En forma multiplexada, son utilizados por ejemplo con un microcontrolador, el cual genera las señales para manejar los puntos comunes (ánodo o cátodo) por un lado y los segmentos, por el otro. Para lograr esto, se utiliza algún transistor PNP para manejar el ánodo común, o NPN para el cátodo común. Para los segmentos; un clásico ULN2003 (o ULN2004, ULN2803, etc.), o en forma directa para los microcontroladores que tienen la capacidad de manejar 20mA por línea de entrada-salida. Por supuesto, hay que limitar la corriente con el uso de una resistencia en cada segmento.

Un Display de este tipo está compuesto por siete u ocho leds de diferentes formas especiales y dispuestas sobre una base de manera que puedan representarse todos los símbolos numéricos y algunas letras. Los primeros siete segmentos son los encargados de formar el símbolo y con el octavo podemos encender y apagar el punto decimal

Este display es de alto consumo energético ya que cada segmento o diodo consume 20 mA con 2V y 30 mA con 5V.

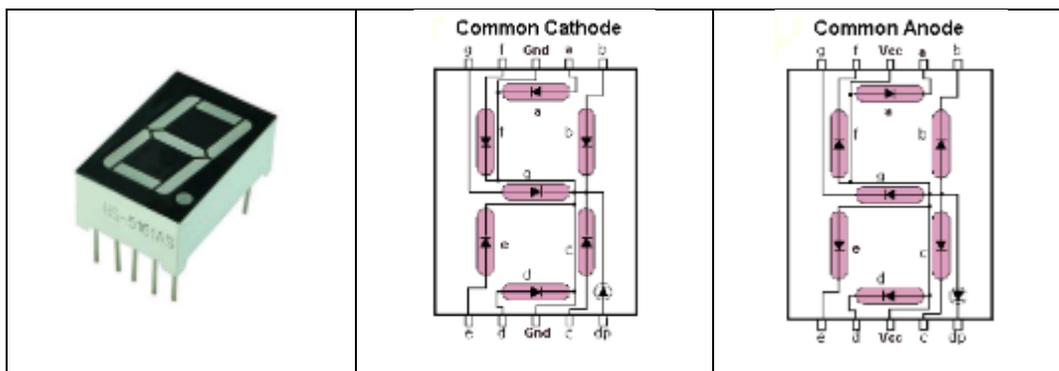


Tabla 19. Display 7 Segmentos AC y CC (Electronic Tools, s.f.)

Los segmentos son denominados convencionalmente de “a” hasta “g”, siendo posible hacer que muestren cualquier numero de 0 a 9 o un carácter alfabético (no todos), mezclando letras mayúsculas y minúsculas, activando estos segmentos en distintas combinaciones.

Catodo Comun								Anodo Comun								Función del Pin											
Enable	Numero	A	B	C	D	E	F	G	Enable	Numero	A	B	C	D	E	F	G	1	2	G	3	4	5	6	7	8	
0	0	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	1	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	1	0	1	1	0	0	0	0	1	1	1	0	0	1	0	1	0	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	2	1	1	0	1	1	0	1	1	0	0	1	0	0	1	0	0	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	3	1	1	1	1	0	0	1	1	0	0	0	1	1	0	0	0	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	4	0	1	1	0	0	1	1	1	0	0	1	1	0	0	0	0	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	5	1	0	1	1	0	1	1	1	0	1	0	1	0	0	0	0	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	6	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	7	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto
0	9	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	Led G	Led F	Pin Común	Led A	Led B	Led E	Led D	Pin Común	Led C	Led del Punto

Tabla 20. Pines, Números y Letras del Display (Electronic Tools, s.f.)

Características

Solidez: excelente

Angulo de visibilidad: 150 grados

Consumo por dígito: 150 mW

Vida media en horas: 100000

Luminosidad: buena

Facilidad de montaje: excelente Vcc (general): 1'5 V La Vcc depende del color del LED. Para un color rojo: Vcc: 1'7 V Vcc (máx): 2 V

Dependiendo de la tensión aplicada obtendremos una intensidad. Es aconsejable no sobrepasar la Vcc recomendada. Si se alcanza la Vcc máxima se puede destruir el segmento.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Protección Cada segmento (y el punto) es un led como cualquier otro. Debido a esto la corriente media que se debe aplicar es de 15 mA. Dependiendo de la lógica que estemos empleando debemos utilizar una resistencia por cada entrada y así no forzar el dispositivo:

Lógica TTL (5V): 220 ohmios

Lógica CMOS (12V): 680 ohmios. Esta resistencia debe ser situada en cada patilla, haciendo de puente entre la señal lógica de excitación y el Display. (Universidad Nacional Autonoma de Mexico, Facultad de Ingenieria, 2010)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.4. CAPITULO 4: TEMPORIZADOR 555

El temporizador IC 555 es un circuito integrado (chip) que se utiliza en la generación de temporizadores, pulsos y oscilaciones. El 555 puede ser utilizado para proporcionar retardos de tiempo, como un oscilador, y como un circuito integrado flip flop. Sus derivados proporcionan hasta cuatro circuitos de sincronización en un solo paquete.

Fue introducido en 1971 por Signetics, el 555 sigue siendo de uso generalizado debido a su facilidad de uso, precio bajo y la estabilidad. Muchas empresas los fabrican en versión de transistores bipolares y también en CMOS de baja potencia. A partir de 2003, se estimaba que mil millones de unidades se fabricaban cada año. Este circuito suele ser utilizado para trabajos sencillos como trabajos escolares, debido a su bajo costo y facilidad de trabajar con él.

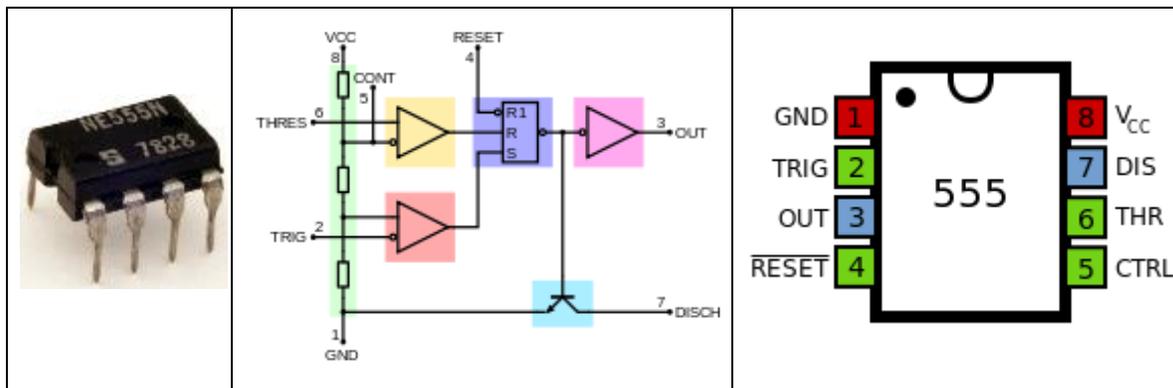


Tabla 21. Temporizador 555 (es.wikipedia.org, 2005)

Descripción de las conexiones

GND (normalmente la 1): es el polo negativo de la alimentación, generalmente tierra (masa).

Disparo (normalmente la 2): Es donde se establece el inicio del tiempo de retardo si el 555 es configurado como monoestable. Este proceso de disparo ocurre cuando esta patilla tiene menos de 1/3 del voltaje de alimentación. Este pulso debe ser de corta duración, pues si se mantiene bajo por mucho tiempo la salida se quedará en alto hasta que la entrada de disparo pase a alto otra vez.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Salida (normalmente la 3): Aquí veremos el resultado de la operación del temporizador, ya sea que esté conectado como monoestable, estable u otro. Cuando la salida es alta, el voltaje será el voltaje de alimentación (Vcc) menos 1.7 V. Esta salida se puede obligar a estar en casi 0 voltios con la ayuda de la patilla de reinicio (normalmente la 4).

Reinicio (normalmente la 4): Si se pone a un nivel por debajo de 0.7 Voltios, pone la patilla de salida a nivel bajo. Si por algún motivo esta patilla no se utiliza hay que conectarla a alimentación para evitar que el temporizador se reinicie.

Control de voltaje (normalmente la 5): Cuando el temporizador se utiliza en el modo de controlador de voltaje, el voltaje en esta patilla puede variar casi desde Vcc (en la práctica como Vcc -1.7 V) hasta casi 0 V (aprox. 2 V menos). Así es posible modificar los tiempos. Puede también configurarse para, por ejemplo, generar pulsos en rampa.

Umbral (normalmente la 6): Es una entrada a un comparador interno que se utiliza para poner la salida a nivel bajo.

Descarga (normalmente la 7): Utilizado para descargar con efectividad el condensador externo utilizado por el temporizador para su funcionamiento.

Voltaje de alimentación (VCC) (normalmente la 8): es la patilla donde se conecta el voltaje de alimentación que va de 4.5 V hasta 16 V. (Tocci, Widmer, & Moss, 2007) (Fairchild Semiconductor Components Industries, LLC., 2013)

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGÍA

Para el desarrollo de este proyecto nos apoyaremos en la metodología de la investigación explicativa; las teorías alrededor de esta constituyen el conjunto organizado de principios, inferencias, creencias, descubrimientos y afirmaciones por medio del cual se interpreta una realidad.

Una teoría o explicación, contiene un conjunto de definiciones y de suposiciones relacionados entre si de manera organizada sistemática; estos supuestos deben ser coherentes con el tema de estudio.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para este proyecto utilizaremos un microcontrolador PIC16F887, una pantalla LCD 16x2 (16 columnas, 2 filas), transistores de 33pF, 1uF, 0.1uF, pulsadores normalmente abiertos, diodos led, sokets de: 4, 8, 12, 16, 24, 28 y 40 pines para los respectivos componentes a medir, Oscilador de cristal de 4MHz y 10Mhz, resistencias de 220, 330, 500, 1kOhm, un PCF8574, un CD4511 para display 7 segmentos de cátodo común y un CD4543 para display 7 segmentos de ánodo común, potenciómetro 10KOhm, transistores, cable

Con cuatro botones (enter, atrás, arriba, abajo) el usuario podrá navegar a través de un menú que le permitirá escoger que componente desea medir.

El primer componente a inspeccionar es el estado de las compuertas lógicas anteriormente mencionadas, para realizar dicha medición, el microcontrolador será programado de manera tal que envíe estados altos al sistema alimentando así las entradas de cada compuerta para que el sistema compruebe la tabla de verdad de dicha compuerta, y pueda generar una salida alta para iluminar un led, que indicara que la compuerta esta buena.

Por ejemplo, tomemos un circuito integrado 74LS08, que es un TTL, cuádruple compuerta AND. Hay que tener en cuenta el sentido en que están numerados los pines. El representado en el gráfico marca una de las compuertas que será puesta a prueba, para ello utilizaremos una fuente regulada de +5V, un banco de pulsadores, un banco de LEDS, el IC que corresponda y una placa de prueba.

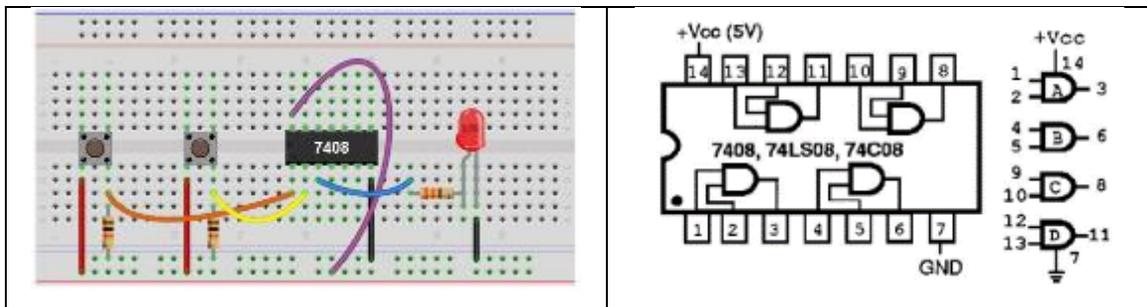


Tabla 22. Conexión Compuerta AND (Lorenzo Flores, 2015)

Las aplicaciones de los transistores no se limitan únicamente a la amplificación de señales. Mediante un diseño apropiado, se pueden utilizar como interruptores para aplicaciones de

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

cómputo y control. Según sea un transistor NPN o PNP, varía la polaridad de las conexiones del emisor, el colector y la base.

En un transistor de tipo NPN, para que la base pueda funcionar como interruptor, debe tener polaridad positiva y el emisor debe tener polaridad negativa para generar el flujo de electrones.

Cuando la base del transistor se le corte la corriente o se conecte al polo negativo, el transistor NPN dejara de conducir (se comporta como un circuito abierto) y no pasara la corriente del emisor a colector.

En un transistor PNP la base debe tener una polaridad negativa.

Este concepto básico aplica para casi todos los transistores pues cuentan con el mismo principio.

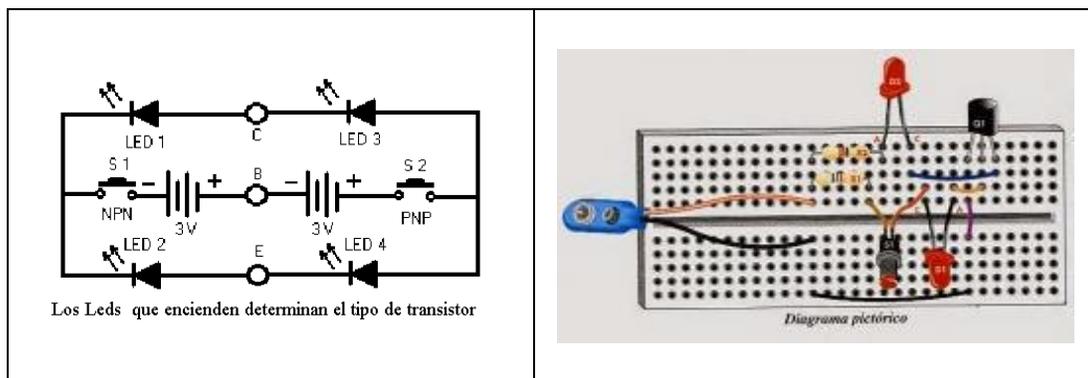


Tabla 23. Conexión Transistor NPN y PNP (Comunidad Internacional de electronicos, 2012)

Para realizar la medición del display de 7 segmentos se debe elegir si es de modo ánodo común o cátodo común (AC o CC), se debe introducir el display en el banco de prueba, para realizar la medición.

Para este módulo el sistema realizara un conteo de 0 a 9, en el número ocho (8) se encenderán todos los segmentos del display, indicando así que dicho componente está en buen estado.

El numero binario para indicar el 8 en el display es: $b'01111111' = 7Fh=8$

LETRA PARA MOSTRAR EN 7-SEG							
DP	G	F	E	D	C	B	A
0	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1
0	0	1	1	1	0	0	1

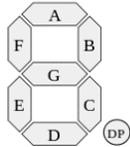
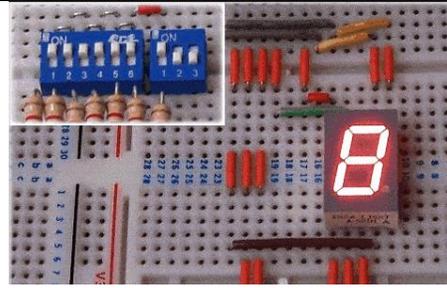



Tabla 24. Visualización Display 7 Segmentos (Area tecnologia, s.f.)

Integramos además un circuito comunicado a un temporizador 555, que se utiliza para activar o desactivar leds durante intervalos de tiempo determinados, es decir se usa como temporizador, donde se podrán medir diferentes intervalos de tiempo de encendido.

Otros usos 555 es para luces intermitentes, regular el tiempo que tarda en apagarse una luz, ajustar el tiempo en una tostadora, etc. (Area Tecnologia, 2016)

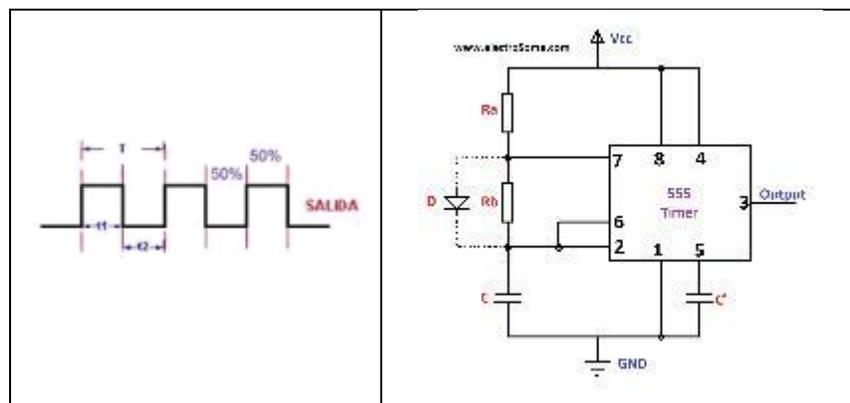
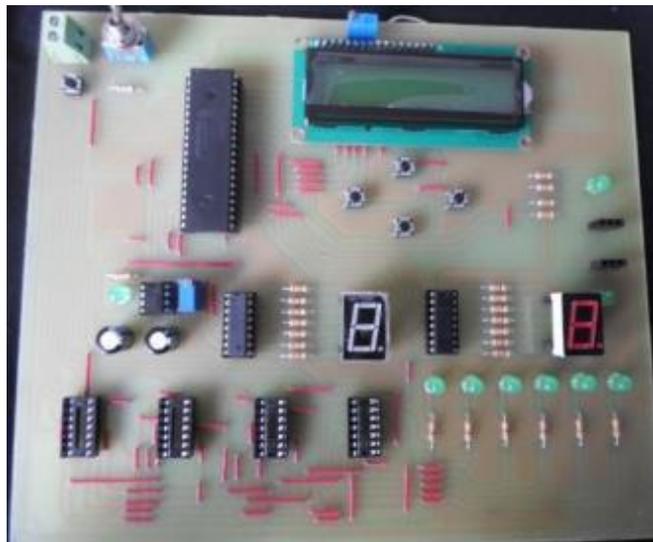


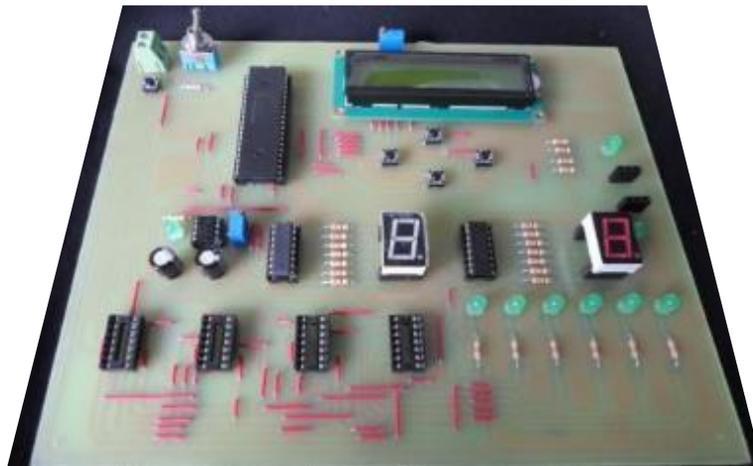
Tabla 25. Timer 555 (ElectroSome, 2014)

4. RESULTADOS

Se logró crear una tarjeta o módulo de manera que pueda medir diferentes componentes electrónicos, totalmente dotada de un acompañamiento visual por medio de zóquets donde insertar los componentes, iluminación led y conexiones especiales que le permite al estudiante interactuar de manera segura con el módulo.



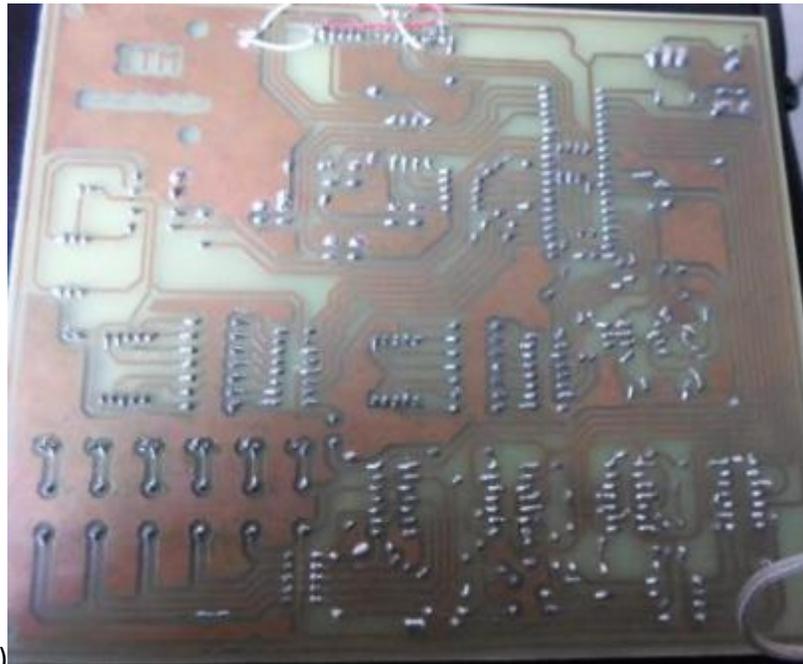
(Rojas Alzate , 2017)



(Rojas Alzate , 2017)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

El proceso de integración de la tarjeta, se dio gracias al diseño en ARES de ISIS PROTEUS, donde se pudo diseñar cada pista interconectada a su componente correspondiente, se realizó la impresión en cobre sobre una baquelita, o plancha plástica, donde los espacios planos en cobre le brindan estabilidad a la señal enviada, y representan una tierra, las pistas o canales que unen los componentes, fueron revisadas y aisladas para evitar cortocircuitos en el módulo.



(Oscar Lian, 2013)

(Rojas Alzate , 2017)

Debido a que el modulo es una donación a la universidad se marcó para que quede constancia del mismo



(Rojas Alzate , 2017)

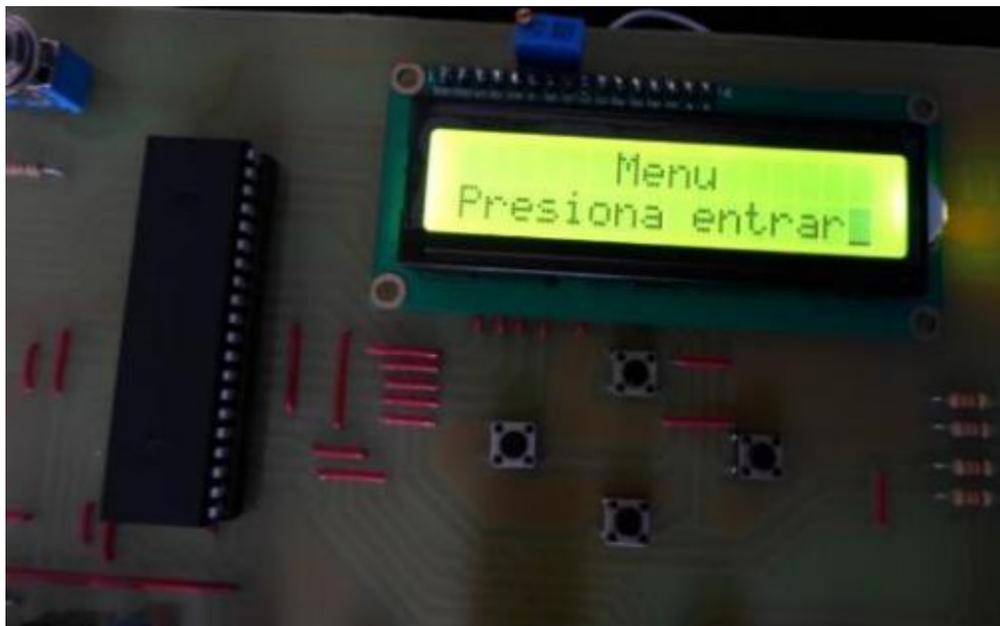
	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Cuando se enciende el módulo, muestra el siguiente mensaje de bienvenida.



(Rojas Alzate , 2017)

Después de esto se visualizará la siguiente pantalla, donde podremos empezar a interactuar con el módulo, con la ayuda de cuatro pulsadores que indican arriba abajo atrás y enter, y una pantalla led que permite la visualización y selección del componente a medir.



(Rojas Alzate , 2017)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5. CONCLUSIONES

Después de realizar una investigación acerca de la necesidad e importancia de la implementación y diseño de un módulo didáctico fundamentado en la programación de un microcontrolador se ha llegado a las siguientes conclusiones.

Se logró el desarrollo del sistema electrónico que cumple con los objetivos propuestos y con las especificaciones requeridas en cuanto a seguridad electrónica y eléctrica por medio del manual.

Se desarrolló un sistema de menús con pulsadores, de forma que el estudiante pueda operar e ingresar al componente que sea requerido en la pantalla LCD, por ende, los registros de desplazamiento es una herramienta útil para el manejo del display.

El ISIS PROTEUS es un software donde se realizó y simuló en tiempo real el circuito para el impreso del mismo, haciendo que el módulo pudiera cumplir con las expectativas de diferentes prácticas

Se realizó el diseño e implementación del sistema con información detallada de tal manera que el estudiante pueda obtener datos oportunos y reales del funcionamiento de los componentes.

Cada etapa o módulo de la tarjeta funciona de manera aislada, de modo tal que solo se pueda medir un componente por vez, para evitar cortos o un mal uso del mismo.

El sistema es mucho más sencillo de utilizar, ya que contó con diferentes aplicaciones reunidas en una sola tarjeta, de esta manera nos evitamos el uso de buses para la interconexión de los diferentes módulos del sistema.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

El desarrollo del manual permite que el estudiante comprenda la operación del módulo para que pueda realizar las operaciones y acceder a las diferentes aplicaciones, de manera segura y contada con un buen uso del mismo.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

6. RECOMENDACIONES Y TRABAJO FUTURO

La conceptualización de cómo puede ser verificado cada elemento y la materialización del mismo a través del módulo, permiten tanto al practicante como al usuario una interacción no solo física sino teórica del manejo y aplicación de los elementos técnicos e informativos.

Se recomienda el uso de Jumpers¹, para la selección de una o más etapas ya, que esta selección se le puede hacer también por medio de programación pero esto demanda el uso de más pines del PIC.

Este sistema fue construido pensando en aplicaciones futuras, ya que puede constar de comunicación serial a través del puerto RS232 y Puerto USB, este último se encuentra en gran desarrollo en nuestro tiempo y es fácil de modificar agregando líneas al código al programa del PIC.

Se recomienda a futuro la implementación de otros elementos de visualización como pantallas táctiles y entornos de usuario más digitales como los espacios virtuales de aprendizaje como la WWW “o “IBT (Internet based training) en los cuales no hay contacto físico, no con los componentes más si un contacto directo con la información y los principios físicos de los mismos sin incurrir en los riesgos del corto circuito, los costos de daños o en los costos de adquisición de elementos.

¹ Un jumper o saltador es un elemento que permite cerrar el circuito eléctrico del que forma parte dos conexiones. Esto puede hacerse mediante soldadura (se derrite suficiente estaño para cerrar el circuito), soldando un cable o alambre entre ambos puntos o, lo más frecuente, conectado dos pines en hilera o paralelo mediante una pieza de plástico que protege el material conductor que cierra el circuito.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Con el laboratorio, se desea que implementen módulos adicionales con aplicaciones y procesos industriales tales como, sensores de temperatura mediante un LM35, reguladores de voltaje, amplificadores de señal, y un sistema de visualización de ondas, que permitan extender la plataforma de trabajo, enriqueciendo así el conocimiento adquirido e incorporando nuevos recursos para la enseñanza, y así mismo facilitar la interacción módulo/estudiante.

El mantenimiento es mínimo, pero es vital para el módulo y para que preserve las condiciones de funcionamiento.

Se requiere que el estudiante tenga conocimientos básicos de electrónica digital, para así poder hacer buen uso del módulo, si esto no es posible, deberá solicitar asesoría y explicación del manual de uso.

Se debe tener cuidado con las conexiones del sistema de alimentación del módulo, ya que una mala conexión puede dañar no solo el módulo sino los componentes. Se recomienda que el voltaje de alimentación no supere los 5V (5 Voltios).

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

- Angulo, V., & Barreto, A. (29 de Agosto de 2013). *Laboratorios Electrónica V.A.* Obtenido de Laboratorio de SCR (Rectificador de silicio): <http://2.bp.blogspot.com/-TGZZ60wROFo/Uh-NIYBtuPI/AAAAAAAAACA/c4ibaU2x9Yo/s1600/scr1.gif>
- Aprendiendo Arduino . (23 de marzo de 2015). *Aprendiendo Arduino*. Obtenido de Microcontroladores: <https://aprendiendoarduino.wordpress.com/2015/03/23/microcontroladores/>
- Area tecnologia. (s.f.). Recuperado el 17 de Febrero de 2017, de Display 7 segmentos: <http://www.areatecnologia.com/electronica/led-display.html>
- Area Tecnologia. (2016). Obtenido de <http://www.areatecnologia.com/electronica/circuito-integrado-555.html>
- Boylestad, N. (2003). *Electrónica: Teoría de circuitos y dispositivos electrónicos*. EU Pearson.
- Boylestad, R. L. (2011, 12a ed.). *Introducción al análisis de circuitos*. Mexico: Pearson Educación de México.
- Boylestad, R. L., & Nashelsky, L. (1996, 6a Ed.). Transistores de efecto campo, Polarización del FET. En *Teoría de circuitos* (pág. Capítulo 5 y 6.). Mexico: Pentice Hall.
- CircuitsToday. (2013, Noviembre 12). *JFET-Junction Field Effect Transistor*. Retrieved from <http://www.circuitstoday.com/wp-content/uploads/2009/08/JFET-N-Channel-and-P-channel-Schematic-Symbol.jpg>
- College of Engineering, University of Illinois. (2015). <http://fabweb.ece.illinois.edu/>. Obtenido de <http://fabweb.ece.illinois.edu/ppt/ece444.ppt>
- Comunidad Internacional de electronicos. (23 de Febrero de 2012). *Probador de Transistores*. Obtenido de http://1.bp.blogspot.com/_hrX26Bn9UBk/SebDTyBtZEI/AAAAAAAAAXg/oX2AMZXO8lw/s400/probtransist.gif
- Electronic Tools. (s.f.). *DISPLAY 7 SEGMENTOS*. Recuperado el 17 de Febrero de 2017, de <http://www.electrontools.com/Home/WP/2016/03/09/display-7-segmentos/>
- Electronica PT. (s.f.). *transistor*. Recuperado el 17 de Febrero de 2017, de TRANSISTOR DARLINGTON - FOTOTRANSISTOR: <https://www.electronica-pt.com/imagens/fototransistor.jpg>

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- ElectroSome. (2014, Noviembre 28). Retrieved from Astable Multivibrator using 555 Timer:
<https://electrosome.com/wp-content/uploads/2012/11/555-timer-astable-multivibraotr.jpg>
- Elsivier, W. T. (1ra edicion 2007). *“Designing Embedded Systems with PIC Microcontrollers: Principles and Applications”*.
- es.wikipedia.org. (02 de Octubre de 2005). *Circuito integrado 555*. Obtenido de
https://es.wikipedia.org/wiki/Circuito_integrado_555
- Fairchild Semiconductor Components Industries, LLC. (2013, Enero). Retrieved from
<https://www.fairchildsemi.com/datasheets/LM/LM555.pdf>
- Fernandez Perez, G. (25 de Noviembre de 2015). Obtenido de COMPUERTAS LOGICAS:
http://4.bp.blogspot.com/-iFQn_R0loEw/Vla0HXChJxl/AAAAAAACDY/LyUQnMyCx9w/s1600/12.png
- J. A. (s.f.). *universidad de granada*. Recuperado el 17 de Febrero de 2017, de Transistores:
http://electronica.ugr.es/~amroldan/asignaturas/curso03-04/cce/practicas/encapsulados/encapsulados/tr_muestra2.gif
- Lajara Vizcaino, J. R., & Pelegrí Sebastián, J. (2007, 1ra ed.). *LabVIEW Entorno gráfico de programación*. Alfaomega.
- Lorenzo Flores, J. E. (26 de Noviembre de 2015). Obtenido de Compuertas Lógicas:
<http://eduardolofl.blogspot.com.co/2015/11/compuertas-logicas.html>
- M, A. (2007). *“Principios de la electronica”*. Madrid, España: McGrawHill.
- Manuales electrónicos 4all. (17 de Octubre de 2013). *Catalogs 4all*. Obtenido de
<http://avellano.fis.usal.es/~compi/xor.png>
- Matric, N., & Andric, D. (2000, 3a ed.). *The PIC microcontroller*. MikroElectronica.
- Medina Salazar, C. (15 de mayo de 2015). *slide share - Compuertas Lógicas NOR, XOR, NAND, XNOR*. Obtenido de diapositiva 16:
<https://image.slidesharecdn.com/compuertaslgicasnovideos-150515201433-lva1-app6892/95/compuertas-lgicas-nor-xor-nand-xnor-16-638.jpg?cb=1431721019>
- Microchip Technology, Inc. (2015). <http://www.microchip.com/>. Obtenido de
<http://ww1.microchip.com/downloads/en/DeviceDoc/40001291H.pdf>
- MikroElektronika. (18 de febrero de 2017). *Microcontroladores PIC – Programación en C con ejemplos*. Obtenido de 3.1 Características Básicas del PIC16f887:
<https://www.google.com.co/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&ua>

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ct=8&ved=0ahUKEwiC0-HutebTAhVB7yYKHb64AeAQjB0IBg&url=https%3A%2F%2Flearn.mikroe.com%2Febooks%2Fmicrocontroladorespicc%2Fchapter%2Fcaracteristicas-basicas-del-pic16f887%2F&psig=

Molina Marticorena, J. (10 de Octubre de 2008). *Compuertas Logicas*. Obtenido de http://www.profesormolina.com.ar/electronica/componentes/int/comp_log/image002.png

Molina Marticorena, J. (10 de Octubre de 2008). *Compuertas Logicas*. Obtenido de http://www.profesormolina.com.ar/electronica/componentes/int/comp_log/image004.png

Molina Marticorena, J. L. (10 de Octubre de 2008). *Compuertas Logicas*. Obtenido de http://www.profesormolina.com.ar/electronica/componentes/int/comp_log/image006.png

Molina Marticorena, J. L. (10 de Octubre de 2008). *Compuertas Logicas*. Obtenido de http://www.profesormolina.com.ar/electronica/componentes/int/comp_log/image010.png

Molina Marticorena, J. L. (10 de Octubre de 2008). *Compuertas Logicas*. Obtenido de http://www.profesormolina.com.ar/electronica/componentes/int/comp_log/image012.png

Molina Marticorena, J. L. (10 de Octubre de 2008). *www.profesormolina.com.ar*, San Juan , Argentina. Obtenido de COMPUERTAS LÓGICAS: http://www.profesormolina.com.ar/electronica/componentes/int/comp_log.htm

Oscar Lian. (07 de Octubre de 2013). Obtenido de HOW TO USE MOSFET – BEGINNER’S TUTORIAL: https://oscarliang.com/ctt/uploads/2013/10/electronics_mosfet_schematic.jpg

Poole, I. (n.d.). *Electronics components*. Retrieved Febrero 17, 2017, from What is a TRIAC - Tutorial: <http://www.radio-electronics.com/info/data/semicond/triac/triac-equivalent-function.gif>

Rodolfo, S. P., Hernández Vázquez, Á. F., & Resendiz Aguilar, V. E. (06 de Junio de 2010). <https://sorprerod.wikispaces.com/>. Obtenido de <https://sorprerod.wikispaces.com/file/view/tesina123.pdf>

Rojas Alzate , K. (2017). *Chequeador de Componentes Electronicos*. Tesis de grado, Instituto Tecnológico Metropolitano, Medellin.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Salinas Delgado, P. S., & Zúñiga Huertas, S. (7 de Enero de 2016).

<http://repositorioacademico.upc.edu.pe/>. Obtenido de
<http://repositorioacademico.upc.edu.pe/upc/handle/10757/593059>

Schiavon, I. M. (1997). <http://www.fceia.unr.edu.ar/eca1/Bibliografia.html>. Obtenido de
<http://www.fceia.unr.edu.ar/eca1/files/teorias/TransistoresdeEfectoDeCampo.pdf>

Scribd Inc. (s.f.). <https://es.scribd.com/>. Obtenido de https://es.scribd.com/doc/1011540/El-transistor-de-Efecto-de-Campo?__cache_revision=1230315207&__user_id=%E2%80%90&enable_docview_caching=1

Tocci, R. J., Widmer, N. S., & Moss, G. L. (2007). *Sistemas digitales: comienzos y aplicaciones*. Pearson Education.

Universidad Nacional Autonoma de Mexico, Facultad de Ingenieria. (18 de Noviembre de 2010).
<http://www.ingenieria.unam.mx>. Obtenido de http://www.ingenieria.unam.mx/crofi/wp-content/uploads/Descargas/Datasheet/Displays/display_de_7_segmentos.pdf

VIDEOSISTEMAS LTD. Videosistemas Corporation Int. (07 de Octubre de 2011).
<http://www.videosistemas.com/proyectos.htm>. Obtenido de
<http://www.videosistemas.com/proyectos/tutorial-transistores.pdf>

Villalba Madrid, G., & Zamora Izquierdo, M. A. (Septiembre de 2008). <http://ocw.um.es/>. Obtenido de <http://ocw.um.es/ingenierias/tecnologia-y-sistemas-electronicos/material-de-clase>:
<http://ocw.um.es/ingenierias/tecnologia-y-sistemas-electronicos/material-de-clase-1/tema-4.-transistores-de-efecto-campo.pdf>

www.neoteo.com. (31 de Marzo de 2009). Obtenido de Midiendo Diodos y Transistores:
http://newimgwp.s3.amazonaws.com/6D12_1024_23479.jpg

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APÉNDICE

APENDICE A GASTOS

Para realizar una estimación del costo económico del proyecto, se han dividido los gastos en tres grupos.

Recursos: Software y programación, Software utilizado y programación del código necesario para llevar a cabo el proyecto.

Componentes y Fabricación de la placa PCB: Costo de los componentes del circuito impreso y el costo de la fabricación del mismo.

Recursos humanos: mano de obra necesaria para realizar el proyecto.

Costo de Software y programación

Dado que el software usado es PIC C Compiler, y este se encuentra instalado en los laboratorios de Mecatrónica y las aulas auxiliares, no genera ningún costo la obtención del Software.

Como el código fue echo en las instalaciones en el tiempo libre del estudiante, construyendo así el proyecto final, y en consecuencia como es un aporte para la institución, no tiene ningún costo.

Costo de componentes y circuito impreso

Para realizar el diseño del circuito impreso, se utilizó Ares que viene integrado en el ISIS Proteus (software utilizado para realizar las simulaciones), Como el fabricante de placas no recibía el archivo en este lenguaje, se realizó otro diseño especificado por el fabricante, compuesto por diversos fotolitos.

La impresión de la placa tuvo un costo de **\$60.000** pesos colombianos debido a su complejidad y su tamaño.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Componentes Electrónicos

Concepto	Cantidad	Precio Unidad	Precio total
Resistencias	25	4paquetes de 1500 (Paquete de diferentes calibres)	6000
Condensadores Electrolíticos	2	450	700
Condensadores Cerámicos	5	450	2250
Cristal 10MHz	1	8000	8000
Led	10	300	3000
Microcontrolador PIC 16F887	1	14000	14000
Pantalla LCD LMO 16L	1	16000	16000
Regulador de voltaje 7805	1	3000	3000
Potenciómetro 10KOhm	1	3500	3500
Pulsadores NO y NC	4	300	1200
Selector tipo codo	1	500	500
Cable	1mt	2000	2000
TOTA			60.150

Recursos humanos: mano de obra necesaria para realizar el proyecto.

Cabe anotar que como es una contribución a la institución y a los laboratorios, el costo de los recursos humanos corresponderá únicamente a la elaboración del circuito impreso, ya que fue hecho por personal externo, por ende, solo se adjuntaran las horas trabajadas en dicho proyecto, que son 64.

Costo total del proyecto

CONCEPTO	COSTO
Recursos Software y programación	0,0
Componentes y fabricación del circuito impreso	60150
Recursos Humanos	0
TOTAL	120150

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APENDICE B MIRCOCNTROLADORES, TEORIA Y PROGRAMA

Estructura de un programa

PIC C Compiler es un inteligente y muy optimizado compilador C que contienen operadores estándar del lenguaje C y funciones incorporados en bibliotecas que son específicas a los registros de PIC, proporcionando a los desarrolladores una herramienta poderosa para el acceso al hardware las funciones del dispositivo desde el nivel de lenguaje C. El compilador CCS contiene más de 307 funciones integradas que simplifiquen el acceso al hardware, mientras que la producción eficiente y altamente optimizado código. Se incluyen funciones de hardware del dispositivo de características tales como: (Matric & Andric, 2000, 3a ed.)

- * Temporizadores y módulos PWM
- * Convertidores A / D
- * De datos on-chip EEPROM
- * LCD controladores
- * Memoria externa, buses
- * Entre otras...

Para escribir un programa en c con el CCS C se deben tener en cuenta una serie de elementos básicos de su estructura.

- **Directivas de procesado:** Controlan la conversión del programa a código de maquina por parte del compilador.
- **Programas o Funciones:** Conjunto de instrucciones. Puede haber uno o varios; en cualquier caso, siempre debe haber uno definido como principal mediante la inclusión de la llamada *main ()*.
- **Instrucciones:** Indican como debe comportarse el PIC en todo momento.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- **Comentarios:** Permiten describir lo que significa cada línea del programa, siempre inicia con //. (Lajara Vizcaino & Pelegrí Sebastián, 2007, 1ra ed.)

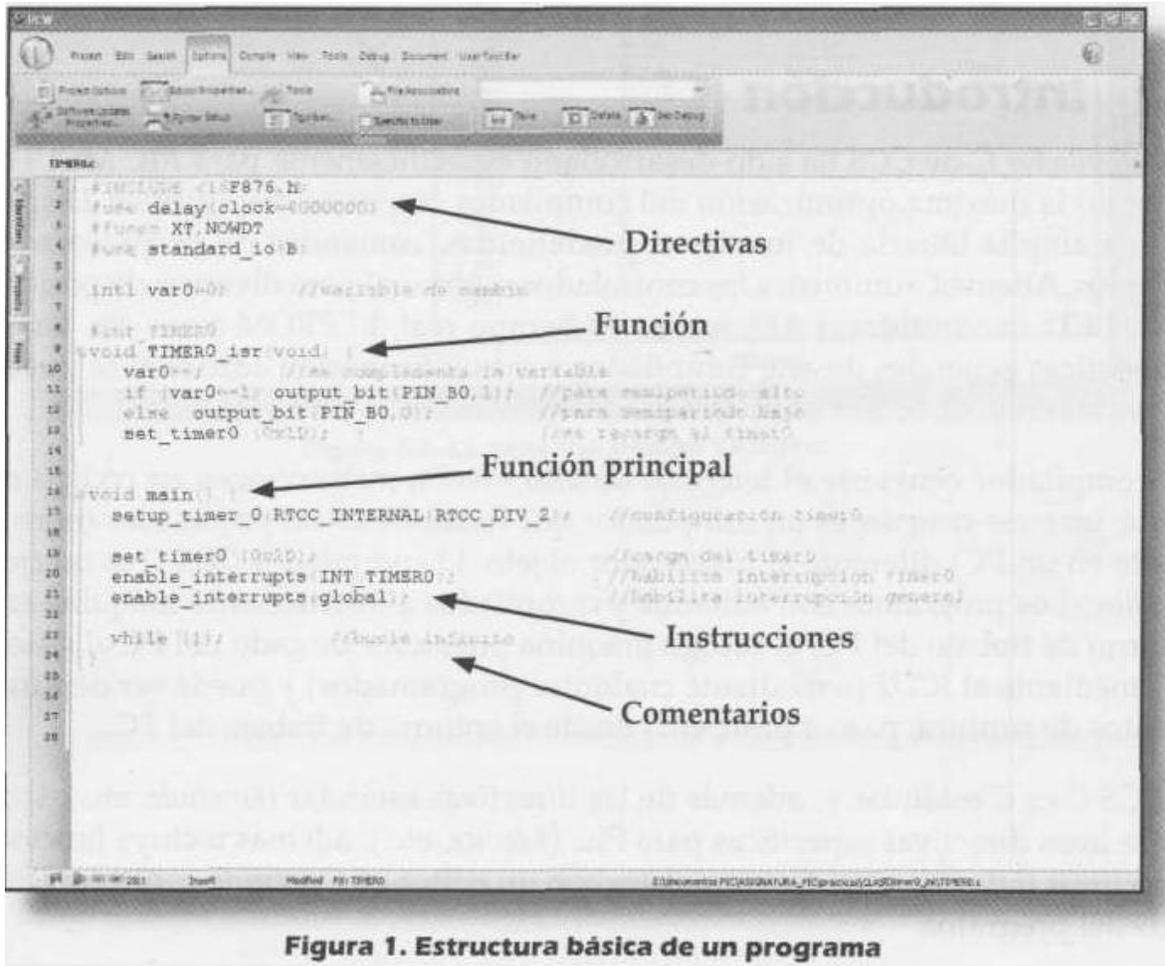


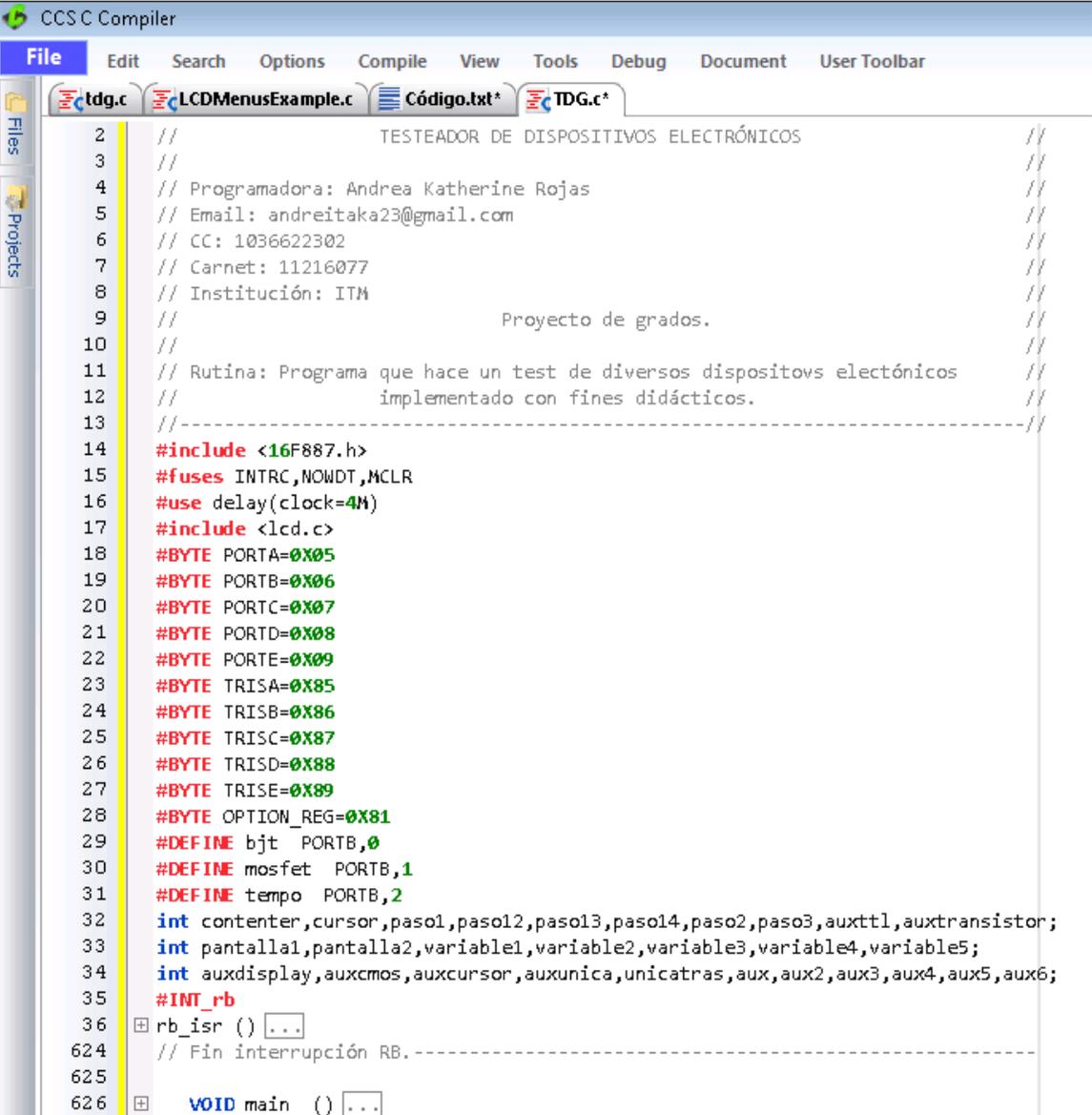
Figura 1. Estructura básica de un programa

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Código fuente del microcontrolador

El microcontrolador es el encargado de recibir la señal de control proveniente de la TAD. Para poder realizar esto se construyó un programa en lenguaje C el cual se encuentra descargado, en el microcontrolador. A continuación, se muestra el código del programa:

Directivas



```

CCS C Compiler
File Edit Search Options Compile View Tools Debug Document User Toolbar
tdg.c LCDMenuExample.c Código.txt* TDG.c*
2 // TESTEADOR DE DISPOSITIVOS ELECTRÓNICOS //
3 // // //
4 // Programadora: Andrea Katherine Rojas //
5 // Email: andreitaka23@gmail.com //
6 // CC: 1036622302 //
7 // Carnet: 11216077 //
8 // Institución: ITM //
9 // // //
10 // Proyecto de grados. //
11 // // //
12 // Rutina: Programa que hace un test de diversos dispositivos electrónicos //
13 // implementado con fines didácticos. //
14 //-----//
14 #include <16F887.h>
15 #fuses INTRC,NOVDT,MCLR
16 #use delay(clock=4M)
17 #include <lcd.c>
18 #BYTE PORTA=0X05
19 #BYTE PORTB=0X06
20 #BYTE PORTC=0X07
21 #BYTE PORTD=0X08
22 #BYTE PORTE=0X09
23 #BYTE TRISA=0X85
24 #BYTE TRISB=0X86
25 #BYTE TRISC=0X87
26 #BYTE TRISD=0X88
27 #BYTE TRISE=0X89
28 #BYTE OPTION_REG=0X81
29 #DEFINE bjt PORTB,0
30 #DEFINE mosfet PORTB,1
31 #DEFINE tempo PORTB,2
32 int contenter,cursor,paso1,paso2,paso3,paso4,paso5,paso6,auxttl,auxtransistor;
33 int pantalla1,pantalla2,variable1,variable2,variable3,variable4,variable5;
34 int auxdisplay,auxcmos,auxcursor,auxunica,unicatras,aux,aux2,aux3,aux4,aux5,aux6;
35 #INT_rb
36 rb_isr () ...
624 // Fin interrupción RB.-----//
625
626 VOID main () ...

```

Función //Interrupción por RB4 Botón arriba.-----

```

CCSC Compiler
File Edit Search Options Compile View Tools Debug Document User Toolbar
tdg.c LCDMenuExample.c Código.txt TDG.c
36 rb_isr ()
37 {
38     byte changes;
39     byte last_b;
40     changes = last_b ^ PORTB;
41     last_b = PORTB;
42     //Interrupcion por RB4 Botón arriba.-----
43     IF (bit_test (changes, 4)&& ! bit_test (last_b, 4))
44     {
45         IF (unicatras == 1)
46         {
47             IF (auxunica == 2)
48             {
49                 // Código para la pantalla 1.-----
50                 IF (pasol != 1 &&contenter == 1)
51                 {
52                     IF (cursor == 1&&auxcursor == 1)
53                     {
54                         cursor++;
55                         lcd_gotoxy (1, cursor);
56                     }
57                     IF (cursor == 2&&auxcursor == 2)
58                     {
59                         cursor = cursor - 1;
60                         lcd_gotoxy (1, cursor);
61                     }
62                     IF (cursor == 1)
63                     {
64                         auxcursor = 1;
65                     }
66                     IF (cursor == 2)
67                     {
68                         auxcursor = 2;
69                     }
70                     pasol = pasol - 1;
71                     IF (pasol == 0)
72                     {
73                         pasol = 1;
74                     }
75                     pantallal = pantallal - 1;
76                     IF (pantallal == 0)
77                     {
78                         pantallal = 1;
79                     }
80                     IF (pantallal == 2)
81                     {
82                         aux = 1;
83                         variable1 = 0;
84                     }
85                 }
86                 // Codigo para la pantalla 12.-----
87                 IF (pasol2 != 1 &&contenter == 2&& (auxtransistor == 0|auxdisplay == 0))
88                 {
89                     IF (cursor == 1&&auxcursor == 1)
90                     {
91                         cursor++;
92                         lcd_gotoxy (1, cursor);
93                     }
94                     IF (cursor == 2&&auxcursor == 2)
95                     {
96                         cursor = cursor - 1;
97                         lcd_gotoxy (1, cursor);
98                     }
99                     IF (cursor == 1)
100                    {
101                        auxcursor = 1;
102                    }
103                    IF (cursor == 2)
104                    {
105                        auxcursor = 2;
106                    }
107                    pasol2 = pasol2 - 1;
108                    IF (pasol2 == 0)
109                    {
110                        pasol2 = 1;

```

```

111     }
112   }
113   //Codigo para la pantalla 13.-----
114   IF (pasol3 != 1 &&contenter == 3&&auxttl == 1)
115   {
116     IF (cursor == 1&&auxcursor == 1)
117     {
118       cursor++;
119       lcd_gotoxy (1, cursor);
120     }
121     IF (cursor == 2&&auxcursor == 2)
122     {
123       cursor = cursor - 1;
124       lcd_gotoxy (1, cursor);
125     }
126     IF (cursor == 1)
127     {
128       auxcursor = 1;
129     }
130     IF (cursor == 2)
131     {
132       auxcursor = 2;
133     }
134     pasol3 = pasol3 - 1;
135     IF (pasol3 == 0)
136     {
137       pasol3 = 1;
138     }
139     pantalla2 = pantalla2 - 1;
140     IF (pantalla2 == 0)
141     {
142       pantalla2 = 1;
143     }
144     IF (pantalla2 == 2)
145     {
146       aux3 = 1;
147       variable3 = 0;
148     }
149     IF (pantalla2 == 6)
150     {
151       aux3 = 1;
152       variable3 = 2;
153     }
154     IF (pantalla2 == 4)
155     {
156       aux3 = 1;
157       variable3 = 1;
158     }
159   }
160   //Codigo para la pantalla 14.-----
161   IF (pasol4 != 1 &&contenter == 3&&auxcmos == 1)
162   {
163     IF (cursor == 1&&auxcursor == 1)
164     {
165       cursor++;
166       lcd_gotoxy (1, cursor);
167     }
168     IF (cursor == 2&&auxcursor == 2)
169     {
170       cursor = cursor - 1;
171       lcd_gotoxy (1, cursor);
172     }
173     IF (cursor == 1)
174     {
175       auxcursor = 1;
176     }
177     IF (cursor == 2)
178     {
179       auxcursor = 2;
180     }
181     pasol4 = pasol4 - 1;
182     IF (pasol4 == 0)
183     {
184       pasol4 = 1;
185     }
186     pantalla2 = pantalla2 - 1;

```

```

187     IF (pantalla2 == 0)
188     {
189         pantalla2 = 1;
190     }
191     IF (pantalla2 == 2)
192     {
193         aux3 = 1;
194         variable3 = 0;
195     }
196     IF (pantalla2 == 6)
197     {
198         aux3 = 1;
199         variable3 = 2;
200     }
201     IF (pantalla2 == 4)
202     {
203         aux3 = 1;
204         variable3 = 1;
205     }
206     }
207     //Codigo para la pantalla 2.-----
208     IF (paso2 != 1 &&contenter == 2&&auxtransistor == 1)
209     {
210         IF (cursor == 1&&auxcursor == 1)
211         {
212             cursor++;
213             lcd_gotoxy (1, cursor);
214         }
215         IF (cursor == 2&&auxcursor == 2)
216         {
217             cursor = cursor - 1;
218             lcd_gotoxy (1, cursor);
219         }
220         IF (cursor == 1)
221         {
222             auxcursor = 1;
223         }
224         IF (cursor == 2)
225         {
226             auxcursor = 2;
227         }
228         paso2 = paso2 - 1;
229     }
230     IF (paso2 == 0)
231     {
232         paso2 = 1;
233     }
234     }
235     //Codigo para la pantalla 3.-----
236     IF (paso3 != 1 &&contenter == 2&&auxdisplay == 1)
237     {
238         IF (cursor == 1&&auxcursor == 1)
239         {
240             cursor++;
241             lcd_gotoxy (1, cursor);
242         }
243         IF (cursor == 2&&auxcursor == 2)
244         {
245             cursor = cursor - 1;
246             lcd_gotoxy (1, cursor);
247         }
248         IF (cursor == 1)
249         {
250             auxcursor = 1;
251         }
252         IF (cursor == 2)
253         {
254             auxcursor = 2;
255         }
256         paso3 = paso3 - 1;
257         IF (paso3 == 0)
258         {
259             paso3 = 1;
260         }
261     }
262     }

```

//Interrupción por RB5 Botón atrás.-----

```

265 //Interrupción por RB5 Botón atrás.-----
266 IF (bit_test (changes, 5)&& ! bit_test (last_b, 5))
267 {
268     IF (auxunica == 2)
269     {
270         IF (contenter == 4)
271         {
272             aux3 = 1;
273             variable3 = 0;
274             paso13 = 1;
275             paso12 = 1;
276             paso14 = 1;
277             cursor = 1;
278             auxcursor = 1;
279             pantalla2 = 1;
280         }
281         IF (contenter == 3)
282         {
283             IF (auxttl == 1||auxcmos == 1)
284             {
285                 aux2 = 1;
286             }
287             IF (auxtransistor == 1)
288             {
289                 aux4 = 1;
290             }
291             IF (auxdisplay == 1)
292             {
293                 aux5 = 1;
294             }
295             variable2 = 0;
296             paso13 = 1;
297             paso12 = 1;
298             paso14 = 1;
299             cursor = 1;
300             auxcursor = 1;
301             auxttl = 0;
302             auxcmos = 0;
303             pantalla2 = 1;
304             paso2 = 1;
305             paso3 = 1;
306         }
307         IF (contenter == 2)
308         {
309             aux = 1;
310             variable1 = 0;
311             paso13 = 1;
312             paso12 = 1;
313             paso14 = 1;
314             paso1 = 1;
315             cursor = 1;
316             auxcursor = 1;
317             pantalla1 = 1;
318             auxtransistor = 0;
319             auxdisplay = 0;
320         }
321         contenter = contenter - 1;
322         IF (contenter == 0)
323         {
324             contenter = 1;
325         }
326         IF (unicatras == 2)
327         {
328             unicatras = 1;
329         }
330     }
331 }
332 //Interrupcion por RB6 Botón abajo.-----
333 IF (bit_test (changes, 6)&& ! bit_test (last_b, 6)) ...
546 //Interrupcion por RB7 Botón entrar.-----
547 IF (bit_test (changes, 7)&& ! bit_test (last_b, 7)) ...
621     delay_ms (100);
622 }
623 // Fin interrupción RB.-----
624
625 VOID main () ...

```

//Interrupción por RB6 Botón abajo.-----

```

332 //Interrupcion por RB6 Botón abajo.-----
333 IF (bit_test (changes, 6)&& ! bit_test (last_b, 6))
334 {
335     IF (unicatras == 1)
336     {
337         IF (auxunica == 2)
338         {
339             //Codigo para la pantalla 1.-----
340             IF (pasol != 4 &&contenter == 1)
341             {
342                 IF (cursor == 1&&auxcursor == 1)
343                 {
344                     cursor++;
345                     lcd_gotoxy (1, cursor);
346                 }
347                 IF (cursor == 2&&auxcursor == 2)
348                 {
349                     cursor = cursor - 1;
350                     lcd_gotoxy (1, cursor);
351                 }
352                 IF (cursor == 1)
353                 {
354                     auxcursor = 1;
355                 }
356                 IF (cursor == 2)
357                 {
358                     auxcursor = 2;
359                 }
360                 pasol = pasol + 1;
361
362                 IF (pasol == 5)
363                 {
364                     pasol = 4;
365                 }
366                 pantallal = pantallal + 1;
367                 IF (pantallal == 3)
368                 {
369                     aux = 1;
370
371                     variablel = 1;
372                 }
373             }
374             //Código para la pantalla 12.-----
375             IF (pasol2 != 2 &&contenter == 2&&auxtransistor == 0&&auxdisplay == 0)
376             {
377                 IF (cursor == 1&&auxcursor == 1)
378                 {
379                     cursor++;
380                     lcd_gotoxy (1, cursor);
381                 }
382                 IF (cursor == 2&&auxcursor == 2)
383                 {
384                     cursor = cursor - 1;
385                     lcd_gotoxy (1, cursor);
386                 }
387                 IF (cursor == 1)
388                 {
389                     auxcursor = 1;
390                 }
391                 IF (cursor == 2)
392                 {
393                     auxcursor = 2;
394                 }
395                 pasol2 = pasol2 + 1;
396
397                 IF (pasol2 == 3)
398                 {
399                     pasol2 = 2;
400                 }
401             }
402             //Código para la pantalla 13.-----
403             IF (pasol3 != 7 &&contenter == 3&&auxttl == 1)
404             {
405                 IF (cursor == 1&&auxcursor == 1)
406                 {
407                     cursor++;
408                     lcd_gotoxy (1, cursor);
409                 }
410             }
411         }
412     }
413 }

```

```

408     }
409     IF (cursor == 2&&auxcursor == 2)
410     {
411         cursor = cursor - 1;
412         lcd_gotoxy (1, cursor);
413     }
414     IF (cursor == 1)
415     {
416         auxcursor = 1;
417     }
418     IF (cursor == 2)
419     {
420         auxcursor = 2;
421     }
422     paso13 = paso13 + 1;
423     IF (paso13 == 8)
424     {
425         paso13 = 7;
426     }
427     pantalla2 = pantalla2 + 1;
428     IF (pantalla2 == 3)
429     {
430         aux3 = 1;
431         variable3 = 1;
432     }
433     IF (pantalla2 == 5)
434     {
435         aux3 = 1;
436         variable3 = 2;
437     }
438     IF (pantalla2 == 7)
439     {
440         aux3 = 1;
441         variable3 = 3;
442     }
443     }
444     //Código para la pantalla 14.-----
445     IF (paso14 != 7 &&contenter == 3&&auxcmos == 1)
446     {
447         IF (cursor == 1&&auxcursor == 1)
448         {
449             cursor++;
450             lcd_gotoxy (1, cursor);
451         }
452         IF (cursor == 2&&auxcursor == 2)
453         {
454             cursor = cursor - 1;
455             lcd_gotoxy (1, cursor);
456         }
457         IF (cursor == 1)
458         {
459             auxcursor = 1;
460         }
461         IF (cursor == 2)
462         {
463             auxcursor = 2;
464         }
465         paso14 = paso14 + 1;
466     }
467     IF (paso14 == 8)
468     {
469         paso14 = 7;
470     }
471     pantalla2 = pantalla2 + 1;
472     IF (pantalla2 == 3)
473     {
474         aux3 = 1;
475         variable3 = 1;
476     }
477     IF (pantalla2 == 5)
478     {
479         aux3 = 1;
480         variable3 = 2;
481     }
482     IF (pantalla2 == 7)
483     {

```

```

484         aux3 = 1;
485         variable3 = 3;
486     }
487 }
488 //Código para la pantalla 2.-----
489 IF (paso2 != 2 &&contenter == 2&&auxtransistor == 1)
490 {
491     IF (cursor == 1&&auxcursor == 1)
492     {
493         cursor++;
494         lcd_gotoxy (1, cursor);
495     }
496     IF (cursor == 2&&auxcursor == 2)
497     {
498         cursor = cursor - 1;
499         lcd_gotoxy (1, cursor);
500     }
501     IF (cursor == 1)
502     {
503         auxcursor = 1;
504     }
505     IF (cursor == 2)
506     {
507         auxcursor = 2;
508     }
509     paso2 = paso2 + 1;
510     IF (paso2 == 3)
511     {
512         paso2 = 2;
513     }
514 }
515 //Código para la pantalla 3.-----
516 IF (paso3 != 2 &&contenter == 2&&auxdisplay == 1)
517 {
518     IF (cursor == 1&&auxcursor == 1)
519     {
520         cursor++;
521         lcd_gotoxy (1, cursor);
522     }
523     IF (cursor == 2&&auxcursor == 2)
524     {
525         cursor = cursor - 1;
526         lcd_gotoxy (1, cursor);
527     }
528     IF (cursor == 1)
529     {
530         auxcursor = 1;
531     }
532     IF (cursor == 2)
533     {
534         auxcursor = 2;
535     }
536     paso3 = paso3 + 1;
537
538     IF (paso3 == 3)
539     {
540         paso3 = 2;
541     }
542 }
543 }
544 }
545 }
546 //Interrupcion por RB7 Botón entrar.-----
547 IF (bit_test (changes, 7)&& ! bit_test (last_b, 7)) {
621     delay_ms (100);
622 }
623 // Fin interrupción RB.-----
624
625 VOID main () {

```

//Interrupción por RB7 Botón entrar.-----

```

546 //Interrupcion por RB7 Botón entrar.-----
547 IF (bit_test (changes, 7)&& ! bit_test (last_b, 7))
548 {
549   IF (unicatras == 1)
550   {
551     IF (pasol == 1&&contenter == 1)
552     {
553       aux2 = 1;
554     }
555     IF (pasol2 == 1&&contenter == 2&& auxtransistor == 0&&auxdisplay == 0)
556     {
557       aux3 = 1;
558       variable3 = 0;
559       auxttl = 1;
560     }
561     // Actuador final AND, NAND, OR, NOR, XOR, XNOR, NOT
562     IF ((pasol3 == 1||pasol3 == 2||pasol3 == 3||pasol3 == 4||pasol3 == 5|pasol3 == 6|pasol3 == 7)&&contenter == 3&&auxttl == 1)
563     {
564       aux6 = 1;
565     }
566     IF (pasol2 == 2&&contenter == 2&& auxtransistor == 0&&auxdisplay == 0)
567     {
568       aux3 = 1;
569       variable3 = 0;
570       auxcmos = 1;
571       cursor = 1;
572       auxcursor = 1;
573     }
574     // Actuador final AND, NAND, OR, NOR, XOR, XNOR, NOT
575     IF ((pasol4 == 1||pasol4 == 2||pasol4 == 3||pasol4 == 4||pasol4 == 5|pasol4 == 6|pasol4 == 7)&&contenter == 3&&auxcmos == 1)
576     {
577       aux6 = 1;
578     }
579     IF (pasol == 2&&contenter == 1)
580     {
581       aux4 = 1;
582       auxtransistor = 1;
583       cursor = 1;
584       auxcursor = 1;
585     }
586     IF (pasol == 1&&contenter == 2&&auxtransistor == 1)
587     {
588       aux6 = 1;
589     }
590     IF (pasol == 2&&contenter == 2&&auxtransistor == 1)
591     {
592       aux6 = 1;
593     }
594     IF (pasol3 == 1&&contenter == 2&&auxdisplay == 1)
595     {
596       aux6 = 1;
597     }
598     IF (pasol3 == 2&&contenter == 2&&auxdisplay == 1)
599     {
600       aux6 = 1;
601     }
602     IF (pasol == 3&&contenter == 1)
603     {
604       aux5 = 1;
605       auxdisplay = 1;
606     }
607     IF (pasol == 4&&contenter == 1)
608     {
609       aux6 = 1;
610     }
611     IF (auxunica == 1)
612     {
613       auxunica++;
614       aux = 1;
615       pasol = 1;
616     }
617     cursor = 1;
618     auxcursor = 1;
619     lcd_gotoxy (1, cursor);
620     contenter++;
621   }
622 }
623 delay_ms (100);
624 }

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Funcion Principal VOID main ()-----

```

627  VOID main ()
628  {
629      TRISA = 0B00000000;
630      TRISB = 0B11110000;
631      TRISC = 0B00000000;
632      TRISD = 0B00000000;
633      OPTION_REG = 0B01111111;
634      TRISE = 0xff;
635      lcd_init ();
636      lcd_gotoxy (4, 1);
637      printf (lcd_putc, "BIENVENIDO");
638      delay_ms (1000);
639      printf (lcd_putc, "\f");
640      delay_ms (500);
641      lcd_gotoxy (4, 1);
642      printf (lcd_putc, " - ITM - ");
643      lcd_gotoxy (2, 2);
644      printf (lcd_putc, "Katherine Rojas");
645      delay_ms (2000);
646      printf (lcd_putc, "\f");
647      lcd_gotoxy (7, 1);
648      printf (lcd_putc, "Menu");
649      lcd_gotoxy (1, 2);
650      printf (lcd_putc, "Presiona entrar");
651      delay_ms (300);
652      lcd_cursor_on (TRUE);
653      enable_interrupts (INT_RB);
654      enable_interrupts (GLOBAL);
655      PORTA = 0B10110000;
656      PORTC = 0B11111111;
657      BIT_CLEAR (bjt);
658      BIT_CLEAR (mosfet);
659      BIT_CLEAR (tempo);
660      pantalla2 = 1;
661      pantalla1 = 1;
662      variable1 = 0;
663      variable2 = 0;
664      variable3 = 0;
665      variable4 = 0;
666      variable5 = 0;
667      cursor = 1;
668      contenter = 0;
669      auxcursor = 1;
670      auxttl = 0;
671      auxcmos = 0;
672      auxtransistor = 0;
673      auxdisplay = 0;
674      paso1 = 0;
675      paso2 = 1;
676      paso3 = 1;
677      paso4 = 1;
678      paso2 = 1;
679      paso3 = 1;
680      auxunica = 1;
681      unicatras = 1;
682      aux = 2;
683      aux2 = 2;
684      aux3 = 2;
685      aux4 = 2;
686      aux5 = 2;
687      aux6 = 2;
688      WHILE (true)
689      {
690          IF (auxunica == 2)
691          {
692              IF (aux == 1)
693              {
694                  SWITCH (variable1)
695                  {
696                      CASE 0:
697                          printf (lcd_putc, "\f");
698                          lcd_gotoxy (2, 1);
699                          printf (lcd_putc, " - Computas");
700                          lcd_gotoxy (2, 2);
701                          printf (lcd_putc, " - Transistores");
702                          IF (pantalla1 == 2)
703                          {
704                              lcd_gotoxy (1, 2);

```

```

705     }
706     ELSE
707     {
708         lcd_gotoxy (1, 1);
709     }
710     aux = 2;
711     BREAK;
712     CASE 1:
713     printf (lcd_putc, "\f");
714     lcd_gotoxy (2, 1);
715     printf (lcd_putc, " - Displays");
716     lcd_gotoxy (2, 2);
717     printf (lcd_putc, " - Temporizador");
718     lcd_gotoxy (1, 1);
719     aux = 2;
720     BREAK;
721     }
722 }
723 IF (aux2 == 1)
724 {
725     SWITCH (variable2)
726     {
727         CASE 0:
728         printf (lcd_putc, "\f");
729         lcd_gotoxy (2, 1);
730         printf (lcd_putc, " - TTL");
731         lcd_gotoxy (2, 2);
732         printf (lcd_putc, " - CMOS");
733         lcd_gotoxy (1, 1);
734         aux2 = 2;
735         BREAK;
736     }
737 }
738 IF (aux3 == 1)
739 {
740     SWITCH (variable3)
741     {
742         CASE 0:
743         printf (lcd_putc, "\f");
744         lcd_gotoxy (2, 1);
745         printf (lcd_putc, " - AND");
746         lcd_gotoxy (2, 2);
747         printf (lcd_putc, " - NAND");
748         lcd_gotoxy (1, 1);
749         IF (pantalla2 == 2)
750         {
751             lcd_gotoxy (1, 2);
752         }
753         ELSE
754         {
755             lcd_gotoxy (1, 1);
756         }
757         aux3 = 2;
758         BREAK;
759         CASE 1:
760         printf (lcd_putc, "\f");
761         lcd_gotoxy (2, 1);
762         printf (lcd_putc, " - OR");
763         lcd_gotoxy (2, 2);
764         printf (lcd_putc, " - NOR");
765         lcd_gotoxy (1, 1);
766         IF (pantalla2 == 4)
767         {
768             lcd_gotoxy (1, 2);
769         }
770         ELSE {
771             lcd_gotoxy (1, 1);
772         }
773         aux3 = 2;
774         BREAK;
775         CASE 2:
776         printf (lcd_putc, "\f");
777         lcd_gotoxy (2, 1);
778         printf (lcd_putc, " - XOR");
779         lcd_gotoxy (2, 2);
780         printf (lcd_putc, " - XNOR");
781         lcd_gotoxy (1, 1);
782         IF (pantalla2 == 6)

```

```

783     {
784         lcd_gotoxy (1, 2);
785     }
786     ELSE
787     {
788         lcd_gotoxy (1, 1);
789     }
790     aux3 = 2;
791     BREAK;
792     CASE 3:
793     printf (lcd_putc, "\f");
794     lcd_gotoxy (2, 1);
795     printf (lcd_putc, " - NOT");
796     lcd_gotoxy (1, 1);
797     aux3 = 2;
798     BREAK;
799     }
800 }
801 IF (aux4 == 1)
802 {
803     SWITCH (variable4)
804     {
805         CASE 0:
806         printf (lcd_putc, "\f");
807         lcd_gotoxy (2, 1);
808         printf (lcd_putc, " - MOSFET");
809         lcd_gotoxy (2, 2);
810         printf (lcd_putc, " - BJT");
811         lcd_gotoxy (1, 1);
812         aux4 = 2;
813         BREAK;
814     }
815 }
816 IF (aux5 == 1)
817 {
818     SWITCH (variable4)
819     {
820         CASE 0:
821         printf (lcd_putc, "\f");
822         lcd_gotoxy (2, 1);
823         printf (lcd_putc, " - Anodo Comun");
824         lcd_gotoxy (2, 2);
825         printf (lcd_putc, " - Catodo Comun");
826         lcd_gotoxy (1, 1);
827         aux5 = 2;
828         BREAK;
829     }
830 }
831 IF (aux6 == 1)
832 {
833     SWITCH (variable5)
834     {
835         CASE 0:
836         printf (lcd_putc, "\f");
837         lcd_gotoxy (4, 1);
838         printf (lcd_putc, "EJECUTANDO");
839         aux6 = 2;
840         unicas = 2;
841         IF ((paso13 == 1||paso13 == 2||paso13 == 3||paso13 == 5||paso13 == 6)&&contenter == 4&&auxttl == 1)
842         {
843             for (INT x = 0; x < 4; x++)
844             {
845                 PORTA = 0B00000011;
846                 delay_ms (1500);
847                 PORTA = 0B00000000;
848                 delay_ms (500);
849             }
850         }
851         IF (paso13 == 4&&contenter == 4&&auxttl == 1)
852         {
853             for (INT x = 0; x < 4; x++)
854             {
855                 PORTA = 0B00001100;
856                 delay_ms (1500);
857                 PORTA = 0B00000000;
858                 delay_ms (500);
859             }
860         }

```

```

861 IF ((paso13 == 7||paso14 == 7) &&contenter == 4&&(auxttl == 1|auxcmos == 1))
862 {
863     for ( INT x = 0; x < 4; x++)
864     {
865         PORTA = 0B10000000;
866         delay_ms (1500);
867         PORTA = 0B00000000;
868         delay_ms (1500);
869     }
870 }
871 IF ((paso14 == 1||paso14 == 2||paso14 == 3||paso14 == 4||paso14 == 5||paso14 == 6)&&contenter == 4&&auxcmos == 1)
872 {
873     for ( INT x = 0; x < 4; x++)
874     {
875         PORTA = 0B00110000;
876         delay_ms (1500);
877         PORTA = 0B00000000;
878         delay_ms (1500);
879     }
880 }
881 IF (paso2 == 1&&contenter == 3&&auxtransistor == 1)
882 {
883     for ( INT x = 0; x < 4; x++)
884     {
885         BIT_SET (bjt);
886         delay_ms (500);
887         BIT_CLEAR (bjt);
888         delay_ms (500);
889     }
890 }
891 IF (paso2 == 2&&contenter == 3&&auxtransistor == 1)
892 {
893     for ( INT x = 0; x < 4; x++)
894     {
895         BIT_SET (mosfet);
896         delay_ms (500);
897         BIT_CLEAR (mosfet);
898         delay_ms (500);
899     }
900 }
901 IF (paso3 == 1&&contenter == 3&&auxdisplay == 1)
902 {
903     PORTC = 0B00000000;
904     delay_ms (500);
905     PORTC = 0B00000001;
906     delay_ms (500);
907     PORTC = 0B00000010;
908     delay_ms (500);
909     PORTC = 0B00000011;
910     delay_ms (500);
911     PORTC = 0B00000100;
912     delay_ms (500);
913     PORTC = 0B00000101;
914     delay_ms (500);
915     PORTC = 0B00000110;
916     delay_ms (500);
917     PORTC = 0B00000111;
918     delay_ms (500);
919     PORTC = 0B00001000;
920     delay_ms (500);
921     PORTC = 0B00001001;
922     delay_ms (500);
923     PORTC = 0B11111111;
924 }
925 IF (paso3 == 2&&contenter == 3&&auxdisplay == 1)
926 {
927     PORTC = 0B00000000;
928     delay_ms (500);
929     PORTC = 0B00010000;
930     delay_ms (500);
931     PORTC = 0B00100000;
932     delay_ms (500);
933     PORTC = 0B00110000;
934     delay_ms (500);
935     PORTC = 0B01000000;
936     delay_ms (500);
937     PORTC = 0B01010000;
938     delay_ms (500);

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

939          PORTC = 0B01100000;
940          delay_ms (500);
941          PORTC = 0B01110000;
942          delay_ms (500);
943          PORTC = 0B10000000;
944          delay_ms (500);
945          PORTC = 0B10010000;
946          delay_ms (500);
947          PORTC = 0B11111111;
948      }
949      IF (paso1 == 4&&contenter == 2)
950      {
951          BIT_SET (tempo);
952          delay_ms (3000);
953          BIT_CLEAR (tempo);
954      }
955      printf (lcd_putc, "\f");
956      lcd_gotoxy (6, 1);
957      printf (lcd_putc, "LISTO");
958      lcd_gotoxy (2, 2);
959      printf (lcd_putc, "Presiona atras");
960      BREAK;
961  }
962  }
963  }
964  }
965  }

```

FIN DEL PROGRAMA

Compilacion si errores.....

The screenshot shows the IDE interface with the following details:

- Source Code (TDG.c):**

```

1 //-----//
2 //          TESTEADOR DE DISPOSITIVOS ELECTRONICOS          //
3 //-----//
4 // Programadora: Andrea Katherine Rojas                      //
5 // Email: andreakata23@gmail.com                             //
6 // CC: 1836023500                                            //
7 // Carnet: 11216877                                         //
8 // Institución: ITM                                         //
9 //-----//
10 //          Proyecto de grados.                             //
11 //-----//
12 // Rutina: Programa que hace un test de diversos dispositivos //
13 //          implementado con fines didácticos.              //
14 //-----//
15 #include <16F887.h>
16 #include TMR1C, WDTCON, INTCON
17 #include <lcd.h>
18 #include <lcd.h>
19 #include <lcd.h>
20 #include <lcd.h>
21 #include <lcd.h>
22 #include <lcd.h>

```
- Status Bar:** 1.1 | Insert | Pj: TDG | C:\Users\ANDREA\Downloads\TDG.c
- Output Window:**

```

>>> Warning 208: "TDG.c" Line 36(1.7): Function not void and does not return a value: rb_in
>>> Warning 216: "TDG.c" Line 36(4.5): Interrupts disabled during call to prevent reentrancy: (tmr1cb_in)
>>> Warning 216: "TDG.c" Line 36(4.5): Interrupts disabled during call to prevent reentrancy: (lcd_send_byte)
Memory usage: ROM=42%  RAM=20%  DE=0%  4 Warnings
Build Successful

```
- Memory Use:**
 - RAM: 20%
 - ROM: 42%

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

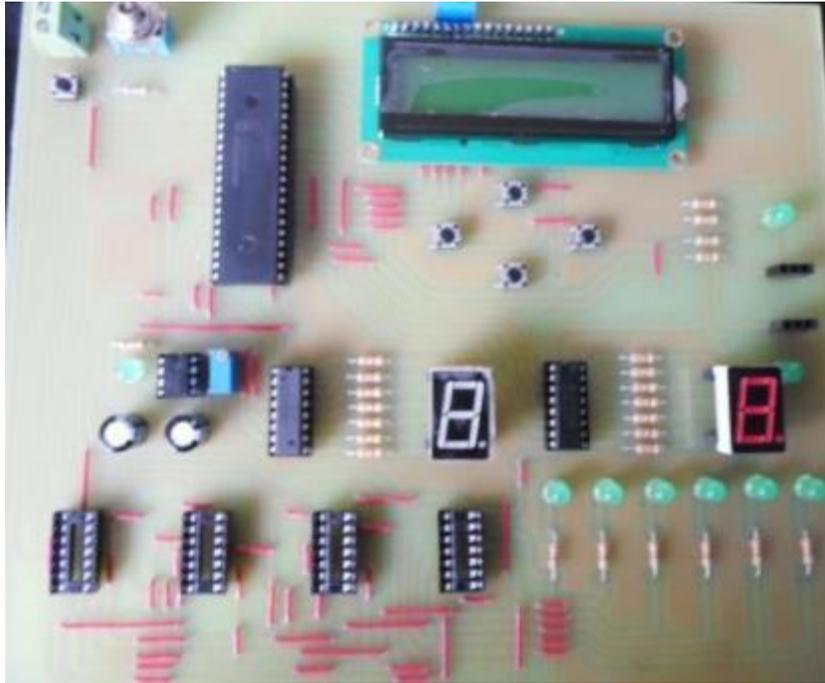
APENDICE C MANUAL DE USO

CONSIDERACIONES BASICAS A LA HORA DE REALIZAR UNA MEDICION EN EL MÓDULO

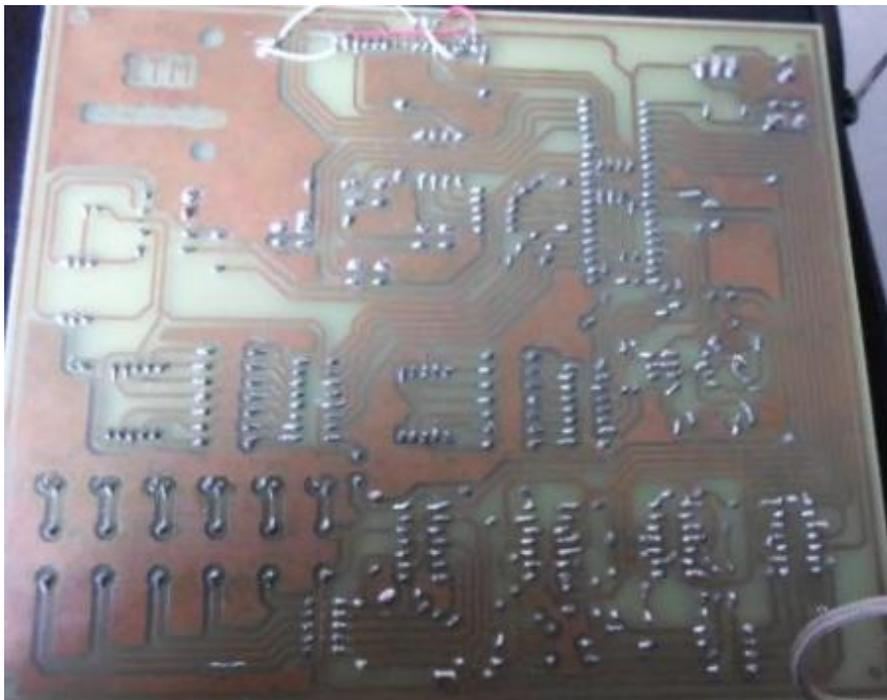
ADVERTENCIA - LEER ANTES DE TRABAJAR

Fallos en la observancia de las siguientes instrucciones pueden ocasionar lesiones en el personal o daños en el módulo.

- No quitar las placas de advertencia o instrucciones de la máquina. Estas placas deben ser legibles en todo momento.
- No utilizar el módulo sin una toma de tierra apropiada para eliminar peligros de choque eléctrico.
- Cuando el módulo está siendo reparada la tensión debe estar desconectada.
- Evitar el contacto directo con las pistas de la tarjeta o alterar las soldaduras de la misma podría acarrear problemas de cortocircuito y problemas en la programación.



(Rojas Alzate , 2017)



(Rojas Alzate , 2017)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Secuencia de encendido y apagado

Para iniciar asegúrese de tener la maquina conectada a una red de 220V / 50- 60Hz, En seguida siga las siguientes instrucciones.

1. Accionar el interruptor ON/OFF y dejarlo en posición ON.



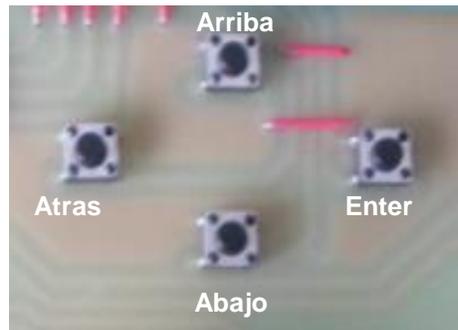
2. Presione enter y seleccione el modo que desee trabajar.



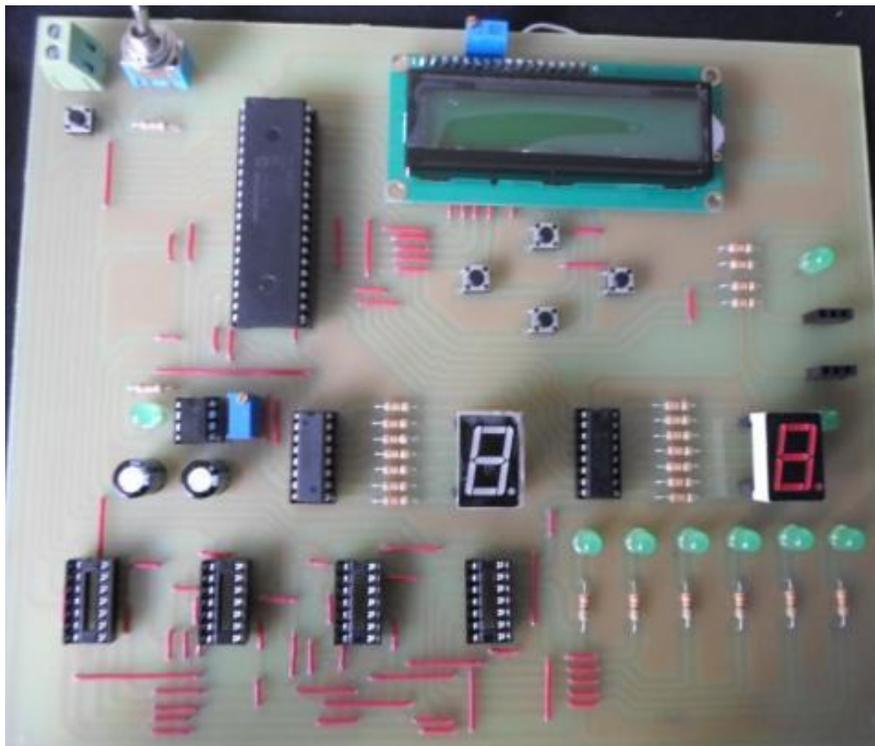
3. Para los modos, es necesario siempre realizar al inicio la búsqueda de cero o posición de referencia.



4. Si desea cambiar el modo de operación de un modo a otro, realice siempre la búsqueda de referencia pulsando arriba o abajo.



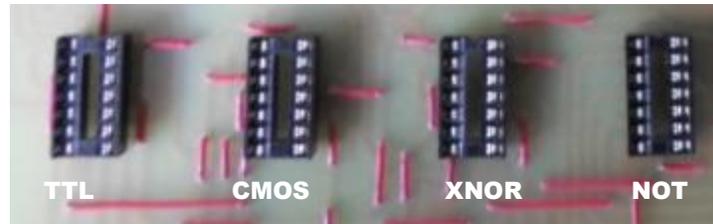
5. Ingrese al componente que desea medir
Recuerde que el componente debe encontrarse en el lugar correcto.



(Rojas Alzate , 2017)

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

PARA MEDIR COMPUERTAS



1. Ingrese al menú y seleccione compuertas con el puntero.



2. A continuación, seleccione el tipo de familia que está utilizando TTL o CMOS



NOTA: En las dos familias encontrara la misma descripción, más las conexiones nunca serán las mismas, por eso siempre verificar la zona donde pone los componentes.

3. Si usted eligió TTL verifique que su componente pertenezca a la siguiente lista,
 - NOT – 74LS04
 - AND – 74LS08
 - NAND – 74LS00

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- OR – 74LS32
- NOR – 74LS02
- XOR – 74LS86
- XNOR – 74LS266

NOTA: recuerde que esta tarjeta posee limitantes, por ende, absténgase de insertar un componente diferente, pues a pesar de que la conexión pueda coincidir podría presentar errores o incluso incurrir en daños tanto del módulo como de su componente.

4. Seleccione con la ayuda del puntero el componente que desea medir, desplazándose con los pulsadores arriba y abajo.



5. Si usted eligió CMOS verifique que su componente pertenezca a la siguiente lista,



- NOT – CD4069
- AND – CD4081
- NAND – CD4011/4093
- OR – CD4071
- NOR – CD4001
- XOR – CD4070/4030

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

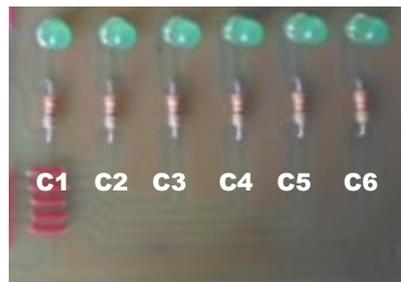
- XNOR – CD4077

NOTA: recuerde que esta tarjeta posee limitantes, por ende, absténgase de insertar un componente diferente, pues a pesar de que la conexión pueda coincidir podría presentar errores o incluso incurrir en daños tanto del módulo como de su componente.

6. Seleccione con la ayuda del puntero el componente que desea medir, desplazándose con los pulsadores arriba y abajo.



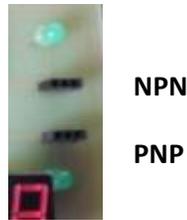
7. A continuación, encenderán los ledes equivalentes a cada compuerta del chip, nótese que se están midiendo componentes cuádruples de dos entradas una salida.



NOTA: Los sockets están debidamente marcados, la compuerta **NOT** se ingresa siempre en el mismo lugar ya que cumple con la misma conexión tanto en **TTL** como en **CMOS**.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

PARA MEDIR TRANSISTORES



1. Ingrese al menú y seleccione transistores con el puntero.



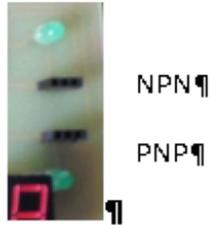
2. A continuación, seleccione el tipo de transistor que desea medir, desplazándose con los pulsadores arriba y abajo



NOTA: recuerde que la conexión para los transistores consta de 3 pines: emisor, base y colector, por ende, la conexión varía en, si son pnp o npn, no obstante, antes de ingresar el componente al módulo procure verificarlo con el multímetro.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- 3.** Nótese que el transistor está actuando como quiche y en el momento que se seleccione procederá a encender las luces correspondientes.

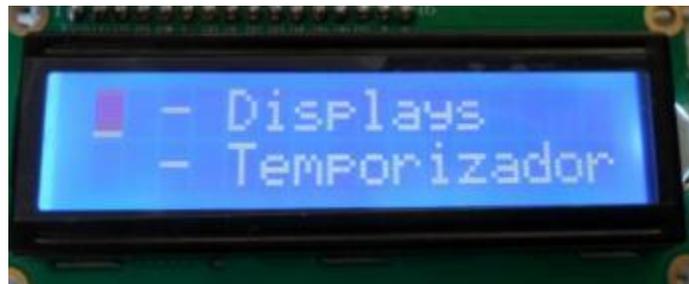


	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

PARA MEDIR DISPLAY SIETE SEGMENTOS



1. Ingrese al menú y seleccione Display con el puntero.



2. A continuación, seleccione el tipo de Display que está utilizando Ánodo Común o Cátodo Común



3. Gracias a la programación de dos circuitos integrados adicionales para accionar el display, el visualizador digital mostrara el número 8. O sea, los siete segmentos activados, ya sea en ánodo común o cátodo común.



	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

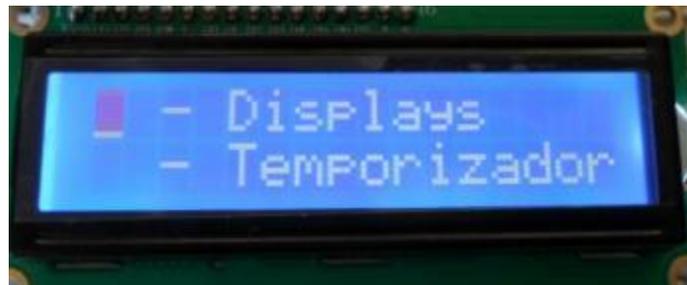
NOTA: Los displays vienen configurados de manera que comparten un punto en común, nótese que se tienen dos ranuras para cada tipo de display, antes de insertar el componente, verificar el tipo con el multímetro, pues podría presentar errores o incluso incurrir en daños tanto del módulo como de su componente.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

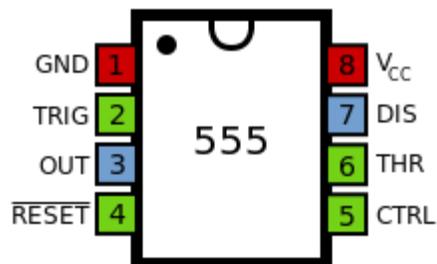
PARA MEDIR TIMER 555



1. Ingrese al menú y seleccione transistores con el puntero.



2. Para realizar la medición del Timer 555, se encenderá una luz parpadeante indicando que nuestra salida se encuentra en buen estado, recuerde que el Timer 555, siempre va a presentar la siguiente configuración, por ende, absténgase de conectar otro componente que no sea compatible a este.



	<p style="text-align: center;">INFORME FINAL DE TRABAJO DE GRADO</p>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

EJECUTANDO

Cuando el estudiante ha escogido su componente a medir, a continuación, el microcontrolador ejecutara la tarea correspondiente a esa área, por ende, absténgase de retirar el microcontrolador con el módulo encendido.



PROCESO FINALIZADO

Una vez el microcontrolador ejecute la tarea y muestre el resultado en los leds y en el display, se habrá finalizado el proceso y usted podrá volver a ejecutar una nueva tarea.



NOTA: Evite retirar el componente en el momento de la medición, pues esto podría incurrir en errores en la ejecución del programa, en daños al módulo o a su componente.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

AYUDAS ADICIONALES

PARA CHEQUEAR TRANSISTORES

El transistor tiene 3 pines, colocas el cable rojo del multímetro en el pin de en medio y la negra a cualquier pin de los extremos, si te da medida es NPN y el que te dé mayor lectura es el emisor,

Si pones el cable negro del multímetro en medio y el rojo a cualquier lado del transistor y te da una medida entonces será PNP

Si te da medida de las 2 formas el transistor no sirve

Observe la referencia del transistor; la mayoría de los transistores que comienzan su referencia con las letras C y D son del tipo NPN, y los que comienzan con las letras A y B suelen ser del tipo PNP.

PARA CHEQUEAR DISPLAY 7 SEGMENTOS

Usa por ejemplo una fuente de 5V en serie con un resistor de unos 220 o 330 ohm. Si cualquier segmento enciende aplicándole el positivo, estando el negativo conectado a una de las patas centrales, es cátodo común. Y viceversa.

Con el multímetro, según qué circuito interno tenga. Prueba de conectar sus puntas al display, la punta roja al pin central que sería el común y la punta negra que sería el negativo a cualquier pin del display podría iluminar apenas los segmentos, esto indicaría que es de ánodo común, si realizas la operación en sentido contrario, conectando el negativo a una de las patas centrales seria cátodo común, aunque en el multímetro te indique overflow, pues la corriente que entrega el multímetro es menor a la que requiere el display o en este caso uno de los segmentos.

