

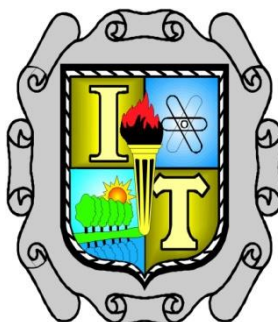


EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE SALTILLO



Año Sabático
1-Feb-2023 a 31-Ene-2024

MODALIDAD A.2
PROYECTO DE INVESTIGACIÓN TECNOLÓGICA

Proyecto:
Probador de Circuitos Digitales
TTL y CMOS de Bajo Costo

Reporte Final

Ing. Ewald Fritsche Ramírez
Docente del Área Eléctrica Electrónica

SALTILLO, COAHUILA

FEBRERO DE 2024

Resultados

1. Resumen del proyecto	4
a) Introducción	4
-Probadores comerciales de circuitos digitales.....	4
-Estructura general.....	6
b) Objetivo del proyecto	7
Objetivo General:	7
Objetivos logrados:	7
c) Metas:	8
d) Desarrollo.....	9
-Calendario	9
-Recopilación de datos técnicos.....	9
Selección de componentes	12
Clasificación de los circuitos	14
Tablas de identificación de parámetros.....	16
-Diseño de placa ZIF	21
Diagrama esquemático	21
Construcción de placa de acoplo.	23
-Algoritmos de pruebas Combinacionales y Secuenciales	28
Algoritmos de Pruebas Combinacionales	31
Algoritmos de Pruebas Secuenciales	33
-Programación de interface gráfica en PC.....	35
-Programación del menú de selección.....	38
-Programación del área de prueba de circuitos.....	41
-Programación de algoritmos de Pruebas combinacionales.....	50
-Programación de Algoritmos de Pruebas Secuenciales.....	52
-Fabricación del circuito impreso PCB-ZIF.....	55
Ensamble del circuito PCB-ZIF.....	55
Montaje del probador de circuitos	57
e) Pruebas y resultados.....	58
-Prueba manual de circuitos	58
-Prueba Automática de circuitos.....	60
-Prueba al Probador de circuitos digitales	62
f) Conclusiones y Observaciones.....	64
-Conclusiones.	64
-Observaciones.....	64
g) Bibliografía	65
2. Objetivo del proyecto	66
-Objetivo General.....	66
-Objetivos específicos:	66
3. Metas: cumplimiento de Metas.....	67
Metas Propuestas:	67

Metas Cumplidas:	67
4. Metodología:.....	70
5. Difusión	71
6. Beneficios y Problemas	72
Beneficios:	72
Problemas:	72
7. Información adicional	73

1. Resumen del proyecto

a) Introducción

El presente proyecto está dirigido a resolver la problemática que se observa cuando los alumnos de los tecnológicos están realizando prácticas o experimentos con circuitos que involucra componentes electrónicos digitales, esta problemática se refleja en la dificultad o la tardanza excesiva en realizar estos experimentos porque es frecuente que los componentes no se encuentran en buen estado, causado por el uso continuo de estos, provocando que se invierta demasiado tiempo en detectar las fallas del experimento dado que se confunden con un error en las conexiones o la propia operación del experimento.

Aunque en los laboratorio de electrónica de los institutos tecnológicos se cuenta con equipo de prueba de circuitos digitales, su uso solo puede ser dentro del área de trabajo en los laboratorios ya que son equipos costosos y en ocasiones escasos, reduciendo la posibilidad de que el alumno avance en los preparativos de su experimento electrónico previo a presentar su resultado en el laboratorio ante el docente revisor.

En los institutos tecnológicos que tienen programas de estudio de carreras como Ingeniería Eléctrica, Electrónica, Ingeniería en Sistemas e Ingeniería Mecánica, manejan materias con temas de electrónica digital; esto hace que el número de alumnos que requieren trabajar en prácticas de laboratorio sea muy grande y si a además se suma la alta demanda de alumnos de la carrera de Mecatrónica, hace muy difícil poder probar previamente todos los componentes electrónicos usados al realizar una práctica o experimento electrónico.

La situación anterior se minimiza o resuelve si se dispone de un probador de circuitos lógicos portátil de uso personal y así tener la certeza del funcionamiento correcto de los componentes electrónicos antes de hacer uso de ellos. Aunque se dispone de equipos probadores comerciales, si estos son del rango económico, están limitados por la cantidad de componentes y el tamaño físico de circuito Integrado a probar, y en el caso de probadores de circuitos de uso profesional, estos presentan un alto costo y tienen que ser resguardados en el laboratorio limitando su acceso.

-Probadores comerciales de circuitos digitales

En el mercado existen diversos equipos probadores de circuitos digitales, algunas versiones económicas son de menos de \$100 Dlls, (ver Imagen 1), están limitados a componentes de 16 a 20 terminales. Su función es la de probar e indicar si el componente se encuentra en buen estado general, no indican qué sección o qué parte de circuito es la que está dañada.

Algunos equipos probadores de circuitos como en la Imagen 2, son equipos portátiles fáciles de usar a un precio aceptable, pero no de los más económicos, estos equipos no indican si es una sección del circuito la que está dañada [1].



Imagen 1. Probador circuitos TTL Económico.



Imagen 2. Probador de circuitos digitales, comercial.

Los probadores de circuitos lógicos de uso industrial tienen un precio elevado del orden de cientos o miles de dólares porque estos son programadores universales de circuitos como el de la Imagen 3, pero de igual manera solo indican si el componente está en buen estado en forma general, sin señalar la parte específica dónde aparezca el daño, no incluye una representación gráfica que muestre la sección específica del daño [2].



Imagen 3. Programador universal y Probador de circuitos digitales.

En temas de electrónica que implica el manejo de circuitos digitales integrados, se enfocan en: Enseñar, Manejar y Aplicar los circuitos digitales, esto también conlleva la necesidad de contar con un sistema de prueba de estos circuitos, pero que tenga características orientadas a un propósito didáctico, sin embargo los probadores del mercado tiene en común lo siguiente:

- No indican que sección está dañada.
- No Interface gráfica didáctica.
- No reproducible por los alumnos, por no contar con la ingeniería de desarrollo.

El presente proyecto busca desarrollar un probador lógico de circuitos digitales con la funcionalidad de un equipo comercial pero además una interfaz gráfica didáctica que le permita al estudiante identificar claramente el funcionamiento del componente y/o las secciones dañadas y tener la certeza de que los circuitos estén en condiciones favorables.

El diseño propuesto se basa en analizar las características de una lista de componentes de acuerdo a las necesidades de la materia de electrónica digital con el fin de determinar la secuencia de pruebas lógicas. Se plantea que este proyecto agilice el desarrollo de los experimentos con circuitos digitales, además de mejorar el conocimiento y manejo de este tipo de circuitos a través de su interface gráfica de operación.

-Estructura general

La Imagen 4, muestra la idea general del probador de circuitos digitales, prácticamente consiste en ensamblar componentes comerciales como el módulo de Arduino mega2560 o compatible, con una placa comercial tipo “Shield” (Placa Acoplada) que permite montar una base ZIF (Zero Insertion Force) para colocar los componentes, el resto es cablear el circuito de soporte para el probador de componentes digitales, la operación es a través de una computadora o una App.



Imagen 4. Probador de circuitos TTL y CMOS propuesto (40Terminales).

Para el desarrollo del probador de circuitos digitales, se buscó que el diseño sea lo más simple posible, dado que una de las ventajas de la construcción de este proyecto es que los alumnos lo construyan por ellos mismos, por lo tanto, su diseño es con pocos componentes y de fácil acceso a estos.

Para el TecNM, el presente proyecto refuerza la enseñanza en el aspecto práctico de temas que implican electrónica digital, así como, contribuye en el auto equipamiento con características orientadas a las necesidades propias de los programas de estudio.

b) Objetivo del proyecto

Objetivo General:

Diseñar y construir un probador lógico de circuitos digitales TTL y CMOS con interface gráfica conectado a una computadora, con el propósito de eliminar el retraso en el desarrollo de experimentos electrónicos con circuitos digitales por causa de componentes dañados, el cual es sumamente útil en la materia de Electrónica Digital de la carrera de Ing. Mecatrónica.

Objetivos logrados:

1. Se logró obtener un equipo operable y de fácil manejo con una interfaz gráfica orientada principalmente a propósito didáctico (Imagen 27. Pantalla de trabajo de prueba de circuitos.).
2. El probador incluye una librería de componentes que cumple con las necesidades de la materia de Electrónica Digital de la carrera de Ing. Mecatrónica y de Diseño Digital de la carrera de Ing. Electrónica. Aunado a lo anterior, se cuenta con una gran ventaja, dado al hecho de tener la ingeniería de desarrollo lo cual permite poder agregar librería rápidamente de acuerdo a necesidades específicas de docentes y alumnos (Tabla 2).
3. Como el proyecto es de bajo costo y de fácil construcción para los alumnos, esto les facilitará desarrollar experimentos o prácticas de laboratorio en temas de electrónica digital con circuitos que han sido aprobados previamente, mejorando así el rendimiento del tiempo requerido en los experimentos.
4. Se desarrolló un proyecto buscando que el costo de construirlo fuese accesible a los alumnos, para que pueda ser fabricado por los alumnos como auto equipamiento.

Costo del probador

- Módulo de Arduino Mega2560 o' compatible	\$ 302.00 MX
- Cable USB-A a USB-B (incluido)	\$ 0.00 MX
- Fabricación de PCB profesional	\$ 11.08 MX c/u en lote de 10Pza
- LEDs y Resistencia	\$ 5.00 MX
- Base tipo ZIF (Zero Insertion Force) 40 Pines	\$ 35.27 MX
Total	\$ 353.35 MX

Nota: los precios son de menudeo, obtenidos de proveedores de material electrónico en páginas de internet.

5. Se logró un equipo con un diseño simple y con componentes comerciales de fácil acceso para que los alumnos y docentes, puedan reproducirlo fácilmente ya que se cuenta con la ingeniería de desarrollo del proyecto.

c) Metas:

Resultados concretos obtenidos en forma cuantitativa respecto a: tesis desarrolladas, publicaciones trabajos de residencia profesional, patentes en trámite, participación en eventos, etc.

Se proyectó al menos enviar para publicación un artículo al Coloquio de investigación multidisciplinaria del tecnológico de Orizaba CIM programado para octubre del 2023, que coincide con el segundo semestre de desarrollo del proyecto, fecha donde ya se cuenta con el avance suficiente para hacer una publicación sobre el proyecto.

Metas cuantificables	Cantidad programada	Cantidad lograda	Observaciones
Incorporación de estudiantes de licenciatura al proyecto (servicio social) :	1	1	Actividad: Desarrollo del software de control,
Artículos científicos publicados en revistas indizadas, enviados	1	1	Coloquio de Investigación Multidisciplinaria, CIM Orizaba (Ver Imagen 5, Evidencia)

Incorporación de estudiantes de licenciatura al proyecto (servicio social):	
Fecha	Nombre de los estudiantes incorporados y su Programa Educativo
22-May-2023	Leonardo Contreras Martínez Ing. En Sistemas Computacionales.

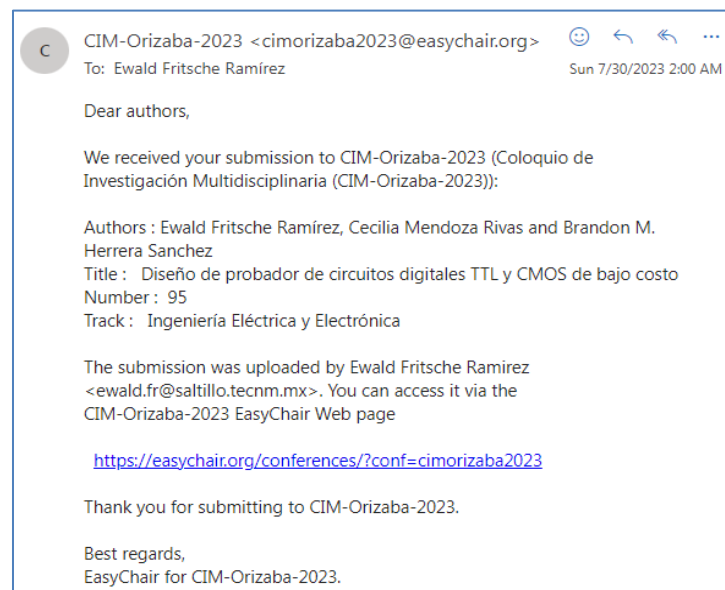


Imagen 5. Evidencia de envío de artículo a publicación a CIM del I.T. de Orizaba

Se proyectó la participación de un estudiante de servicio social del área de sistemas para trabajar con el desarrollo del software de control del probador de circuitos digitales.

Se asignó al alumno Leonardo Contreras Martínez, trabajar con la interface gráfica y estructura del área de trabajo del software.

d) Desarrollo

-Calendario

Cronograma Año Sabatico 2023		2023												2024
	ACTIVIDAD	1-Feb	2-Mar	3-Abr	4-May	5-Jun	6-Jul	7-Ago	8-Sep	9-Oct	10-Nov	11-Dic	12-Ene	
1	Recopilación de datos técnicos y clasificación de los componentes de prueba													
2	Diseño de placa de acoplo ZIF-Microcontrolador													
3	Fabricación de placa de acoplo ZIF-Microcontrolador													
4	Desarrollo de la secuencia de prueba en Circuitos combinacionales													
5	Desarrollo de la secuencia de prueba en Circuitos Secuenciales													
6	Programación interface gráfica en PC													
	ENTREGA DEL REPORTE INTERMEDIO 50%.	1° febrero al 31 de julio de 2023												
7	Programación interface gráfica y menú de selección en PC													
8	Programación de algoritmos de pruebas lógicas en microcontrolador													
9	Programación de algoritmos de pruebas lógicas en PC													
10	Desarrollo del Manual de usuario y construcción													
	ENTREGA DEL REPORTE FINAL 100%.	1° febrero de 2023 al 31 de enero de 2024												

-Recopilación de datos técnicos

Nomenclatura de Circuitos TTL. Para recopilar la información de los circuitos que van a ser incluidos en la lista de componentes del probador de circuitos digitales, se requiere conocer la nomenclatura de lo que son los circuitos digitales TTL y CMOS para tener una identificación precisa de estos.

Los circuitos TTL están divididos en la familia 74 y 54 aunque esto solamente definen características físicas y eléctricas la funcionalidad es la misma y su nomenclatura son muy similares.

Los circuitos más comunes son los circuitos del fabricante Texas Intrument Co. dado que ellos popularizaron el desarrollo de los diferentes circuitos digitales en los años 50, 60 la siguiente descripción se basa en la nomenclatura en los números de circuito de Texas Intrument [3].

Núm. De Componente **SN74LS02ND**

Prefijo del Fabricante "SN"

Familia "74, 54"

Tecnología de Fabricación "L, S, LS, ALS, F" etc.

Número o Función Lógica "00, 02, 06, 12, 32" etc.

Empaquetado (Forma física) "D, N, NDJ, NS" etc.

- Prefijo del Fabricante "SN". El nombre asignado por el fabricante a los circuitos TTL, inician con un prefijo dos o tres letras que definen prácticamente al fabricante.

- Familia "74, 54", Enseguida aparece el número 74 o 54 que es la familia de los circuitos, el 74 corresponde a especificaciones para circuitos de uso comercial es decir en aparatos y equipos como electrodomésticos, el 54, es para circuitos que tienen especificaciones de uso Industrial o militar, no significa que son circuitos de uso exclusivo para la industria o la milicia, sino que sus especificaciones

son más exigentes porque soportan físicamente más esfuerzo como vibraciones y también manejan un rango de voltaje mayor que los circuitos comerciales y esto les permite que sean usados en aplicaciones donde se mantienen funcionando de manera permanente. Podemos encontrar a las dos familias de circuitos disponibles en el mercado sin distinción.

- Tecnología de Fabricación “L, S, LS, ALS, F”. Seguido se identifica la tecnología de fabricación, es decir, la tecnología del proceso interno de construcción, dado que a medida que se fue mejorando la tecnología de fabricación de circuitos integrados se fue mejorando el rendimiento de los circuitos todo encaminado a dos tendencias, reducir el consumo de energía y elevar la velocidad de operación.

Enseguida se listan algunas de las tecnologías de fabricación aplicadas, con la tendencia de más rápido y más eficiente en el consumo de energía.

TTL: serie estándar. Ejemp. 7400, 7402, 7408.

TTL-L (low power): serie de bajo consumo.

TTL-S (schottky): serie rápida (usa diodos Schottky).

TTL-AS (advanced schottky): versión mejorada de la serie anterior.

TTL-LS (low power schottky): combinación de las tecnologías L y S (es la familia más extendida).

TTL-ALS (advanced low power schottky): versión mejorada de la serie LSS.

TTL-F (FAST : fairchild advanced schottky).

TTL-AF (advanced FAST): versión mejorada de la serie F.

TTL-HCT (high speed C-MOS): Serie HC dotada de niveles lógicos compatibles con TTL.

TTL-G (GHz C-MOS): GHz.

- Número o Función Lógica “00, 02, 06,12, 32”, Es precisamente el número del circuito o lo que es la función lógica que define la operación digital, es este número lo que hace diferente a cada uno de los circuitos de la familia TTL, por ejemplo el SN74LS00ND son cuatro compuertas de dos entradas tipo NAND, el SN74LS02ND son cuatro compuertas de dos entradas tipo NOR.

Los circuitos TTL de la serie estándar, son los primeros circuitos fabricados y en su nomenclatura simplemente se señalaban como circuitos 7400,7402, 7404 etc. no se especificaba la tecnología de fabricación.

- Empaquetado (Forma física) “D, N, NDJ, NS”, Finalmente una identificación de la forma física (Ver Imagen 6) y que también define la distribución de las terminales, por ejemplo tipo DIP o SOIC [3].

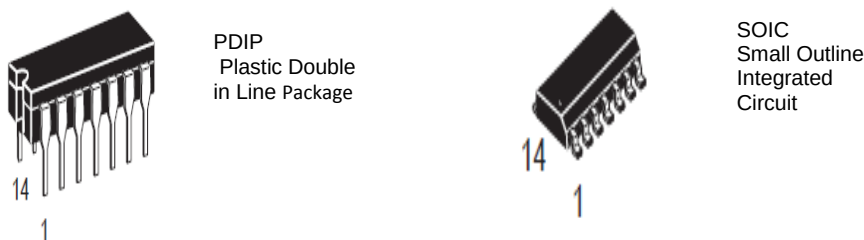


Imagen 6. Empaquetado de circuitos TTL [3].

**SN5401, SN54LS01,
SN7401, SN74LS01**
APRIL 1985 — REVISED MARCH 1988

QUADRUPLE 2-INPUT POSITIVE-NAND GATES WITH OPEN-COLLECTOR OUTPUTS

- Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

description

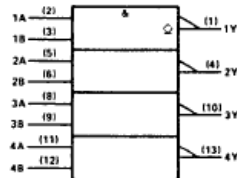
These devices contain four independent 2-input NAND gates. The open-collector outputs require pull-up resistors to perform correctly. They may be connected to other open-collector outputs to implement active-low wired-OR or active-high wired-AND functions. Open-collector devices are often used to generate higher V_{OH} levels.

The SN5401 and SN54LS01 are characterized for operation over the full military temperature range of -55°C to 125°C . The SN7401 and SN74LS01 are characterized for operation from 0°C to 70°C .

FUNCTION TABLE (each gate)

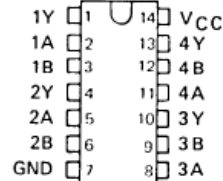
INPUTS		OUTPUT
A	B	Y
H	H	L
L	X	H
X	L	H

logic symbol†

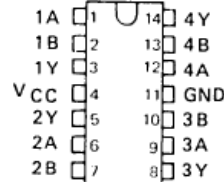


†This symbol is in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.
Pin numbers shown are for D, J, N, and W packages.

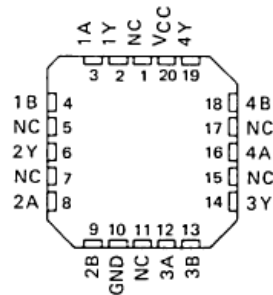
SN5401 . . . J PACKAGE
SN54LS01 . . . J OR W PACKAGE
SN7401 . . . N PACKAGE
SN74LS01 . . . D OR N PACKAGE
(TOP VIEW)



SN5401 . . . W PACKAGE
(TOP VIEW)



SN54LS01 . . . FK PACKAGE
(TOP VIEW)



NC - No internal connection

2
TTL Devices

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS
INSTRUMENTS**
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

2-9

Imagen 7. Muestra de hoja de datos del componente TTL 7401 [3].

La información y descripción de cada componente se encuentra en la hoja de datos o en inglés “dataSheet” del fabricante, ahí se encuentra la información técnica del funcionamiento, las características eléctricas, el diagrama de conexión en la forma física y diagramas de aplicación.

En la Imagen 7 se presenta una muestra de la hoja de datos del circuito 7401 de Texas Instrument.

Nomenclatura de circuitos CMOS.

En el caso de los circuitos CMOS, estos se identifican como familia 4000 con una estructura en el nombre semejante al caso de los circuitos TTL, donde cada parte del nombre representa una característica del circuito. La siguiente nomenclatura está basada en los componentes fabricados por Motorola donde su prefijo es MC1 y la familia es la 4000.

Núm. De Componente **MC14000BCP**

Prefijo del Fabricante “MC1”

Familia “4000” o “40”

Número o Función Lógica “00, 02, 06, 07” etc.

Serie (Tecnología de Fabricación) “B, UB”

Empaquetado (Forma física) “CP, CL, D”

El nombre de un circuito CMOS está conformado por el prefijo, que define al fabricante, enseguida viene el número que identifica la familia que es la familia 4000 o también llamada familia 40, aparece un número 40 después del prefijo, enseguida del 40 aparece el número de la función lógica, como es el caso del 00, 02, 40, 06, etc. este número es el que define la función lógica de nuestro circuito, es prácticamente el número del circuito, enseguida se identifica lo que es la serie, que viene a ser el equivalente a la tecnología de fabricación, y finalmente se tiene la identificación del empaquetado, es decir, la forma física.

La Imagen 8 Muestra de hoja de datos del componente CMOS 4006 [4], se presenta la primera página de la hoja de datos del circuito MC14006B de Motorola.

Selección de componentes

La lista de componentes que se incluyen para el desarrollo de este proyecto, “Probador de circuitos digitales TTL y CMOS”, está basada en dos criterios: Uno, la lista de componentes utilizados en temas de electrónica digital disponible en el laboratorio de electrónica digital del tecnológico de Saltillo, instituto donde se realiza el presente proyecto. Dos, lista de componentes utilizados, según las necesidades en particular de los profesores que imparten la materia electrónica digital de la carrera de Ingeniería Mecatrónica y de la materia de diseño digital de la carrera de Ingeniería Electrónica.

MC14006B

18-Bit Static Shift Register

The MC14006B shift register is comprised of four separate shift register sections sharing a common clock: two sections have four stages, and two sections have five stages with an output tap on both the fourth and fifth stages. This makes it possible to obtain a shift register of 4, 5, 8, 9, 10, 12, 13, 14, 16, 17, or 18 bits by appropriate selection of inputs and outputs. This part is particularly useful in serial shift registers and time delay circuits.

- Output Transitions Occur on the Falling Edge of the Clock Pulse
- Fully Static Operation
- Can be Cascaded to Provide Longer Shift Register Lengths
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-power TTL Loads or One Low-power Schottky TTL Load Over the Rated Temperature Range
- Pin-for-Pin Replacement for CD4006B

MAXIMUM RATINGS* (Voltages Referenced to V_{SS})

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage	- 0.5 to + 18.0	V
V_{in}, V_{out}	Input or Output Voltage (DC or Transient)	- 0.5 to $V_{DD} + 0.5$	V
I_{in}, I_{out}	Input or Output Current (DC or Transient), per Pin	± 10	mA
P_D	Power Dissipation, per Package†	500	mW
T_{stg}	Storage Temperature	- 65 to + 150	°C
T_L	Lead Temperature (8-Second Soldering)	260	°C

* Maximum Ratings are those values beyond which damage to the device may occur.

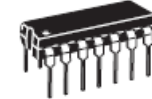
† Temperature Derating:

Plastic "P and D/DW" Packages: - 7.0 mW/°C From 65°C To 125°C

Ceramic "L" Packages: - 12 mW/°C From 100°C To 125°C



L SUFFIX
CERAMIC
CASE 632



P SUFFIX
PLASTIC
CASE 646



D SUFFIX
SOIC
CASE 751A

ORDERING INFORMATION

MC14XXXBCP Plastic
MC14XXXBCL Ceramic
MC14XXXBD SOIC

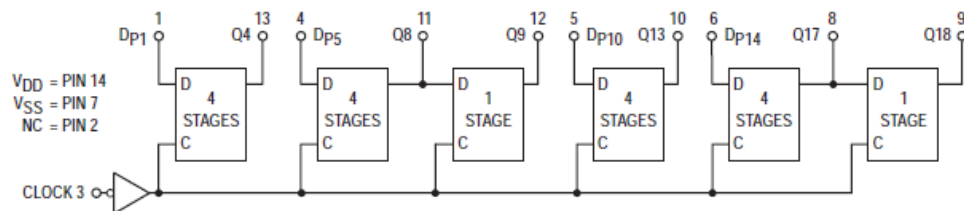
$T_A = -55^\circ$ to 125°C for all packages.

TRUTH TABLE (Single Stage)

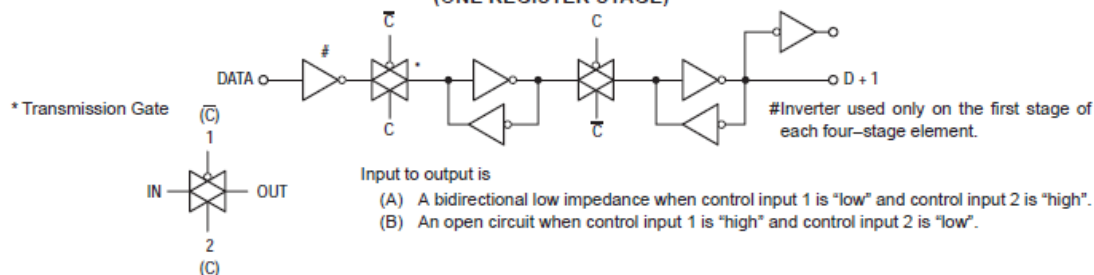
D_n	C	Q_{n+1}
0	~	0
1	~	1
x	~	Q_n

X = Don't Care

BLOCK DIAGRAM



LOGIC DIAGRAM (ONE REGISTER STAGE)



En la Tabla 2, se muestra la lista de los números de componentes TTL y CMOS que se van a incluir en la etapa del desarrollo de este prototipo, se tienen 113 circuitos TTL y 37 circuitos CMOS, que suman un total de 150 componentes diferentes, para ser programados en el software como etapa inicial, lo cual es la cantidad especificada en el objetivo y metas del proyecto de acuerdo a la planeación y tiempo de desarrollo.

Clasificación de los circuitos

La clasificación de los circuitos es de suma importancia porque define las operaciones lógicas que se tienen que hacer para probar los diferentes circuitos, el listado de la Tabla 1 muestra la clasificación de los tipos de circuitos según su operación lógica, es decir, los tipos de circuitos diferentes según la función lógica que realizan, cada uno de los circuitos de la Tabla 2, se ubica en algún tipo de operación lógica de la Tabla 1.

La información del circuito digital respecto a la función lógica la encontramos también en la hoja de datos, el nombre del componente normalmente es suficiente para identificar qué tipo de circuito, si esto no es suficiente, el nombre del circuito es acompañado con una descripción de la operación lógica del circuito como se muestra en la Imagen 7.

Es de suma importancia identificar que las categorías es una cantidad delimitada, si consideramos que cada categoría implica una secuencia de pruebas, por lo tanto, se pudo tomar como un listado no muy grande y esto permite programar en el software una cantidad de secuencias de control para poder llevar a cabo las pruebas en los circuitos.

Tabla 1. Clasificación de circuitos digitales.

	Funciones logicas
1	Compuerta Digital And,Or,Not, Etc.
2	Compuerta Digital con Habilitador. And,Or,Not, Etc.
3	Codificador
4	Decodificador Demultiplexor
5	Selector multiplexor
6	Decodificador de Prioridad
7	Compuertas en cascada
8	Divisor Binario
9	Contador Binario
10	Flip-Flop J-K
11	Flip-Flop R-S
12	Flip-Flop D
13	Latch
14	Sumador Binario
15	Comparador Digital
16	Registro de Corrimiento

Tabla 2. Lista de compuertas TTL y CMOS a manejar por el probador de circuitos.

Parte	Compuertas TTL - Descripción	Parte	Compuertas TTL - Descripción
1 7400	QUAD 2-INPUT NAND GATE	58 7492	DIVIDE-BY-12 COUNTER (SEPARATE DIVIDE-BY-2 AND DIVIDE-BY-6 SECTIONS)
2 7401	QUADRUPLE 2-INPUT POSITIVE-NAND GATES	59 7493	4-BIT BINARY COUNTER (SEPARATE DIVIDE-BY-2 AND DIVIDE-BY-8 SECTIONS)
3 7402	QUAD 2-INPUT NOR GATE	60 7494	4-81T SHIFT REGISTERS
4 7403	QUADRUPLE 2-INPUT POSITIVE-NAND GATES	61 7496	5-81T SHIFT REGISTERS
5 7404	HEX INVERTER GATE	62 74107	DUAL J-K FLIP-FLOP, CLEAR
6 7405	HEX INVERTER GATE	63 74109	DUAL J-K POSITIVE-EDGE-TRIGGERED FLLP-FLOPS WITH PRESET AND CLEAR
7 7406	HEX INVERTER BUFFERS	64 74111	DUAL J-K MASTER-SLAVE FLLP-FLOPS WITH DATA LOCKOUT
8 7407	HEX BUFFERS	65 74112	DUAL J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOP, CLEAR AND PRESET
9 7408	QUAD 2-INPUT AND GATE	66 74116	DUAL 4-81T LATCHES WITH CLEAR
10 7409	QUAD 2-INPUT AND GATE	67 74125	QUADRUPLE BUS BUFFERS WITH 3-STATE OUTPUTS
11 7410	TRIPLE 3-INPUT NAND GATE	68 74126	QUAD BUS BUFFER, POSITIVE ENABLE
12 7411	TRIPLE 3-INPUT POSITIVE-AND GATES	69 74134	12-INPUT POSITIVE-NAND GATES WITH 3-STATE OUTPUTS
13 7412	TRIPLE 3-INPUT NAND GATE	70 74135	QUADRUPLE EXCLUSIVE-OR/NOR GATES
14 7413	DUAL 4-INPUT POSITIVE-NAND	71 74136	QUAD 2-INPUT XOR GATE
15 7414	HEX SCHMITT-TRIGGER INVERTERS	72 74137	3-LINE TO 8-LINE DECODERS/DEMULPLEXERS WITH ADDRESS LATCHES
16 7415	TRIPLE 3-INPUT POSITIVE-AND GATES	73 74138	3-TO-8 LINE DECODER/DEMULPLEXER, INVERTING OUTPUTS
17 7416	HEX INVERTER BUFFERS/DRIVERS	74 74139	DUAL 2-LINE TO 4-LINE DECODERS/DEMULPLEXERS
18 7417	HEX BUFFER DRIVER GATE	75 74145	BCD TO DECIMAL DECODER/DRIVER
19 7420	DUAL 4-INPUT NAND GATE	76 74147	10-LINE TO 4-LINE PRIORITY ENCODER
20 7421	DUAL 4-INPUT POSITIVE-AND GATES	77 74148	8-LINE TO 3-LINE PRIORITY ENCODER
21 7425	DUAL 4-INPUT NOR GATES WITH STROBE	78 74150	16-LINE TO 1-LINE DATA SELECTOR/MULTIPLEXER
22 7426	QUADRUPLE 2-INPUT HIGH-VOLTAGE INTERFACE POSITIVE-NAND GATES	79 74151	8-LINE TO 1-LINE DATA SELECTOR/MULTIPLEXER
23 7427	TRIPLE 3-INPUT NOR GATE	80 74153	DUAL 4-LINE TO 1-LINE DATA SELECTOR/MULTIPLEXER, NON-INVERTING OUTPUTS
24 7428	QUADRUPLE 2-INPUT POSITIVE-NOR BUFFERS	81 74154	4-TO-16 LINE DECODER/DEMULPLEXER, INVERTING OUTPUTS
25 7430	8-INPUT POSITIVE-NAND GATES	82 74155	DUAL 2-TO-4 LINE DECODER/DEMULPLEXER, INVERTING OUTPUTS
26 7432	QUAD 2-INPUT OR GATE	83 74156	DUAL 2-LINE TO 4-LINE DEMULPLEXERS
27 7433	QUADRUPLE 2-INPUT POSITIVE-NOR BUFFERS	84 74157	QUAD 2-LINE TO 1-LINE DATA SELECTOR/MULTIPLEXER, NON-INVERTING OUTPUTS
28 7437	QUADRUPLE 2-INPUT POSITIVE-NAND BUFFERS	85 74158	QUADRUPLE 2-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS
29 7439	QUADRUPLE 2-INPUT POSITIVE-NAND BUFFERS	86 74159	4-LINE TO 16-LINE DECODERS/DEMULPLEXERS
30 7440	DUAL 4-INPUT POSITIVE-NAND BUFFERS	87 74161	SYNCHRONOUS PRESETTABLE 4-BIT BINARY COUNTER, ASYNCHRONOUS CLEAR
31 7442	BCD TO DECIMAL DECODER	88 74163	SYNCHRONOUS PRESETTABLE 4-BIT BINARY COUNTER, SYNCHRONOUS CLEAR
32 7445	BCD TO DECIMAL DECODER/DRIVER	89 74164	8-BIT SERIAL-IN PARALLEL-OUT (SIPO) SHIFT REGISTER, ASYNCHRONOUS CLEAR
33 7446	BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS	90 74165	8-BIT PARALLEL-IN SERIAL-OUT (PISO) SHIFT REGISTER, PARALLEL LOAD
34 7447	BCD TO 7-SEGMENT DECODER/DRIVER	91 74171	QUADRUPLE D-TYPE FLLP-FLOPS WITH CLEAR
35 7448	BCD TO 7-SEGMENT DECODER/DRIVER	92 74174	HEX D FLIP-FLOP, COMMON ASYNCHRONOUS CLEAR
36 7449	BCD TO 7-SEGMENT DECODER/DRIVER	93 74175	QUADRUPLE D-TYPE FLLP-FLOPS WITH CLEAR
37 7450	DUAL 2-WIDE 2-INPUT AND-OR-INVERT GATES (ONE GATE EXPANDABLE)	94 74176	PRESETTABLE DECADE (BI-QUINARY) COUNTER/LATCH
38 7451	DUAL 2-2-INPUT AND-OR-INVERT (AOI) GATE	95 74181	4-BIT ARITHMETIC LOGIC UNIT AND FUNCTION GENERATOR
39 7453	EXPANDABLE 4-WIDE AND-OR-INVERT GATES	96 74183	DUAL CARRY-SAVE FULL ADDERS
40 7454	2-2-2-2-INPUT AND-OR-INVERT GATE	97 74190	SYNCHRONOUS PRESETTABLE UP/DOWN 4-BIT DECADE COUNTER
41 7455	2-WIDE 4-INPUT AND-OR-INVERT GATES	98 74191	SYNCHRONOUS UP/DOWN COUNTERS WITH DOWN/UP MODE CONTROL
42 7465	4-2-3-2 INPUT AND-OR-INVERT GATES	99 74192	SYNCHRONOUS PRESETTABLE UP/DOWN 4-BIT DECADE COUNTER, CLEAR
43 7468	DUAL 4-BIT DECADE COUNTERS	100 74193	SYNCHRONOUS PRESETTABLE UP/DOWN 4-BIT BINARY COUNTER, CLEAR
44 7469	DUAL 4-BIT BINARY COUNTERS	101 74194	4-BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTER
45 7470	AND-GATED J-K POSITIVE-EDGE-TRIGGERED FLLP-FLOPS WITH PRESET AND CLEAR	102 74195	4-BIT PARALLEL-ACCESS SHIFT REGISTERS
46 7472	AND-GATED J-K MASTER-SLAVE FLLP-FLOPS WITH PRESET AND CLEAR	103 74196	50/30/100-MHZ PRESETTABLE DECADE OR BINARY COUNTERS/LATCHES
47 7473	DUAL J-K FLIP-FLOP, ASYNCHRONOUS CLEAR	104 74198	8-BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTER
48 7474	DUAL D POSITIVE EDGE TRIGGERED FLIP-FLOP, ASYNCHRONOUS PRESET AND CLEAR	105 74199	8-BIT SHIFT REGISTERS
49 7475	4-BIT BISTABLE LATCH, COMPLEMENTARY OUTPUTS	106 74240	OCTAL BUFFERS AND LINE DRIVERS WITH 3-STATE OUTPUTS
50 7476	DUAL J-K FLIP-FLOP, ASYNCHRONOUS PRESET AND CLEAR	107 74241	OCTAL BUFFER, NON-INVERTING OUTPUTS
51 7477	4-BIT BISTABLE LATCHES	108 74242	QUADRUPLE BUS TRANSCEIVERS
52 7478	DUAL J-K FLLP-FLOPS WITH PRESET, COMMON CLOCK, AND COMMON CLEAR	109 74243	QUADRUPLE BUS TRANSCEIVERS
53 7483	4-BIT BINARY FULL ADDER	110 74244	OCTAL BUFFER, NON-INVERTING OUTPUTS
54 7485	4-BIT MAGNITUDE COMPARATORS	111 74245	OCTAL BUS TRANSCEIVER, NON-INVERTING OUTPUTS
55 7486	QUAD 2-INPUT XOR GATE	112 74266	QUAD 2-INPUT XNOR GATE
56 7490	DECADE COUNTER (SEPARATE DIVIDE-BY-2 AND DIVIDE-BY-5 SECTIONS)	113 74283	4-BIT BINARY FULL ADDER (HAS CARRY IN FUNCTION)
57 7491	8-BIT SHIFT REGISTERS		

Parte	Compuertas CMOS - Descripción	Parte	Compuertas CMOS - Descripción
1 4000	DUAL 3-INPUT "NOR" GATE PLUS INVERTER	20 4040	12-BIT BINARY COUNTER
2 4001	QUAD 2-INPUT NOR GATE	21 4042	QUAD TRANSPARENT LATCH
3 4002	DUAL 4-INPUT NOR GATA	22 4043	QUAD "NOR" R-S LATCH
4 4006	DUAL COMPLEMENTARY PAIR PLUS INVERTER	23 4044	QUAD "NAND" R-S LATCH
5 4008	4-BIT FULL ADDER	24 4049	HEX BUFFERS INVERTING
6 4011	QUAD 2-INPUT NAND GATE	25 4050	HEX BUFFERS NONINVERTING
7 4012	DUAL 4-INPUT NAND GATE	26 4068	8-INPUT NAND GATE
8 4013	DUAL TYPE D FLIP-FLOP	27 4069	HEX INVERTER
9 4014	8-BIT STATIC SHIFT REGISTER	28 4070	QUAD EXCLUSIVE "OR" GATE
10 4015	DUAL 4-BIT STATIC SHIFT REGISTER	29 4071	QUAD 2-INPUT OR GATE
11 4017	DECADE COUNTER	30 4072	DUAL 4-INPUT OR GATE
12 4020	14-BIT BINARY COUNTER	31 4073	TRIPLE 3-INPUT AND GATE
13 4021	8-BIT STATIC SHIFT REGISTER	32 4075	TRIPLE 3-INPUT OR GATE
14 4023	TRIPLE 3-INPUT NAND GATE	33 4077	QUAD EXCLUSIVE "NOR" GATE
15 4025	TRIPLE 3-INPUT NOR GATE	34 4078	8-INPUT NOR GATE
16 4027	DUAL J-K FLLP-FLOP	35 4081	QUAD 2-INPUT AND GATE
17 4028	BCD-TO-DECIMAL DECODER BINARY-TO-OCTAL DECODER	36 4082	DUAL 4-INPUT AND GATE
18 4029	BINARY/DECADE UP/DOWN COUNTER	37 4093	QUAD 2-INPUT "NAND" SCHMITT TRIGGER
19 4035	4-BIT PARALLEL-IN/PARALLEL-OUT SHIFT REGISTER		

DM7400

Quad 2-Input NAND Gates

General Description

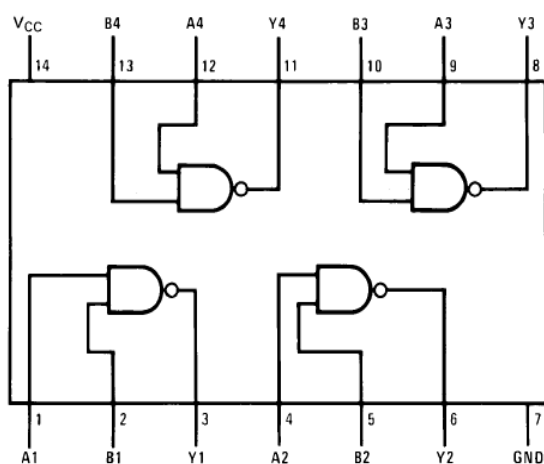
This device contains four independent gates each of which performs the logic NAND function.

Ordering Code:

Order Number	Package Number	Package Description
DM7400M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM7400N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = \overline{AB}$$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

Imagen 9. Descripción de la función lógica en la hoja de datos [5].

Tablas de identificación de parámetros

La información recopilada de las hojas de datos de cada componente, como la mostrada en la Imagen 9, es vaciada en tablas que representan la base de datos con la información que requiere el software para seleccionar el número de componente, las pruebas que tiene que hacer y la información que va a presentar como resultado de la operación del circuito a probar.

En la Tabla 3 se muestra un listado de las tablas-Base de datos que organizan la información recopilada para el software de control del probador de circuitos.

Tabla 3. Listado de tablas-Base de datos del Probador.

	Tabla	Descripción
1	Nombres	Contiene la lista de nombres de circuitos TTL y CMOS a probar
2	Terminales	Información del nombre de terminales y el tipo, Entrada y/o salida
3	Descripción	Identificación descriptiva del fabricante, según la función lógica
4	PinOut	Describe la posición de terminales en la figura conocida como PinOut
5	Pruebas	Pruebas y parámetros de acuerdo a la lógica del componente a probar

Las tablas que se describen a continuación representan una sección de cada una de las propias tablas en donde se tiene la información que el software de control del circuito probador toma como referencia y entrelaza para poder realizar todo el proceso de pruebas.

Tabla de nombres de número de parte

La primera tabla consultada por el software para iniciar el proceso de prueba de un componente es la Tabla 4, la cual contiene la lista de los números de parte que el probador de circuito es capaz de procesar, en esa misma tabla se observa información adicional tal como: la función lógica que realiza, el fabricante del componente, el número de parte. El proceso de prueba inicia con la selección del componente de la columna marcada como número de parte. Para el presente proyecto están contemplados 150 componentes diferentes aunque algunos fabricantes manejan número de parte con variantes en su nomenclatura debido a las características eléctricas o al empaquetado físico. Además, fabricantes diferentes, producen componentes equivalentes a otros fabricantes con diferente número de parte pero su función lógica es la misma, por esa razón en la Tabla 4 aparece la columna del fabricante y su número de componente, aunque finalmente como función lógica y como proceso de prueba es el mismo.

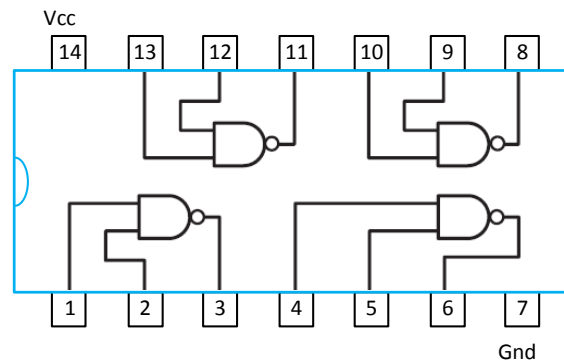


Imagen 10. Diagrama de terminales del circuito 7400 conocido como PinOut.

La columna IC-Info de la Tabla 4 enlaza la Tabla 5 que contiene la información de los parámetros de los circuitos, en la Tabla 5 se encuentra la Distribución Terminales PinOut” como en la Imagen 10, además, información de la posición en Base ZIF, que determina a través de una figura cómo se debe colocar el componente como se muestra en la Imagen 11.

Tabla 4. Tabla Parcial de Identificación del número de componente a probar.

	Tipo	Funcion	Fabricante	Numero de Parte	IC Info	Enpaquetado
1	TTL-54	NAND	TI Standard	SN54S00	7400-1-TI	PDIP
2	TTL-54	NAND	Texas Instrument	SN5400J	7400-1-TI	CDIP
3	TTL-54	NAND	Texas Instrument	SN54S00J	7400-1-TI	CDIP
4	TTL-54	NAND	Texas Instrument	SN54LS00J	7400-1-TI	CDIP
1	TTL-74	NAND	TI Standard	7400	7400-1-TI	PDIP
5	TTL-74	NAND	Texas Instrument	SN74S00N	7400-1-TI	PDIP
6	TTL-74	NAND	Texas Instrument	SN74LS00N	7400-1-TI	PDIP
7	TTL-74	NAND	Texas Instrument	SN74LS00NSR	7400-1-TI	PDIP
8	TTL-74	NAND	Motorola	MC74F00J	7400-MO-DIP	CDIP
9	TTL-74	NAND	Motorola	MC74F00N	7400-MO-DIP	PDIP
10	TTL-74	NAND	Motorola	MC54F00J	7400-MO-DIP	CDIP
11	TTL-74	NAND	Motorola	MC54F00N	7400-MO-DIP	PDIP
12	TTL-74	NAND	Fairchild	DM7400N	7400-FA-DIP	PDIP
13	TTL-74	NAND	TI Standard	7401	7401-1-TI	PDIP
14	TTL-74	NOR	TI Standard	7402	7402-1-TI	PDIP
15	TTL-74	NAND	TI Standard	7403	7403-1-TI	PDIP
16	TTL-74	Inverter	TI Standard	7404	7404-1-TI	PDIP
17	TTL-74	Inverter	TI Standard	7405	7405-1-TI	PDIP
18	TTL-74	Inverter	TI Standard	7406	7406-1-TI	PDIP
19	TTL-74	Buffers	TI Standard	7407	7407-1-TI	PDIP
20	TTL-74	AND	TI Standard	7408	7408-1-TI	PDIP
21	TTL-74	AND	TI Standard	7409	7409-1-TI	PDIP
22	TTL-74	NAND	TI Standard	7410	7410-1-TI	PDIP
23	TTL-74	AND	TI Standard	7411	7411-1-TI	PDIP
24	TTL-74	NAND	TI Standard	7412	7412-1-TI	PDIP

En la parte derecha de la Tabla 5 se tiene columna la prueba que se realiza y la hoja de datos del componente. La aparente redundancia de la información en la tabla, es porque está diseñada para manejar información de diferentes fabricantes, y en esta muestra de la tabla contiene información de componentes únicamente de Texas Instrument.

Tabla 5. Tabla IC-Info Parámetros del circuito seleccionado.

IC Info	Patas	Terminales Base ZIF	Distribución Física Terminales PinOut	Nombre Visuales de Terminales	Nombre Terminales PinOut	Descripcion	Prueba	Datasheet
7401-1-TI	14	ZIF DIP-14-300-100	7401-1-TI-DL14.PNG	7401-1-TI	7401-1-TI	7401-0	7401-1-TI	SN74xx01
7402-1-TI	14	ZIF DIP-14-300-100	7402-1-TI-DL14.PNG	7402-1-TI	7402-1-TI	7402-0	7402-1-TI	SN74xx02
7403-1-TI	14	ZIF DIP-14-300-100	7403-1-TI-DL14.PNG	7403-1-TI	7403-1-TI	7403-0	7403-1-TI	SN74xx03
7404-1-TI	14	ZIF DIP-14-300-100	7404-1-TI-DL14.PNG	7404-1-TI	7404-1-TI	7404-0	7404-1-TI	SN74xx04
7405-1-TI	14	ZIF DIP-14-300-100	7405-1-TI-DL14.PNG	7405-1-TI	7405-1-TI	7405-0	7405-1-TI	SN74xx05
7406-1-TI	14	ZIF DIP-14-300-100	7406-1-TI-DL14.PNG	7406-1-TI	7406-1-TI	7406-0	7406-1-TI	SN74xx06
7407-1-TI	14	ZIF DIP-14-300-100	7407-1-TI-DL14.PNG	7407-1-TI	7407-1-TI	7407-0	7407-1-TI	SN74xx07
7408-1-TI	14	ZIF DIP-14-300-100	7408-1-TI-DL14.PNG	7408-1-TI	7408-1-TI	7408-0	7408-1-TI	SN74xx08

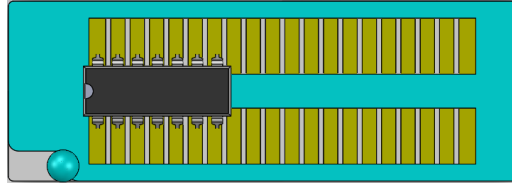


Imagen 11. Ubicación de componente en base ZIF.

Tabla de descripción de componente

La Tabla 6, es la descripción de función lógica del componente dado por el fabricante, se encuentra en la hoja de datos, esa descripción identifica inmediatamente la función lógica y el tipo de circuito. El software de control presenta la descripción del circuito que se está probando, de esta manera si llega a haber alguna confusión o se tenga alguna duda del circuito que se ha colocado para prueba, se pueda verificar, claro está que las pruebas se van a realizar en base al componente seleccionado pero se puede identificar qué tipo de circuito es, a través de la descripción.

Tabla 6. Descripción de función lógica del componente, dado por el fabricante.

	Num Parte	Descripción de circuitos
1	7400	Quad 2-Input Nand Gate
2	7401	Quad 2-Input NAND Gate Open Collector
3	7402	Quad 2-Input NOR Gate
4	7403	Quadruple 2-Input Positive-Nand Gates
5	7404	Hex Inverter Gate
6	7405	Hex Inverters With Open-Collector Outputs
7	7406	Hex Inverter Buffers With Open-Collector High-Voltage Outputs
8	7407	Hex Inverter Buffers With Open-Collector High-Voltage Outputs
9	7408	Quad 2-Input AND Gate
10	7409	Quad 2-Input AND Gate
11	7410	Triple 3-Input NAND Gate
12	7411	Triple 3-Input Positive-And Gates

Tabla de terminales

La Tabla 7, es una tabla de especial interés porque determina las terminales del circuito que se va a probar cuando ya está montado en la base tipo ZIF, terminales numeradas del uno a la 20 por un lado y de la 21 a la 40 por el siguiente lado del circuito. Para apreciar mejor la Tabla 7, es necesario que sea una imagen a colores porque los colores identifican el tipo de terminal y de esa manera se facilita la identificación de la información, en la parte superior de la tabla está la codificación del significado de cada color para poder leer más fácilmente el contenido de la tabla, esa tabla es de especial interés también, porque aparte de definir las terminales, visualmente nos permite identificar dónde van colocados los voltajes de alimentación tanto, la tierra en color negro y el voltaje alimentación de 5 voltios en color rojo, además de identifica cada una de las terminales del componente a probar, entradas de color azul y salidas de color verde. Visualmente ayuda a identificar los componentes y a reducir errores en la generación de la misma tabla, dado que los colores facilitan identificar el tipo y la ubicación de las terminales.

Tabla 7. Terminales del Circuito a probar en Base ZIF.

			S1	Salida 2 estados				S2	Salida Colector Abierto				S3	Salida 3 Estados				S4	Salida Bidireccional 3 Estados							
			S1	Entrada				E2	NC No conectado				E3	Vcc				E4	Gnd							
				Negado									1	2 3 4				5	6 7 8 9 10 11 12							
114	Parte	Enpaque- tado	Patas	1	2	3	4	5	6	7	8	9	10		40	39	38	37	36	35	34	33	32	31	30	29
1	7400-0		14	1A	1B	1Y	2A	2B	2Y	Gnd					Vcc	4B	4A	4Y	3B	3A	3Y					
2	7400-1	CDFP	14	1A	1B	1Y	Vcc	2Y	2A	2B					4Y	4B	4A	Gnd	3B	3A	3Y					
3	7400-2	PLCC	20	NC	1A	1B	1Y	NC	2A	NC	2B	2Y	Gnd		Vcc	4B	4A	NC	4Y	NC	3B	3A	3Y	NC		
4	7401-0	DIP300	14	1Y	1A	1B	2Y	2A	2B	Gnd					Vcc	4Y	4B	4A	3Y	3B	3A					
5	7401-1	CDFP	14	1A	1B	1Y	Vcc	2Y	2A	2B					4Y	4B	4A	Gnd	3B	3A	3Y					
6	7401-2	PLCC	20	NC	1Y	1A	1B	NC	2Y	NC	2A	2B	Gnd		Vcc	4Y	4B	NC	4A	NC	3Y	3B	3A	NC		
7	7402-0	DIP300	14	1Y	1A	1B	2Y	2A	2B	Gnd					Vcc	4Y	4B	4A	3Y	3B	3A					
8	7403-0	DIP300	14	1A	1B	1Y	2A	2B	2Y	Gnd					Vcc	4B	4A	4Y	3B	3A	3Y					
9	7404-0	DIP300	14	1A	1Y	2A	2Y	3A	3Y	Gnd					Vcc	6A	6Y	5A	Y5	4A	4Y					
10	7405-0	DIP300	14	1A	1Y	2A	2Y	3A	3Y	Gnd					Vcc	6A	6Y	5A	Y5	4A	4Y					
11	7406-0	DIP300	14	1A	1Y	2A	2Y	3A	3Y	Gnd					Vcc	6A	6Y	5A	Y5	4A	4Y					
12	7407-0	DIP300	14	1A	1Y	2A	2Y	3A	3Y	Gnd					Vcc	6A	6Y	5A	Y5	4A	4Y					
13	7408-0	DIP300	14	1A	1B	1Y	2A	2B	2Y	Gnd					Vcc	4B	4A	4Y	3B	3A	3Y					
14	7409-0	DIP300	14	1A	1B	1Y	2A	2B	2Y	Gnd					Vcc	4B	4A	4Y	3B	3A	3Y					
15	7410-0	DIP300	14	1A	1B	2A	2B	2C	2Y	Gnd					Vcc	1C	1Y	3C	3B	3A	3Y					
16	7411-0	DIP300	14	1A	1B	2A	2B	2C	2Y	Gnd					Vcc	1C	1Y	3C	3B	3A	3Y					
17	7412-0	DIP300	14	1A	1B	2A	2B	2C	2Y	Gnd					Vcc	1C	1Y	3C	3B	3A	3Y					
18	7413-0	DIP300	14	1A	1B	NC	1C	1D	1Y	Gnd					Vcc	2D	2C	NC	2B	2A	2Y					
19	7414-0	DIP300	14	1A	1Y	2A	2Y	3A	3Y	Gnd					Vcc	6A	6Y	5A	Y5	4A	4Y					
20	7415-0	DIP300	14	1A	1B	2A	2B	2C	2Y	Gnd					Vcc	1C	1Y	3C	3B	3A	3Y					
21	7416-0	DIP300	14	1A	1Y	2A	2Y	3A	3Y	Gnd					Vcc	6A	6Y	5A	Y5	4A	4Y					
22	7417-0	DIP300	14	1A	1Y	2A	2Y	3A	3Y	Gnd					Vcc	6A	6Y	5A	Y5	4A	4Y					
23	7419-0	DIP300	14	1A	1Y	2A	2Y	3A	3Y	Gnd					Vcc	6A	6Y	5A	Y5	4A	4Y					
24	7420-0	DIP300	14	1A	1B	NC	1C	1D	1Y	Gnd					Vcc	2D	2C	NC	2B	2A	2Y					

Tabla de distribución de terminales

La Tabla 8 determina la posición de los nombres de acuerdo a la distribución de las terminales en el componente físico, si es un integrado tipo de doble línea o bien si es un integrado cuadrado, esto es para la representación gráfica de la interface didáctica en el software, en donde se muestra el componente, el estado de las terminales, los nombres y en algunos circuitos hasta el diagrama interior del componente cuando esté sencillo.

Tabla 8. Tabla de posición de los nombres en diagrama PinOut.

			Dimencion de figura en in		Posición Norte, Sur etc	Cantidad	Coordenadas		Offset en %	Nombres de terminales																						
	PinOut	Enpaquetado	X	Y			X	Y																								
1	7400-0	DIP300	2.77	1.56	N2	7	0.22	0.05	0.39	Vcc	4B	4A	4Y	3B	3A	3Y	S2	7	0.22	1.56	0.39	1A	1B	1Y	2A	2B	2Y	Gnd				
2	7400-2	PLCC	3.25	3.25	N2	5	0.86	0.05	0.4	1B	1A	NC	Vcc	4B	E2	5	0	0.8	0.4	4A	NC	4Y	NC	3B	S2	5	0.86					3.2
3	7402-0	DIP300	2.77	1.56	N2	7	0.22	0.05	0.39	Vcc	4Y	4B	4A	3Y	3B	3A	S2	7	0.22	1.56	0.39	1Y	1A	1B	2Y	2A	2B	Gnd				
4	7404-0	DIP300	2.77	1.56	N2	7	0.22	0.05	0.39	Vcc	6A	6Y	5A	Y5	4A	4Y	S2	7	0.22	1.56	0.39	1A	1Y	2A	2Y	3A	3Y	Gnd				
5	7442-0	DIP300	3.1	1.56	N2	7	0.22	0.05	0.39	0	1	2	3	4	5	6	Gnd	S2	8	0.22	1.56	0.39	Vcc	A	B	C	D	9				
6	7446-0	DIP300	3.1	1.56	N2	7	0.22	0.05	0.39	B	C	LT ¹	BI ¹ /RBO ¹	RBI ¹	D	A	Gnd	S3	8	0.22	1.56	0.39	Vcc	f	g	a	b	c				
7	7468-0	DIP300	3.1	1.56	N2	7	0.22	0.05	0.39	1CLKA	1QB	QD	1CLR ¹	2QC	NC	2QA	Gnd	S4	8	0.22	1.56	0.39	Vcc	1CLKB	1QA	1QC	QD	2CLR				
8	7469-0	DIP300	3.1	1.56	N2	7	0.22	0.05	0.39	1CLKA	1QB	QD	1CLR ¹	2QC	NC	2QA	Gnd	S5	8	0.22	1.56	0.39	Vcc	1CLKB	1QA	1QC	QD	2CLR				
9	74116-0	DIP300	3.55	1.56	N2	7	0.22	0.05	0.39	1CLR ¹	1C1 ¹	1C2 ¹	1D1	1Q1	1D2	1Q2	1D3	1Q3	1D4	1Q4	Gnd	S2	10	0.215	1.56	0.39	Vcc	2Q4				
10	74150-0	DIP300	3.55	1.56	N2	7	0.22	0.05	0.39	E7	E6	E5	E4	E3	E2	E1	E0	G ¹	W	D	Gnd	S3	10	0.215	1.56	0.39	Vcc	E8				
11	74181-0	DIP600	3.55	1.56	N2	7	0.22	0.05	0.39	B0 ¹	A0 ¹	S3	S2	S1	S0	Cn	M	F0 ¹	F1 ¹	F2 ¹	Gnd	S4	10	0.215	1.56	0.39	Vcc	A1 ¹				

-Diseño de placa ZIF

Para el desarrollo del probador de circuitos digitales, se busca que el diseño sea lo más simple posible, dado que una de las ventajas de la construcción de este proyecto es que los alumnos puedan construirlo por ellos mismos, por lo tanto, se requiere que su diseño sea fácil de llevarlo a cabo físicamente, con pocos componentes y de fácil acceso a estos. En el diseño del probador de circuitos digitales se emplean componentes comerciales como se muestran en la Imagen 12, En esta parte del proyecto se desarrolla la placa de interface de la base tipo ZIF (Zero Insertion Force) para montar los componentes electrónicos digitales a probar. La placa de base ZIF se ensambla directamente en el módulo de desarrollo tipo arduino Mega2560 o compatible.

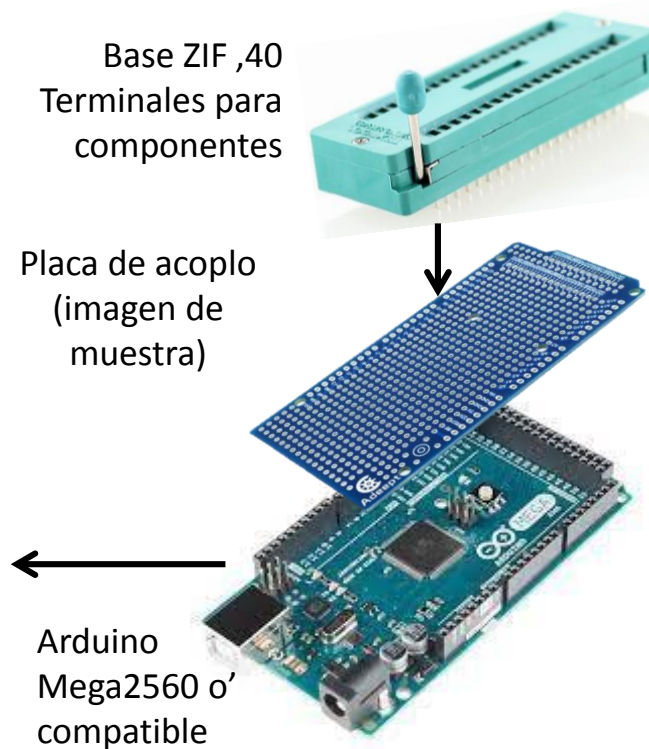
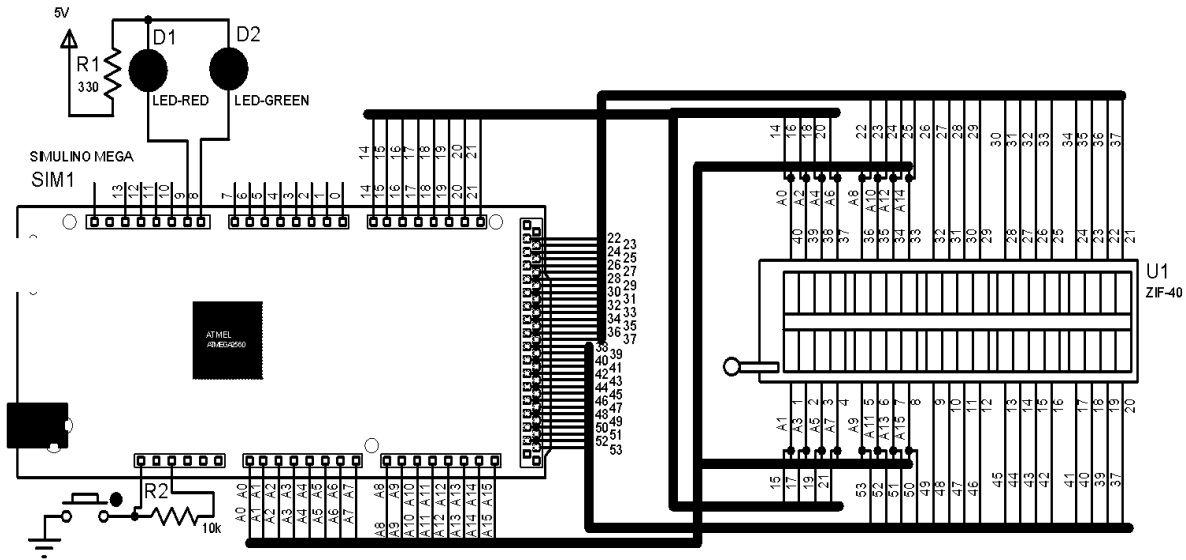


Imagen 12. Probador de circuitos TTL y CMOS propuesto (40 Terminales).

Diagrama esquemático

El diagrama utilizado para la construcción del proyecto probador de circuitos digitales se muestra en la Imagen 13. En este diagrama se muestra la base tipo ZIF conectada al módulo mega2560 la razón de por qué se utiliza ese módulo es que tiene suficientes salidas para conectar las 40 terminales de la base ZIF y permite probar componentes de dos terminales hasta 40 terminales.

Aunque en el diagrama de la Imagen 13, se puede observar complejo porque tiene muchos trazos, en realidad son solo conexiones entre la base de componente tipo ZIF con el módulo de desarrollo del arduino y como componentes adicionales se tienen dos LEDs y una resistencia, de esa manera, el número de componentes en realidad son muy pocos para el proyecto que se está desarrollando.



Construcción de placa de acoplo.

La construcción de la placa para conectar la base ZIF se puede elaborar en tres niveles de desarrollo dependiendo del propósito principal de uso.

Circuito prototipo en tablilla de experimentos (Protoboard) (Nivel 1)

Se desarrolla el prototipo de manera experimental solo como un medio de ejemplo. Prototipo experimental de laboratorio usando tablilla de experimentos de electrónica, de tal manera que el alumno puede construir el sistema probador de circuitos rápidamente con propósito demostrativo y con el mínimo de recurso.

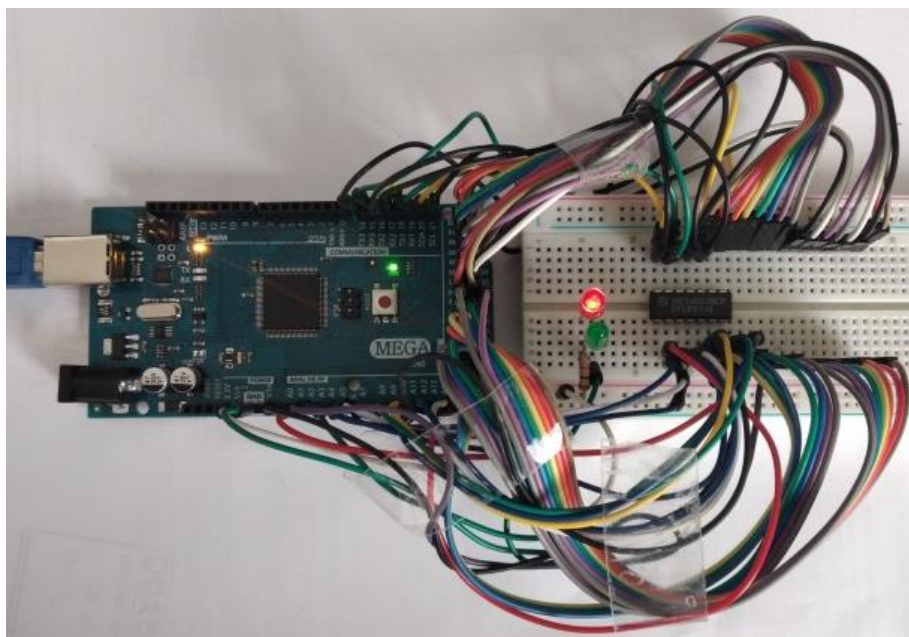


Imagen 15.Circuito prototipo en tablilla de experimentos (Nivel 1).

En la Imagen 15. Se observar la construcción del circuito en tablilla de experimentos, que es principalmente para propósitos demostrativos y comprobar su funcionamiento, porque no es práctico utilizarlo de ese modo, dado que es una construcción de manera temporal. Se observa en la imagen, el uso de la tablilla y un cableado de conectores prefabricados que conectan el módulo de arduino con la tablilla, de esa manera se construye rápidamente el probador de circuitos digitales, aunque tiene todo el funcionamiento de la lógica de prueba la construcción es de manera temporal.

Circuito prototipo en Impreso pre-fabricado (Nivel 2)

La construcción del probador de circuitos lógicos también se puede hacer usando una placa de circuito impreso PCB (Printed Circuit Board) prefabricada y de uso comercial para montaje de proyectos en módulos de arduino Mega2560, permite que tanto los alumnos como profesores puedan construir o ensamblar manualmente el probador de circuitos a un nivel de uso práctico y totalmente funcional.

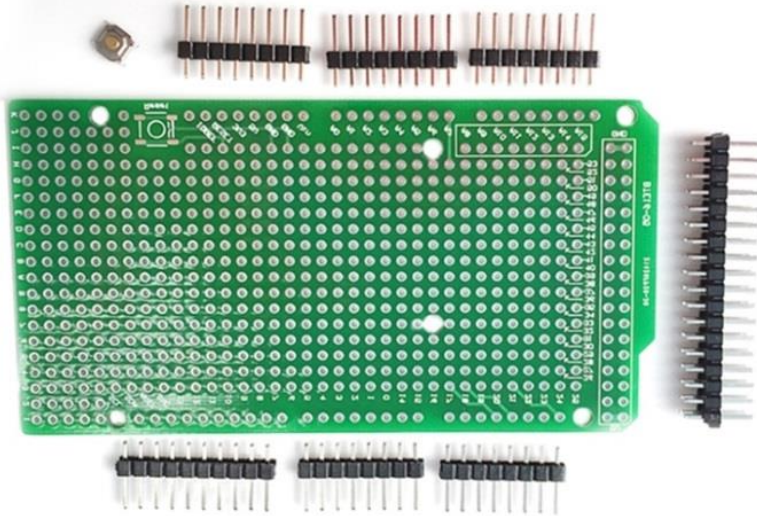


Imagen 16, Placa PCB tipo Sobre Montaje (Shield) para arduino Mega 2560.

En la en la Imagen 16, se muestra una fotografía de la placa que se puede adquirir comercialmente, incluye los conectores que requiere para ser montada sobre el módulo arduino 2560, de tal manera que solo se tiene que cablear y montar componentes en la perforaciones prefabricadas de acuerdo al diagrama, este medio de construcción es un modo rápido de obtener el desarrollo del probador de circuitos totalmente funcional. Comercialmente es fácil de obtener esta placa de PCB ya que existen muchos proveedores y su precio es relativamente bajo.

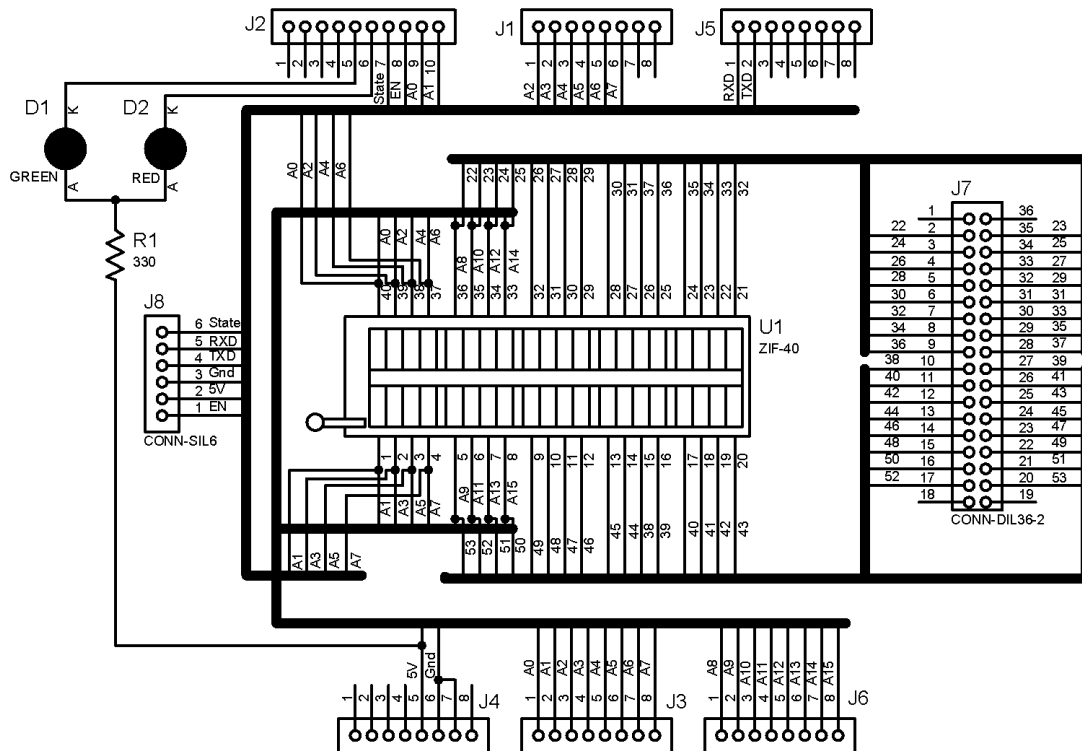


Imagen 17. Diagrama esquemático de la placa PCB y base ZIF.

En la Imagen 17 se tiene el diagrama esquemático a ser ensamblado en la placa de circuito impreso comercial, el diagrama ya no incluye lo que es el módulo mega2560, en este diagrama se muestran los conectores aproximadamente en la posición en que se ubican en la placa del circuito impreso, de tal manera que facilite la conexión para poder hacer un ensamble manual menos tedioso.



Imagen 18. Vista superior de Placa ensamblada manualmente.

La fotografía que se observa en la Imagen 18 es una vista superior del ensamble manual del probador de circuitos usando la placa PCB comercial, también se observan la base ZIF para el componente a probar y dos LEDs indicadores del estado de operación del probador, **Operación** y **Desconectado**, El estado desconectado es para intercambiar el componente a probar y reducir riesgo de daño.

Circuito prototipo en impreso profesional (Nivel 3)

Un modo muy profesional de elaborar el probador de circuitos digitales, es desarrollando el trazado de circuito impreso con algún software para poder enviar el diseño de PCB a alguna empresa que fabrique circuitos impreso tipo PCB de calidad comercial o industrial, permitiendo que el alumno construya un probador de circuitos con una presentación excelente, así como manejo físico y la funcionalidad lógica total del probador.

Actualmente hay empresas que fabrican placas de PCB de 10cmx10cm, en lotes de 10 piezas a un precio de aproximadamente \$11.08 MXN c/u, e inclusive, esas mismas empresas cuentan con el software para desarrollar el trazado de circuitos impresos PCB de manera gratuita y es software de fácil manejo de tal manera que se puede aprender a usarlo en pocos días [7].

La Imagen 19, muestra el trazado PCB de la placa para montar la base ZIF del probador de circuitos digitales, esa imagen solo se puede apreciar bien si está a colores porque muestra la capa superior e inferior del trazado o enrutado de conexiones, así como la ubicación de los componentes. En este caso el trazado ha sido hecho manualmente para poder organizar las líneas de manera que sirva de guía si el circuito se construye con placa prefabricada de PCB, ya que está mostrando todo el trazado en líneas paralelas que ayude a construir el circuito.

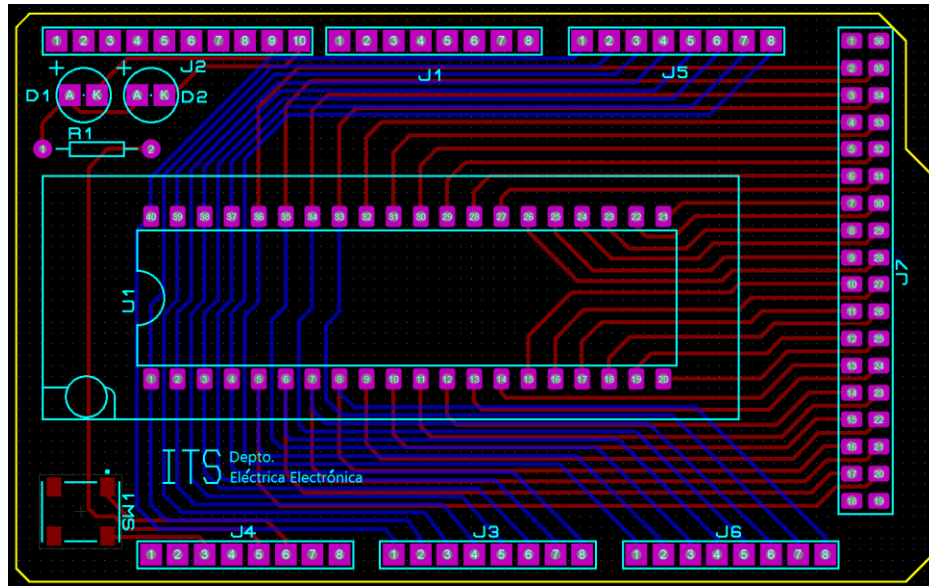


Imagen 19. Trazado de impreso de placa PCB-ZIF.

Sin embargo, ese trazado manual del circuito PSB es algo lento. Para la etapa de desarrollar un circuito de PCB de manera rápida pero profesional se puede recurrir a las características del Software de circuitos impresos donde se puede enrutar las líneas de conexión de manera automática como se muestra en la Imagen 20.

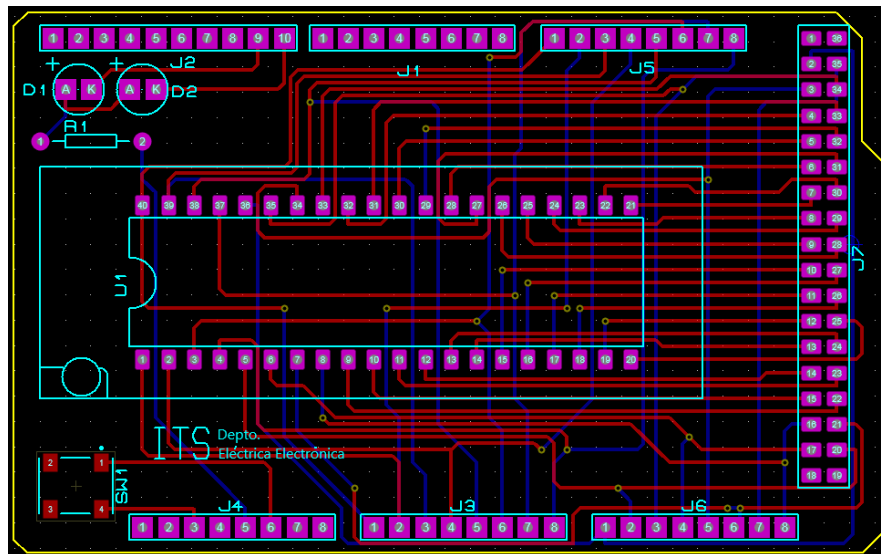


Imagen 20. Trazado de impreso PCB en software de modo automático.

El modo de trazar circuitos impresos en software de manera automática, como la Imagen 20, es muy eficaz porque solo se tiene que desarrollar el circuito esquemático, ubicar los componentes y trazar el auto enrutado de las líneas de conexión, aunque a la vista parezca desordenado el circuito queda

trazado eléctricamente correcto. El diseño se envía a fabricar a alguna empresa y lo construyen sin ningún problema.

En este caso se está haciendo un diseño de propósito educativo y orientado a que los alumnos lo puedan construir fácilmente. El trazado de la Imagen 19, es más claro y entendible pero eléctricamente es el mismo que el de la Imagen 20.

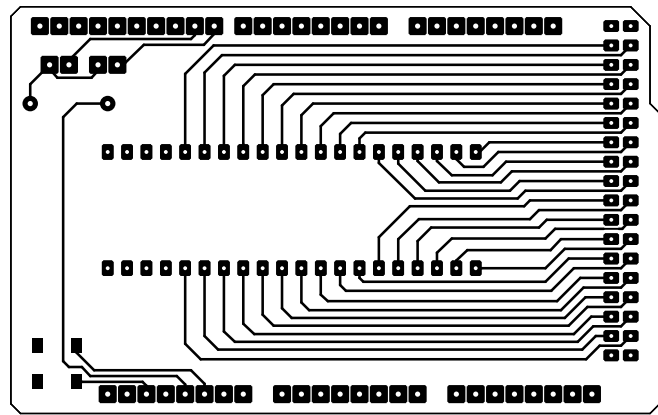


Imagen 21. Vista de la capa superior del circuito PCB de la imagen 12.

Con la finalidad de apreciar el trazado del circuito impreso PCB en imágenes que no son a colores, la Imagen 21, muestra la vista Superior o la capa superior del enrutado y se aprecia que el trazado de las líneas está directamente relacionada con la posición de las terminales de los componentes tanto de los conectores como de la base ZIF de 40 terminales, de tal manera que si ese circuito se construye manualmente es muy sencillo seguirlo porque todas las líneas están de una forma paralela.

La Imagen 22, es una vista de la capa inferior del circuito PCB, dado que este circuito fue trazado a dos capas se observa la misma situación que la Imagen 21, las conexiones están hechas con trazos paralelos lo que facilita que al construirse este circuito manualmente en una placa prefabricada se puedan seguir las líneas y se reduzca la posibilidad de cometer errores al conectar el cableado.

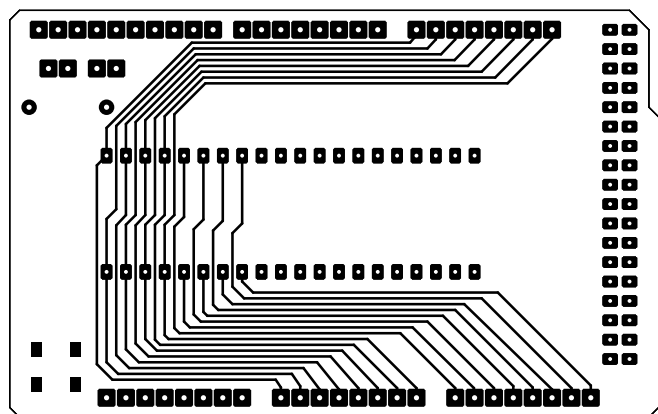


Imagen 22. Vista de la capa inferior del circuito PCB de la imagen 12.

-Algoritmos de pruebas Combinacionales y Secuenciales

En la Imagen 24 se tiene el diagrama de flujo para cuando se ejecuta todo el proceso de pruebas en forma automática. En el lado derecho se muestra un ciclo de forma que lo primero es leer la tabla de pruebas lógicas y en cada prueba leída llama a un método que ejecuta esa prueba, ya que la tabla leída contiene el nombre de la prueba y los parámetros que requiere la prueba. En el centro de la Imagen 24 se listan las pruebas posibles que se pueden ejecutar en un circuito según la lógica interna. Cada circuito requiere de una o varias pruebas, a su vez cada prueba requiere de ejecutar uno o varios de los comandos listados en la Imagen 24. Estos comandos estructuran un bloque de datos que se transmiten directamente al módulo del arduino.

28

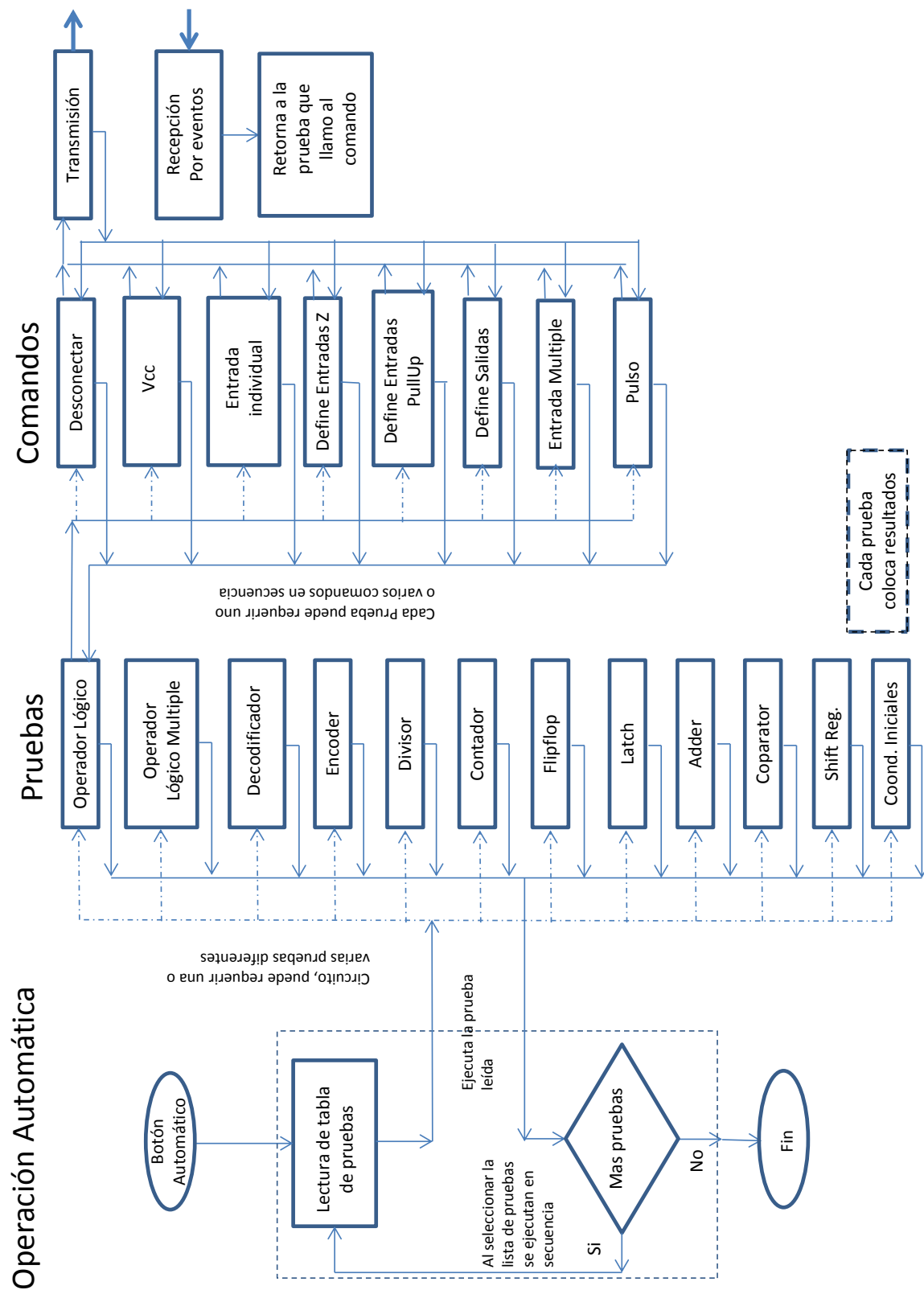


Imagen 24. Diagrama a bloque de la estructura de pruebas lógicas.

Variables globales del sistema

Para el desarrollo del sistema se definen algunas variables globales porque sus datos se requieren en varios métodos o procesos y a su vez nos permiten la transferencia de parámetros.

Variables: Globales

Declarar global: Arreglo DatosTx[6] tipo Byte o' Char, es la cadena que va a Transmitir
Declarar global: Arreglo DatosRx[6] tipo Byte o' Char, es la cadena que va a Recibir
Declarar global: Arreglo Num_Par_Tx[8] tipo Byte o' Char, bytes a Transmitir
Num_Par_Tx[0,2,2,5,5,5,5,2] Parametra de transmisión en comandos
Declarar global: Arreglo Num_Par_Rx[8] tipo Byte o' Char, Bytes a Transmitir
Num_Par_Rx[0,0,5,0,0,0,5,5 Num_Par_Tx[0,2,2,5,5,5,5,2] Parametros de resepción en comandos
NumByte, tipo Byte o' Char Posicion del byte transmitido o recibido en la cadena
Comando Tipo Boolean cadena completa

Declarar global: Arreglo Entradas[5] tipo Byte o' Char, es la cadena de cuales terminales (40) son entradas
Declarar global: Arreglo Salidas[5] tipo Byte o' Char, es la cadena de cuales terminales (40) son salidas
Declarar global: Arreglo Transmitido Anterior[5] tipo Byte o' Char, Ultimo estatus de las entradas transmitidas
Declarar global: Arreglo Recivido Anterior[5] tipo Byte o' Char, Ultimo estatus de las salidas recibidas
Declarar global: Arreglo Alimentacion[2] Terminales de Vcc y Gnd

Algoritmos de Comunicación

Se tiene dos algoritmos para la comunicación, transmisión y recepción, estos se representan a través de seudo código.

Nombre del Método: Recibir (Recepción por eventos)

Parámetros de entrada:

Vartiable DatosRx[6] Pasa globalmente
Estructura de la cadena DatosRx [Código del comando, Par1,Par2 . . . Par5]

Variables locales:

Codigo, tipo Byte o' Char
Parametros

Seudo código:

La variable NumByte, es inicializada con cero en el arranque del programa
Si <Dato recibido en el puerto serie> Entonces
 DatosRx[NumByte]=Lectura_de_un_byte_del_puerto_serie
 Numbyte=NumByte+1
 Codigo= DatosRx[0] El codigo es almacenado en el elemento cero
 Parametros=Num_Par_Rx[Código-1]+1 Parametros del comando
 Si NumByte=Parametros Entonces
 Numbyte=0 Reinicia conteo para siguiente comando
 Comando= Verdadero Bandera para indicar comando completo
 Fin de condicional
Fin de condicional

Fin del Método:

Nombre del Método: Transmitir

Parámetros de entrada:

Vartiable DatosTx[6] Pasa globalmente
Estructura de la cadena DatosTx [Código del Comando, Par1,Par2 . . . Par5]

Variables locales:

Codigo,Dato Tipo Byte o' Char
Parametros, Tipo Byte o' Char

Seudo código:

Codigo=DatosTx [0] Contiene el código del comando a transmitir
Parametros= Num_Par_Tx[Codigo] Lee cuantos parámetros tiene el código del comando
Para Dato=0 Hasta Parametros
 TransmitirByte (DatosTx [Dato]) Transmite toda la cadena, el código y los parámetros
 Siguiete Dato

Fin del Método:

Algoritmos de Pruebas Combinacionales

En el diagrama bloques de la Imagen 24 se muestra la lista de todas las pruebas posibles de los circuitos digitales y en la Tabla 9 se muestra la lista de las pruebas que corresponden a circuitos combinacionales [8], para los cuales en esta sección se presentan los algoritmos.

Tabla 9. Pruebas lógicas Combinacionales.

Pruebas lógicas combinacionales
Operador lógico
Condiciones iniciales
Demultiplexor(decoder)
Multiplexor
Encoder
Comparador
Sumador

Nombre del Método: Operador Lógico

Descripción: Ejecuta la prueba en una compuerta individual
AND,OR NOT etc. los parámetros son: función lógica, terminales de entrada y terminal de salida

Parámetros de entrada:

Vartiable Prueba[6] *Pasa globalmente*
Estructura de la cadena Prueba [Función, Par1,Par2 . . . Par5]

Variables locales:

Dato, Tipo Byte o' Char

Seudo código:

```
Para Dato=0 Hasta 2^Entradas
  DatosTx[6] = Combinación de entrada(Dato)
  Llama a EntradaMultiple (DatosTx[])
  Espera respuesta
  Procesa la salida(DatosRX)
Siguiente Dato
```

Fin del Método:

Nombre del Método: Condiciones iniciales

Descripción: Asigna valores predeterminadas a las entradas

Parámetros de entrada:

Vartiable Prueba[6] *Pasa globalmente*
Estructura de la cadena Prueba [Función, Par1,Par2 . . . Par5]

Variables locales:

Seudo código:

```
DatosTx[6] = Carga combinación de entrada
Llama a EntradaMultiple (DatosTx[])
Espera respuesta
Procesa todaslas salidas(DatosRX)
```

Fin del Método:

Nombre del Método: Demultiplexor (decodificado)

Descripción: Revisa secuencia de 1 a n Demultiplexor

Parámetros de entrada:

Vartiable Prueba[6] *Pasa globalmente*
Estructura de la cadena Prueba [Función, Par1,Par2 . . . Par5]

Variables locales:

Dato, Tipo Byte o' Char

Seudo código:

```
Entradas= Lee cuantas entradas
Para Selector=0 Hasta 2^Entradas
  DatosTx[] = Carga(Selector)
  DatosTx[] = Carga(Entrada cero)
  Llama a EntradaMultiple (DatosTx[])
  Espera respuesta
  Procesa salida seleccionada(DatosRX)
```



```
DatosTx[] = Carga(Entrada Uno)
Llama a EntradaMultiple (DatosTx[])
Espera respuesta
Procesa salida seleccionada(DatosRX)
Siguiente Selector
```

Fin del Método:

Nombre del Método: Multiplexor

Descripción: Revisa secuencia de n a 1 multiplexor

Parámetros de entrada:

Vartiable Prueba[6] [Pasa globalmente](#)
[Estructura de la cadena Prueba \[Función, Par1,Par2 . . . Par5\]](#)

Variables locales:

Selector, Tipo Byte o' Char

Seudo código:

```
Entradas= Lee cuantas entradas
Para Selector=0 Hasta 2^Entradas
  DatosTx[] = Carga(Selector)
  DatosTx[] = Carga(Entrada cero,Selector)
  Llama a EntradaMultiple (DatosTx[])
  Espera respuesta
  Procesa la salida(DatosRX)
  DatosTx[] = Carga(Entrada Uno, ,Selector)
  Llama a EntradaMultiple (DatosTx[])
  Espera respuesta
  Procesa la salida (DatosRX)
Siguiente Selector
```

Fin del Método:

Nombre del Método: Comparador

Descripción: Compara dos números binarios

Parámetros de entrada:

Vartiable Prueba[6] [Pasa globalmente](#)
[Estructura de la cadena Prueba \[Función, Par1,Par2 . . . Par5\]](#)

Variables locales:

A,B, Tipo Byte o' Char

Seudo código:

```
A= Lee dato de entra A
B= Lee dato de entra B
DatosTx[] = Carga(A en A)
DatosTx[] = Carga(B en B)
Llama a EntradaMultiple (DatosTx[])
Espera respuesta
Procesa A>B(DatosRX)
Procesa A<B(DatosRX)
Procesa A=B(DatosRX)
DatosTx[] = Carga(B en A)
DatosTx[] = Carga(A en B)
Llama a EntradaMultiple (DatosTx[])
Espera respuesta
Procesa A>B(DatosRX)
Procesa A<B(DatosRX)
Procesa A=B(DatosRX)
```

Fin del Método:

Nombre del Método: Sumador

Descripción: Compara dos números binarios A y B

Parámetros de entrada:

Vartiable Prueba[6] [Pasa globalmente](#)
[Estructura de la cadena Prueba \[Función, Par1,Par2 . . . Par5\]](#)

Variables locales:

A,B, Carry, Tipo Byte o' Char

Seudo código:

```
Entradas= Lee cuantos bits
Para A=0 Hasta 2^Entradas
Para B=0 Hasta 2^Entradas
Para Carry=0 Hasta 1
  DatosTx[] = Carga(A)
```

```

DatosTx[] = Carga(B)
DatosTx[] = Carga(Carry)
Llama a EntradaMultiple (DatosTx[])
Espera respuesta
Procesa Suma(DatosRX)
Siguiente Carry
Siguiente Dato B
Siguiente Dato A

```

Fin del Método:

Algoritmos de Pruebas Secuenciales

De la lista de pruebas de la Imagen 24 se extrae la lista de las pruebas que corresponden a circuitos Secuenciales [9] y se muestran en la Tabla 10.

Tabla 10. Pruebas lógicas Secuenciales.

Pruebas lógicas Secuenciales
Flip-Flop
Latch
Contador
Registro de corrimiento

Nombre del Método: Flip-Flop Set

Descripción: Prueba individual de elemento de memoria Flip-Flop tipo R-S,D,J-K

Parámetros de entrada:

Vartiable Prueba[6] Pasa globalmente
Estructura de la cadena Prueba [Función, Par1,Par2 . . . Par5]

Variables locales:

Set,Q,Q' Tipo Byte o' Char

Seudo código:

```

Set= Lee núm. Terminal
Q= Lee núm. Terminal
Q'= Lee núm. Terminal
Llama a Pulso en(SDn,0 o' 1
Espera respuesta
Procesa QnQn'Set(DatosRX)

```

Fin del Método:

Nombre del Método: Flip-Flop CLR

Descripción: Prueba individual de elemento de memoria Flip-Flop tipo R-S,D,J-K

Parámetros de entrada:

Vartiable Prueba[6] Pasa globalmente
Estructura de la cadena Prueba [Función, Par1,Par2 . . . Par5]

Variables locales:

Clr,Q,Q' Tipo Byte o' Char

Seudo código:

```

Clr= Lee Lee núm. Terminal
Q= Lee Lee núm. Terminal
Q'= Lee Lee núm. Terminal
Llama a Pulso en(SDn,0 o' 1
Espera respuesta
Procesa QQ' Clear(DatosRX)

```

Fin del Método:

Nombre del Método: Flip-Flop D-Latch

Descripción: Prueba individual de elemento de memoria Flip-Flop tipo R-S,D,J-K

Parámetros de entrada:

Vartiable Prueba[6] Pasa globalmente
Estructura de la cadena Prueba [Función, Par1,Par2 . . . Par5]

Variables locales:

TCLK,TQ,TQ' Tipo Byte o' Char

Seudo código:

```

TCLK= Lee núm. Terminal
TQ= Lee núm. Terminal
TQ'= Lee núm. Terminal
TD= Lee núm. Terminal
Asigna (DatosTx[],D=0)
Llama a Pulso en(CLK,0 o' 1
Espera respuesta
Procesa QQ'D(DatosRX)
Asigna (DatosTx[],D=1)
Llama a Pulso en(CLK,0 o' 1
Espera respuesta
Procesa QQ'D(DatosRX)

```

Fin del Método:

Nombre del Método: Flip-Flop J-K

Descripción: Prueba individual de elemento de memoria Flip-Flop tipo R-S,D,J-K

Parámetros de entrada:

Vartiable Prueba[6] [Pasa globalmente](#)
[Estructura de la cadena Prueba \[Función, Par1,Par2 . . . Par5\]](#)

Variables locales:

TCLK,TQ,TQ',TJ,TK Tipo Byte o' Char

Seudo código:

```

TCLK= Lee núm. Terminal
TQ= Lee núm. Terminal
TQ'= Lee núm. Terminal
TJ= Lee núm. Terminal
TK= Lee núm. Terminal
Asigna (DatosTx[],J=0)
Asigna (DatosTx[],K=0)
Llama a EntradaMultiple (DatosTx[])
Llama a Pulso en(CLK,0 o' 1
Espera respuesta
Procesa QQ'NoCambio(DatosRX)
Asigna (DatosTx[],J=0)
Asigna (DatosTx[],K=1)
Llama a EntradaMultiple (DatosTx[])
Llama a Pulso en(CLK,0 o' 1
Espera respuesta
Procesa QQ'Clear(DatosRX)
Asigna (DatosTx[],J=1)
Asigna (DatosTx[],K=0)
Llama a EntradaMultiple (DatosTx[])
Llama a Pulso en(CLK,0 o' 1
Espera respuesta
Procesa QQ'Set(DatosRX)
Asigna (DatosTx[],J=1)
Asigna (DatosTx[],K=0)
Llama a EntradaMultiple (DatosTx[])
Llama a Pulso en(CLK,0 o' 1
Espera respuesta
Procesa QQ'Togle(DatosRX)

```

Fin del Método:

Nombre del Método: Contador

Descripción: Prueba de conteo 0 a N

Parámetros de entrada:

Vartiable Prueba[6] [Pasa globalmente](#)
[Estructura de la cadena Prueba \[Función, Par1,Par2 . . . Par5\]](#)

Variables locales:

TCLK,TQ,TQ' Tipo Byte o' Char

Seudo código:

```

TCLK= Lee núm. Terminal
Conteo= Lee núm. Terminal
TQ= Lee núm. Terminal
Llama a Pulso en(CLK,0 o' 1
Espera respuesta
Procesa Conteo(DatosRX)

```

Fin del Método:

Nombre del Método: Registro de corrimiento SiPo

Descripción: Prueba desplazamiento en registros (SiPo, PiSo, SiSo)

Parámetros de entrada:

Vartiable Prueba[6] Pasa globalmente

Estructura de la cadena Prueba [Función, Par1, Par2 ... Par5]

Variables locales:

TCLK, TQ, Tin, Bits Tipo Byte o' Char

Seudo código:

```
TCLK= Lee núm. Terminal
Tin= Lee núm. Terminal
Bits= Lee núm. bits
TQ= Lee núm. Terminal
Doto=0
Para A=0 Hasta Bits-1
  Dato=Not(Dato)
  Asigna (DatosTx[],In=Dato)
  Llama a Pulso en(CLK,0 o' 1
  Espera respuesta
  Procesa corrimiento(DatosRX)
Siguiente Bit
```

Fin del Método:

-Programación de interface gráfica en PC

El diseño del Software del probador de circuitos digitales, se está desarrollando de tal manera que la circuitería sea lo más simple posible, por lo tanto, todo el proceso se está llevando a cabo en el software, así se tiene la información de las terminales, valores que debe tomar cada entrada y/o salidas, si son grupos de terminales o es individual, alimentación de voltaje y toda la lógica de la ubicación del circuito relacionado a la posición en el módulo físico de prueba.

Diseño de la pantalla de inicio del programa

La Imagen 25 muestra la imagen de inicio del software del probador de circuitos. En forma general el software cuenta con seis áreas como se describen a continuación.

Área 1. Menú de operaciones. El menú de operaciones muestra una serie de iconos de trabajo.

-Selección, el primer icono es la operación de selección del componente a probar.

-Manual. Esta opción de trabajo es para hacer una prueba didáctica, se asigna valores directamente a las entradas del circuito y se muestra el resultado leído en las terminales de salida del circuito físico, esto, forma un espejo de los valores de entrada y salida en el circuito que se está probando, el estudiante la operación de la lógica del circuito.

-Auto. La opción de automático realiza la prueba del circuito en forma continua y a todo el componente. La estructura del software está preparada para realizar una o varias pruebas diferentes según la lógica interna del circuito, ejecuta las pruebas consecutivamente y muestra el estado de operación del componente.

-Stop. Esta opción es principalmente detener el proceso cuando está en modo manual, dado que en modo automático la ejecución se hace continua y se termina en su totalidad, así que no requiere detenerse el proceso a medio camino.

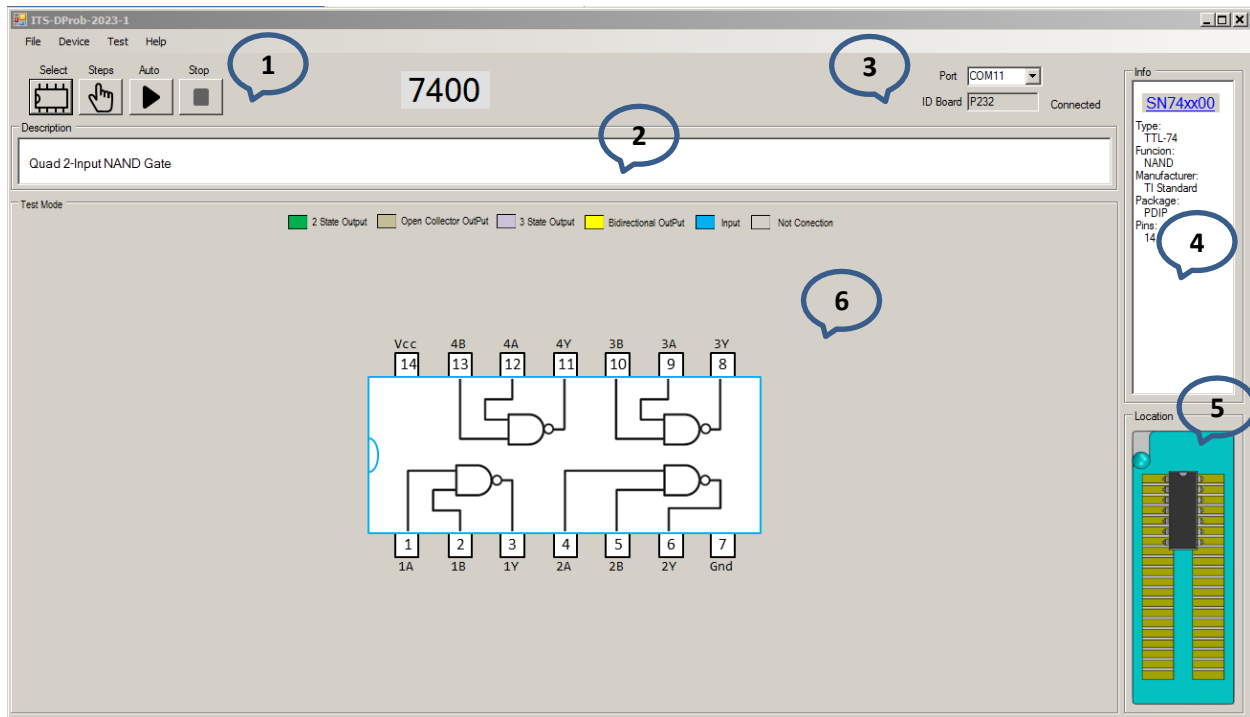


Imagen 25. Pantalla de inicio del software probador de circuitos.

Área 2, Descripción. Esta sección es para colocar la descripción del componente que identifica la función lógica que tiene este circuito, esa descripción la da el fabricante en la hoja de datos del componente.

Área 3, Comunicación. En el área 3 se muestran los controles de comunicación, se tiene la selección del puerto donde se detecta el circuito probador, se muestra la identificación del módulo probador y el estatus de conexión.

Área 4, Información. Es un espacio para colocar la información del componente como: Fabricante, empaquetado, acceso a la hoja de datos del componente y otros.

Área 5, Ubicación. Indica dónde se coloca y la orientación el componente físicamente en la base tipo ZIF del probador de circuitos.

Área 6, Área de trabajo. Esta es el área de trabajo donde se muestra el componente, terminales y sus nombres, así como, el estado lógico de las entradas y salidas y la condición de operación (Bueno o Dañado).

La programación se realizó utilizando programación visual [10], esto tiene la ventaja de que se construye una plantilla base llamada “Forma” (Form en inglés) como es el caso de la Imagen 25, donde está establecido la presentación de la información, se colocan botones de control como se muestran en esa misma imagen, para: seleccionar el componente, pruebas manuales o para hacer pruebas automáticas, también se definen áreas para presentar información como es el caso del área 4, donde se

presenta información de tipo texto y áreas de información donde se visualizan figuras como es el caso del área 5 y el área 6.

La programación visual también maneja programación por objetos y por eventos [11], como es el hecho de presionar un botón genera un evento y se ejecuta una sección del programa muy específica, por lo tanto eso le da una estructura de programación más versátil.

```
'##### Event
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    PortSelected = ""
    ChipSelected = ""
    Btn_ChipSelect.Enabled = False
    Btn_Steps.Enabled = False
    Btn_Stop.Enabled = False
    Btn_Auto.Enabled = False
    Btn_PinOut.Enabled = False
    LnkL_DataSheet.Visible = False
    ChipDisplay.Visible = False
    CreatePinLabels() 'Crea las 40 PinLabels de los nombres de terminales en modo invisible
    CreateIOsArrows()
    CreateResultLabels()
    ' Iconos(0) = My.Resources.Resources.Picture1
    LoadIOsArrows()
    TBox_IDBoard.Text = "IDBoard"
End Sub
```

Listado 1 inicio de la forma de la Imagen 25.

Dada la extensión del total del programa, se mostrarán secciones representativas de la operación de este, como es el caso del listado 1 donde se muestra la sección de programa que se ejecuta Al momento de iniciar el software de control del probador de circuitos. En este caso se definen algunos estatus como son la condición de operación de los botones de control como es, el botón de Selección de Componente, Operación Manual, Operación Automática, porque al inicio del programa no se ha seleccionado un componente digital de tal manera que algunos botones están deshabilitados, aunado a lo anterior se ejecuta una serie de procedimientos o subrutinas que van a crear la estructura inicial del programa.

```
Sub CreatePinLabels()
    ' Crea las etiquetas 1-40 de los nombres de terminales del circuito
    Dim Pin As Integer
    Dim myfont As New Font("Consolas", 10, FontStyle.Regular)
    For Pin = 40 To 1 Step -1
        PinLabels(Pin) = New Label()
        PinLabels(Pin).Visible = False
        PinLabels(Pin).AutoSize = True
        PinLabels(Pin).Font = myfont
        PinLabels(Pin).TextAlign = ContentAlignment.MiddleLeft
        Me.GroupBox2.Controls.Add(PinLabels(Pin))
    Next
End Sub
```

Listado 2 Crea los espacios de texto conocidos como “Etiqueta” ó “Labels” para los nombres de cada terminal del circuito.

El Listado 2, muestra una de los procedimientos que son llamados cuando se abre inicialmente el programa del probador de circuitos, en este procedimiento se crea los espacios para colocar el nombre de las terminales del circuito que se va a aprobar, estos espacios son objetos dentro de la programación y los objetos creados en el listado 2 son conocidos como “Etiquetas” o “Labels” en inglés. Cuando se selecciona un componente de prueba entonces se dispone de los nombres específicos de cada terminal y se colocan en la etiqueta que le fue asignada a cada una de las terminales.

```
Sub CreateIOsArrows()
    'Crea las PictureBox de las flechas de entrada y salida
    Dim Pin As Integer
    For Pin = 1 To 40
        PBox_IOs_Arrow(Pin) = New PictureBox()
        PBox_IOs_Arrow(Pin).Visible = False
        Me.GroupBox2.Controls.Add(PBox_IOs_Arrow(Pin))
        LB_IO_Status(Pin) = New Label()
        LB_IO_Status(Pin).Visible = False
        Me.PBox_IOs_Arrow(Pin).Controls.Add(LB_IO_Status(Pin))
    Next
    AddHandler DirectCast(PBox_IOs_Arrow(40), PictureBox).Click, AddressOf PBox_IOs_Arrow_40_Click
    AddHandler DirectCast(PBox_IOs_Arrow(39), PictureBox).Click, AddressOf PBox_IOs_Arrow_39_Click
    AddHandler DirectCast(PBox_IOs_Arrow(38), PictureBox).Click, AddressOf PBox_IOs_Arrow_38_Click
    .
    .
    AddHandler DirectCast(PBox_IOs_Arrow(3), PictureBox).Click, AddressOf PBox_IOs_Arrow_3_Click
    AddHandler DirectCast(PBox_IOs_Arrow(2), PictureBox).Click, AddressOf PBox_IOs_Arrow_2_Click
    AddHandler DirectCast(PBox_IOs_Arrow(1), PictureBox).Click, AddressOf PBox_IOs_Arrow_1_Click
End Sub
```

Listado 3 Creación de objetos tipo caja de imagen (PictureBox).

Listado 3 es de suma importancia porque crea el objeto tipo imagen que en programación se conoce como “PictureBox” donde se van a colocar las imágenes que le corresponden al icono indicador de entrada y salida de cada terminal del circuito digital, además es ese procedimiento crea la conexión como evento para cada uno de esas imágenes sea seleccionada al hacer click y ejecute un procedimiento específico.

-Programación del menú de selección

La Selección del componente es un menú tradicional de lección de columnas de opciones, en la Imagen 26 se muestra el menú de opciones para seleccionar el componente a probar. Se puede seleccionar por el tipo de circuito, TTL-54, TTL-74, CMOS y se ha agregado una opción para hacer auto prueba pero del propio módulo probador de circuitos.

Además, se puede seleccionar el componente a probar: Por la función lógica, por el fabricante o bien directamente por el número de parte en el renglón de búsqueda “Search”.

Una vez seleccionado el número de parte, este es el apuntador de todas las tablas de datos que contienen la información del componente para procesarlo tal como: Figura que va a representar al

circuito, nombres de las terminales, distribución física, descripción lógica del componente y principalmente qué pruebas se le van a hacer a cada componente.

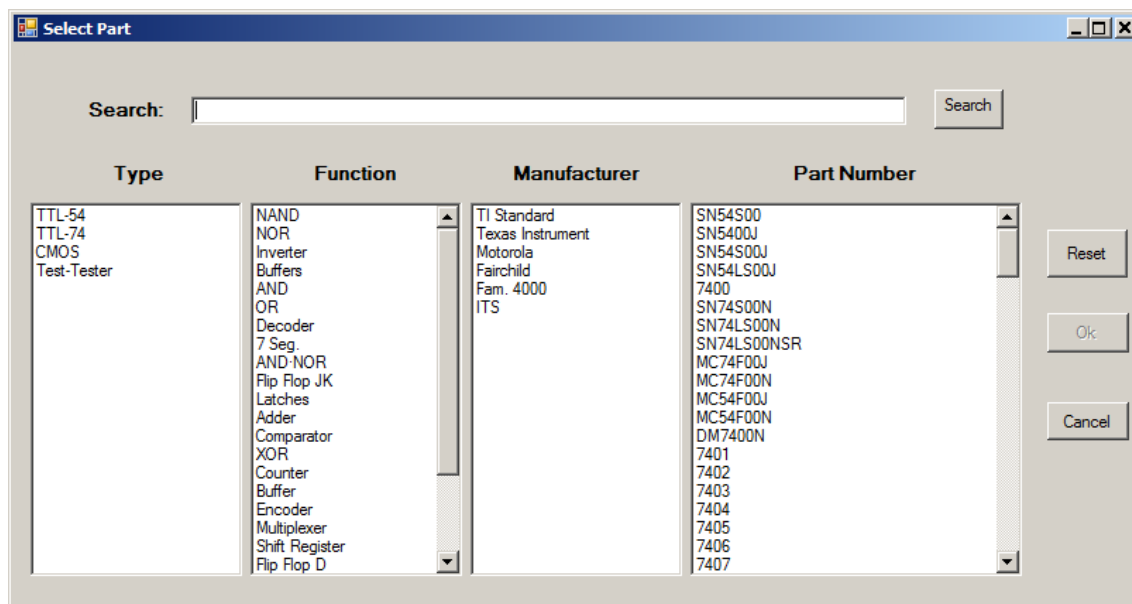


Imagen 26. Selección del componente a probar.

La información para llenar el menú de la Imagen 26 para la selección del componente de aprobar, se obtiene de la Tabla 11 de identificación del número de componente. En la tabla se tienen las siguientes columnas: Tipo de circuito, Función lógica, Fabricante, Número de parte que maneja el fabricante, aunque la función lógica sea la misma cada fabricante puede manejar un número de parte diferente y la columna IC-Info que determina el componente seleccionado en forma general y define la información del circuito como el nombre terminales, pruebas etc. que se muestran en tablas más adelante.

Tabla 11 Identificación del número de componente a probar (Tabla Parcial).

	Tipo	Funcion	Fabricante	Numero de Parte	IC Info
1	TTL-54	NAND	TI Standard	SN54S00	7400-1-TI
2	TTL-54	NAND	Texas Instrument	SN5400J	7400-1-TI
3	TTL-54	NAND	Texas Instrument	SN54S00J	7400-1-TI
4	TTL-54	NAND	Texas Instrument	SN54LS00J	7400-1-TI
1	TTL-74	NAND	TI Standard	7400	7400-1-TI
5	TTL-74	NAND	Texas Instrument	SN74S00N	7400-1-TI
6	TTL-74	NAND	Texas Instrument	SN74LS00N	7400-1-TI
7	TTL-74	NAND	Texas Instrument	SN74LS00NSR	7400-1-TI
8	TTL-74	NAND	Motorola	MC74F00J	7400-MO-DIP
9	TTL-74	NAND	Motorola	MC74F00N	7400-MO-DIP
10	TTL-74	NAND	Motorola	MC54F00J	7400-MO-DIP
11	TTL-74	NAND	Motorola	MC54F00N	7400-MO-DIP
12	TTL-74	NAND	Fairchild	DM7400N	7400-FA-DIP
13	TTL-74	NAND	TI Standard	7401	7401-1-TI
14	TTL-74	NOR	TI Standard	7402	7402-1-TI
15	TTL-74	NAND	TI Standard	7403	7403-1-TI
16	TTL-74	Inverter	TI Standard	7404	7404-1-TI

```

Private Sub Btn_Reset_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Btn_Ok.Enabled = False
    Fill_Type("All")
    Fill_Part("All") 'Condicion-"All"/"Inside", Llena Numero de Parte con toda la info
    Fill_Function("All")
    Fill_Manufacturer("All")
End Sub

Private Sub LBox_Type_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As Sys
    Btn_Ok.Enabled = False
    Fill_Function("Type")
    Fill_Manufacturer("Type")
    Fill_Part("Type")
End Sub

Private Sub LBox_Function_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
    Dim Condition As String
    Condition = ""
    Btn_Ok.Enabled = False
    If LBox_Type.SelectedIndex = -1 Then Condition = "Func"
    If LBox_Type.SelectedIndex > -1 Then Condition = "TypeFunc"
    Fill_Manufacturer(Condition)
    Fill_Part(Condition)
End Sub

```

Listado 4 Controles de selección de componente.

El listado 4, muestra los procedimientos que se ejecutan cuando se selecciona el botón de reset, (primera parte del listado) en esta sección se ejecutan comandos para llenar las columnas de la imagen 26 con la información. Ahora bien, si se selecciona la opción tipo de circuitos TTL-74, entonces se va a llenar la columna “función” solamente con los circuitos que corresponden al grupo de TTL-74, y de igual manera la columna del “fabricante” se llena con los datos solamente de los circuitos que pertenecen al grupo de TTL-74. En general, cada columna se llena bajo esa lógica, de tal manera que se va delimitando el circuito que se busca y finalmente se selecciona el componente en la columna correspondiente a “Número de parte”.

```

Private Sub Fill_Function(ByVal Condition As String)
    Dim i, j As Integer
    Dim Duplicate As Boolean
    Dim TextType As String
    '-----
    LBox_Function.Items.Clear()
    If Condition = "All" Then ' Llena Function con toda la informacion
        LBox_Function.Items.Add(VarMatriz(1, 2))
        For i = 1 To NumParts
            Duplicate = False
            For j = 0 To LBox_Function.Items.Count - 1
                If VarMatriz(i, 2) = LBox_Function.Items(j) Then
                    Duplicate = True
                    Exit For
                End If
            Next
            If Duplicate = False Then
                LBox_Function.Items.Add(VarMatriz(i, 2))
            End If
        Next
    End If
End Sub

```

Listado 5 Proceso de llenado de la columna función en el menú de selección.

El listado 5 muestra un fragmento de la sección del programa que llena la columna de la función lógica en el menú de selección de componente, aquí se tienen dos casos, uno cuando se llena con todas las funciones lógicas que existen y dos, cuando se ha seleccionado un tipo de circuito, entonces se llena solamente con las funciones lógicas que corresponden al tipo seleccionado.

-Programación del área de prueba de circuitos

La Imagen 27 muestra la pantalla de trabajo del circuito donde se muestra la operación lógica del circuito, por medio de flechas se indica el sentido de la señal lógica, el color de la flecha indica el tipo de terminal (Salida 2 Estados, Salida Colector Abierto, Salida de 3 Estados, Etc.). En la parte de arriba del circuito mostrado en la Imagen 27 se muestra la codificación de los colores de las flechas de entrada y salida.

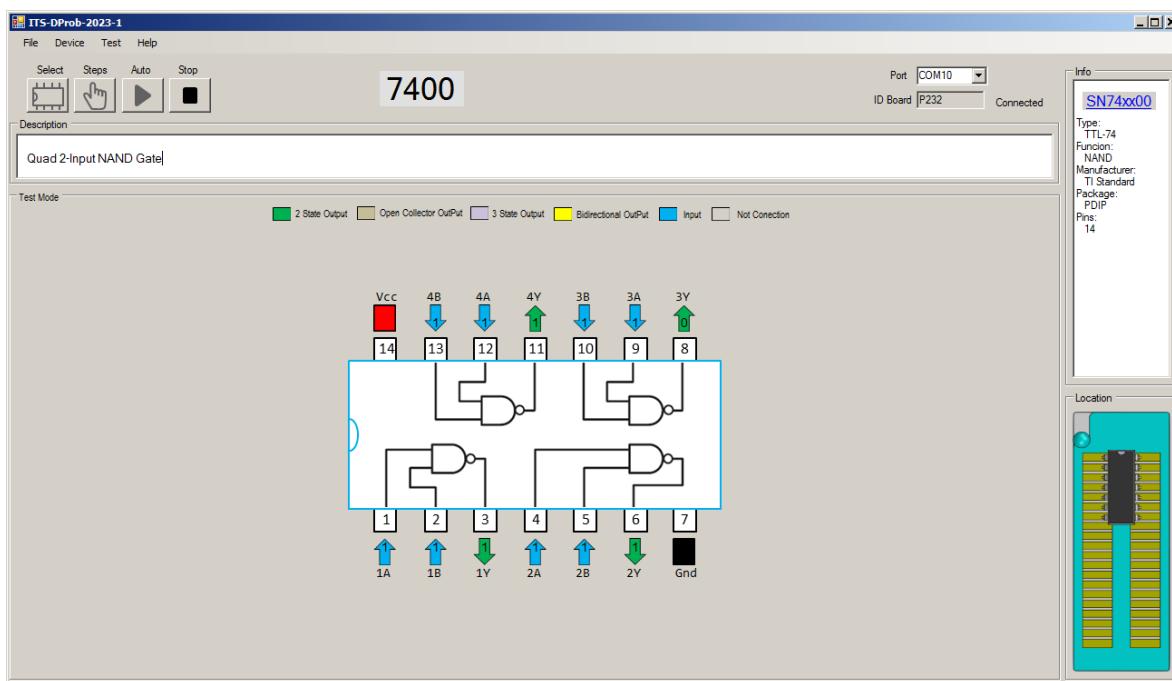


Imagen 27. Pantalla de trabajo de prueba de circuitos.

El valor del estado lógico 0 ó 1 de las entradas y salidas del circuito, aparece en cada una de las flechas indicadoras, como se muestra en la Imagen 28.

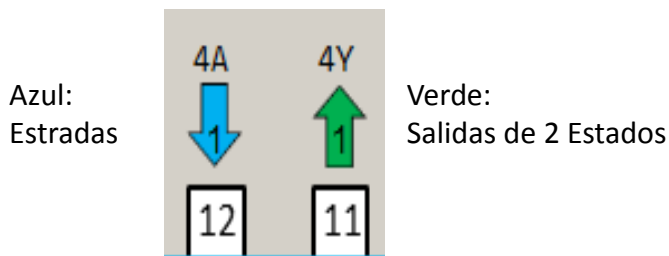


Imagen 28 Flechas indicadoras de sentido lógico y su valor, 0 ó 1.

Las pruebas de circuitos se pueden realizar de dos formas.

Prueba manual- La prueba manual se activa seleccionando el icono de la manita mostrado en la Imagen 29, y consiste en aplicar los valores lógicos en las entradas del circuito manualmente a través de hacer click en la parte libre de las flechas de sentido lógico como se muestra en la Imagen 30, causando una conmutación del valor lógico que tiene esa entrada, es decir, si se encuentra en valor 1 conmuta a valor 0. El módulo probador de circuitos aplica los nuevos valores de entrada y hace una lectura de todas las salidas del circuito, de tal manera que, cualquier salida que cambie de estado se ve reflejada en el valor lógico mostrado en pantalla. La decisión de si el circuito está funcionando correctamente es del usuario del módulo probador.

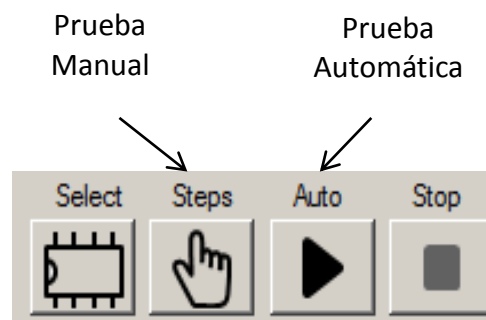


Imagen 29 Botón de prueba manual

Click de conmutación

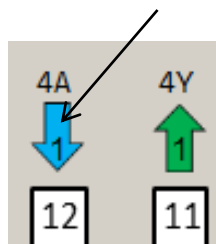


Imagen 30 Conmutación del valor de la entrada.

Prueba Automática. Para la prueba automática se tiene el botón “Auto” mostrado en la Imagen 29, en esta prueba se realiza una secuencia de conmutaciones de las entradas y el resultado de las salidas se compara con la lógica del circuito. Automáticamente se determina si el circuito está funcionando correctamente. En la Imagen 31 se muestra el resultado de una prueba automática, en esta prueba se indica con un palomeado verde si está correcto o bien tachado en rojo si esa parte de circuito no funciona. La secuencia de las pruebas se aplica en forma individual a cada compuerta. Para el circuito de la Imagen 31 que corresponde a un 7400 se realizan 4 pruebas de compuerta NAND de 2 entradas.

Esta forma de representar la operación de un circuito al momento de probarlo, es con propósito didáctico, porque muestra las secciones dañadas del componente y permite determinar fallas en un circuito experimental o practica de laboratorio.

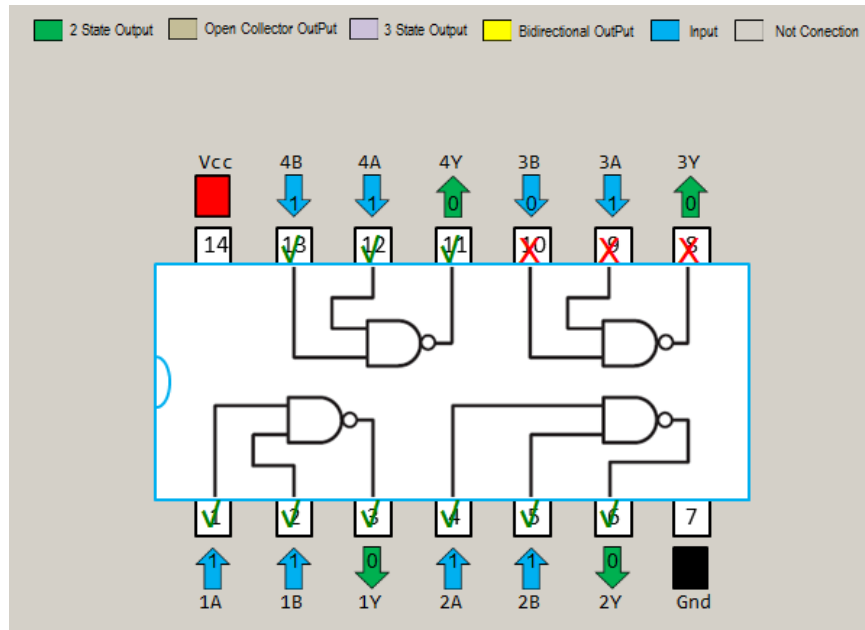


Imagen 31 Representación de Resultado de prueba Automática.

La programación del área visual de prueba de circuito mostrada en la Imagen 31, está ligado a tablas de información para representar: la imagen del circuito, las flechas indicadoras y el resultado. La imagen del diagrama electrónico del componente digital, es un objeto tipo imagen en el cual se carga un archivo que es señalado en la Tabla 12 columna 4, cada circuito seleccionado va a tener una imagen representativa de las terminales y del diagrama interno. El software cuenta con una carpeta donde están todos los archivos que son las imágenes de los circuitos que maneja el probador.

Tabla 12. Tabla IC-Info Parámetros del circuito seleccionado.

IC Info	Patas	Terminales Base ZIF	Distribución Física Terminales PinOut	Nombre Visuales de Terminales	Nombre Terminales PinOut	Descripcion	Prueba	Datasheet	Prueba Automatica
7401-1-TI	14	ZIF DIP-14-300-100	7401-1-TI-DL14.PNG	7401-1-TI	7401-1-TI	7401-0	7401-1-TI	SN74xx01	SI
7402-1-TI	14	ZIF DIP-14-300-100	7402-1-TI-DL14.PNG	7402-1-TI	7402-1-TI	7402-0	7402-1-TI	SN74xx02	SI
7403-1-TI	14	ZIF DIP-14-300-100	7403-1-TI-DL14.PNG	7403-1-TI	7403-1-TI	7403-0	7403-1-TI	SN74xx03	SI
7404-1-TI	14	ZIF DIP-14-300-100	7404-1-TI-DL14.PNG	7404-1-TI	7404-1-TI	7404-0	7404-1-TI	SN74xx04	SI
7405-1-TI	14	ZIF DIP-14-300-100	7405-1-TI-DL14.PNG	7405-1-TI	7405-1-TI	7405-0	7405-1-TI	SN74xx05	SI
7406-1-TI	14	ZIF DIP-14-300-100	7406-1-TI-DL14.PNG	7406-1-TI	7406-1-TI	7406-0	7406-1-TI	SN74xx06	SI

La tabla 11, también muestra en la columna 3 cuál es la imagen que indica dónde se coloca el componente en la base tipo ZIF. Además, la columna 5 apunta a la Tabla 13, donde se tienen los nombres de las terminales numerados del 1 al 40, de acuerdo a su posición en la base ZIF como en la Imagen 32, la columna 6 es parecida a la columna 5, pero esta apunta a una tabla semejante a la Tabla 13 pero con los nombres genéricos de las terminales (no dependen del Fabricante), Estos nombres genéricos se usan internamente en el software al ejecutar la secuencia de pruebas.

Aunque la tabla 11 parece que tiene columnas con datos redundantes, esto es porque la tabla se llenó con la información de los circuitos diseñados por la compañía Texas instrument, principalmente de la versión estándar. Algunos fabricantes de componente digitales colocan nombres diferentes en las terminales, cambia su descripción o inclusive algunos fabricantes tiene números de parte de componentes con tecnología de fabricación como LS,F,HC etc. que otros no tienen en su catálogo. Por lo anterior, las tablas de datos están diseñadas para poder ser actualizadas con librería con componentes de diferentes fabricantes.

Tabla 13 Nombre de terminales del circuito.

		Patas	1	2	3	4	5	6	7	8	9	10	11	30	31	32	33	34	35	36	37	38	39	40	
150	Lab																								
1	7400-1-TI	14	1A	1B	1Y	2A	2B	2Y	Gnd									3Y	3A	3B	4Y	4A	4B	Vcc	
2	7400-FP	14	1A	1B	1Y	Vcc	2Y	2A	2B									3Y	3A	3B	Gnd	4A	4B	4Y	
3	7400-LCCC	20	NC1	1A	1B	1Y	NC2	2A	NC3	2B	2Y	Gnd				NC4	3Y	3A	3B	NC5	4Y	NC6	4A	4B	Vcc
4	7401-1-TI	14	1Y	1A	1B	2Y	2A	2B	Gnd									3A	3B	3Y	4A	4B	4Y	Vcc	
5	7401-2	14	1A	1B	1Y	Vcc	2Y	2A	2B									3Y	3A	3B	Gnd	4A	4B	4Y	
6	7401-3	20	NC1	1Y	1A	1B	NC2	2Y	NC3	2A	2B	Gnd				NC4	3A	3B	3Y	NC5	4A	NC6	4B	4Y	Vcc
7	7402-1-TI	14	1Y	1A	1B	2Y	2A	2B	Gnd									3A	3B	3Y	4A	4B	4Y	Vcc	
8	7403-1-TI	14	1A	1B	1Y	2A	2B	2Y	Gnd									3Y	3A	3B	4Y	4A	4B	Vcc	
9	7404-1-TI	14	1A	1Y	2A	2Y	3A	3Y	Gnd									4Y	4A	5Y	5A	6Y	6A	Vcc	

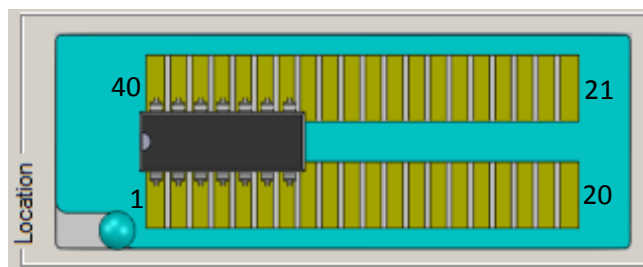


Imagen 32 Numeración de terminales del circuito según base ZIF.

El procedimiento para colocar los nombres de las terminales del circuito como se ve en la Imagen 31 es a través de la tabla 13. Esta Define la posición de los nombres en las terminales del circuito como se muestra en la Imagen 33 a). La tabla 13 tiene información de las dimensiones de la figura columna 3 y 4 y la columna 5 el primer lado donde va a ir el grupo de nombres, en este caso la posición Norte indica la parte de arriba de la de la imagen del circuito. El primer nombre se ubica en la coordenada (41,0) pixeles y los siguientes están a un desplazamiento de 55 pixeles.

Tabla 14. PPinOut-Ubicación de nombre de terminales.

		Dimencion de figura en in		Posición Norte, Sur etc	Cantidad	Coordenadas		Offset en %	Nombres de terminales																			
Pin	Out	X	Y			X	Y																					
7400-1-TI		14	412	215	N	7	41	0	55	Vcc	4B	4A	4Y	3B	3A	3Y	S	7	41	215	55	1A	1B	1Y	2A	2B	2Y	Gnd
7401-1-TI		14	412	215	N	7	41	0	55	Vcc	4Y	4B	4A	3Y	3B	3A	S	7	41	215	55	1Y	1A	1B	2Y	2A	2B	Gnd
7402-1-TI		14	412	215	N	7	41	0	55	Vcc	4Y	4B	4A	3Y	3B	3A	S	7	41	215	55	1Y	1A	1B	2Y	2A	2B	Gnd
7403-1-TI		14	412	215	N	7	41	0	55	Vcc	4B	4A	4Y	3B	3A	3Y	S	7	41	215	55	1A	1B	1Y	2A	2B	2Y	Gnd
7404-1-TI		14	412	215	N	7	41	0	55	Vcc	6A	6Y	5A	5Y	4A	4Y	S	7	41	215	55	1A	1Y	2A	2Y	3A	3Y	Gnd
7405-1-TI		14	412	215	N	7	41	0	55	Vcc	6A	6Y	5A	5Y	4A	4Y	S	7	41	215	55	1A	1Y	2A	2Y	3A	3Y	Gnd
7406-1-TI		14	412	215	N	7	41	0	55	Vcc	6A	6Y	5A	5Y	4A	4Y	S	7	41	215	55	1A	1Y	2A	2Y	3A	3Y	Gnd
7407-1-TI		14	412	215	N	7	41	0	55	Vcc	6A	6Y	5A	5Y	4A	4Y	S	7	41	215	55	1A	1Y	2A	2Y	3A	3Y	Gnd

La Tabla 14 define la posición de los nombres en grupos, específicamente la información mostrada del componente 7400 tiene 2 grupos, Uno en la parte Norte (Arriba) “N” de 7 nombre y otro en la parte sur “S” (Abajo) también de 7 nombres, si el circuito tiene una forma física como la mostrada en la Imagen 33 b), entonces tendría cuatro grupos de nombres.

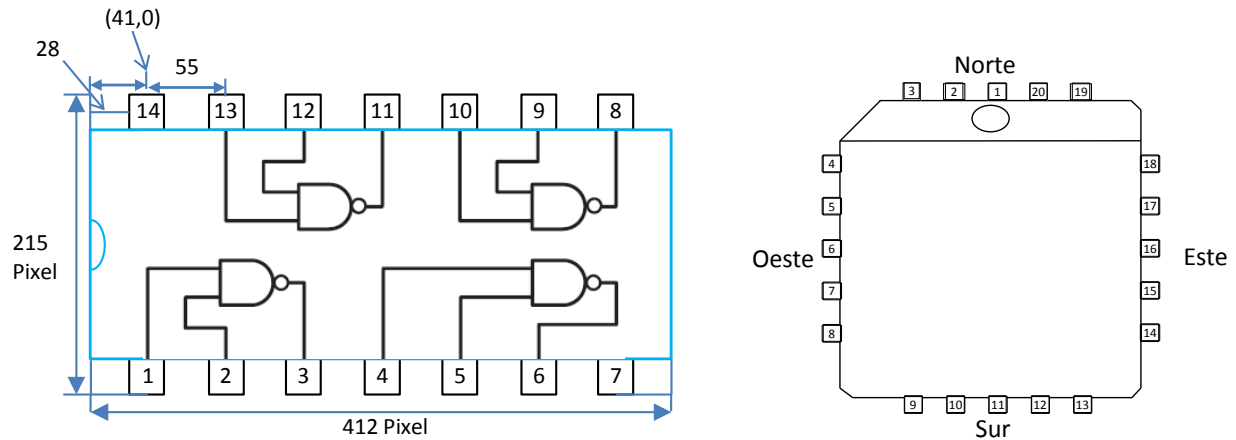


Imagen 33 Coordenadas de la posición de terminales, a) Empaquetado DIP, b) Empaquetado LCCC.

```

For PinNames = 1 To Grup_Total_Pins
    ColName = GrupNames + 4 + PinNames
    PinZIF40 = FindZIFPin(Vec_LocationPinNames(1, ColName))

    TempText = Vec_LocationPinNames(1, ColName)
    PinLabels(PinZIF40).Text = Vec_40VisualPinsNames(1, PinZIF40) 'TempText
    'TempText.Substring(0, 2)
    If Mid(TempText, 1, 2) = "NC" Then PinLabels(PinZIF40).Text = "NC"
    OrgX = PBox_WorkChip.Location.X + PinIniX
    OrgY = PBox_WorkChip.Location.Y + PinIniY
    Result_Labels(PinZIF40).Text = "-"

    Select Case Orientation
        Case "N"
            PinX = OrgX + Offset * (PinNames - 1) - PinLabels(PinZIF40).Width / 2
            PinY = OrgY - PinLabels(PinZIF40).Size.Height * 1
            LB_Result_X = PinIniX + Offset * (PinNames - 1) - Result_Labels(PinZIF40).Width / 2
            LB_Result_Y = PinIniY + Result_Labels(PinZIF40).Size.Height * 0
        Case "S"
            PinX = OrgX + Offset * (PinNames - 1) - PinLabels(PinZIF40).Width / 2
            PinY = OrgY + PinLabels(PinZIF40).Size.Height * 0
            LB_Result_X = PinIniX + Offset * (PinNames - 1) - Result_Labels(PinZIF40).Width / 2
            LB_Result_Y = PinIniY - Result_Labels(PinZIF40).Size.Height * 1
        Case "E"
            PinX = PBox_WorkChip.Location.X + PinIniX + PinLabels(PinZIF40).Size.Height / 2
            PinY = PBox_WorkChip.Location.Y + PinIniY + Offset * (PinNames - 1) - PinLabels(PinZIF40).Width / 2
        Case "O"
            PinX = PBox_WorkChip.Location.X + PinIniX - (PinLabels(PinZIF40).Size.Height / 2) - PinLabels(PinZIF40).Width / 2
            PinY = PBox_WorkChip.Location.Y + PinIniY + Offset * (PinNames - 1) - PinLabels(PinZIF40).Width / 2
    End Select
    PinLabels(PinZIF40).Location = New Point(PinX, PinY)
    PinLabels(PinZIF40).Visible = True
    ' Result_Labels(PinZIF40).ForeColor = Color.Green

    Result_Labels(PinZIF40).Location = New Point(LB_Result_X, LB_Result_Y)
    ' Result_Labels(PinZIF40).Visible = False
Next

```

Listado 6. Código para colocar el nombre en las terminales del circuito.

El listado 6 muestra la programación que coloca el nombre en las terminales del circuito como en la imagen 33, basado en la tabla 13. En la primera parte del listado se tiene la lectura del grupo de nombres de terminal y en qué lado se van a colocar, en la parte central del código se observa la sección donde identifica la ubicación Norte, Sur, Este y Oeste donde van a ser colocados los nombres, esta sección únicamente calcula la coordenada en píxeles y en la parte final del listado simplemente asigna la coordenada al objeto "Pinlabel" que contiene el nombre de la terminal del circuito. El listado 6 se ejecuta tantas veces como grupos de terminales aparecen en la Tabla 14.

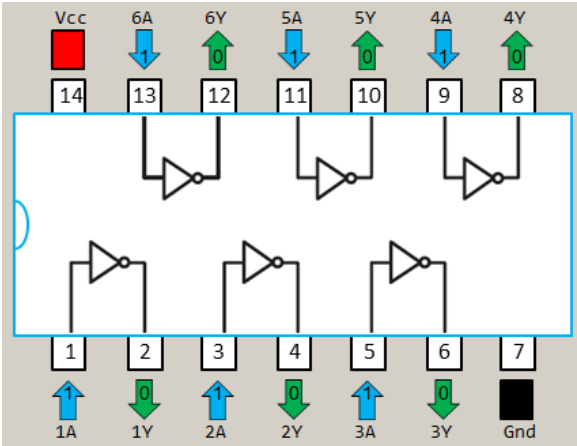


Imagen 34, Flechas indicadoras de entrada y salida en el circuito de prueba.

Colocar las flechas indicadoras del tipo y dirección de terminales es una parte crucial porque representa un aspecto didáctico del software, como se muestra en la imagen 34. Para llevar a cabo esta representación didáctica se combina varias tablas de datos como es la tabla 14, que tiene la información de cuáles terminales son entrada o salida y de qué tipo.

	Norte	Sur	Este	Oeste
Entradas				
	00E2EN	01E2ES	02S2EE	03S2EO
Salida 2 Estados 2E				
	04S2EN	05S2ES	06S2EE	07S2EO
Salida Colector abierto				
	08SCON	09SCOS	10SCOE	11SCOO
Salida 3 Estados 3E				
	12S3EN	13S3ES	14S3EE	15S3EO
Vcc				
	32Vcc	33Vcc	34Vcc	35Vcc
Gnd				
	36Gnd	37Gnd	38Gnd	39Gnd
NC				
	NC	NC	NC	NC

Imagen 35. Flechas indicadora del tipo y dirección de terminales.

Adicionalmente, requiere la información de la forma física de las flechas y la orientación que toma según las entradas y salidas, finalmente se integra el estatus lógico es decir el valor lógico en la entrada y lo que se obtiene en la salida en cada una de las flechas indicadoras.

Tabla 15. Tipo de entrada y salidas de cada terminal del circuito.

	1	2	3	4	5	6	7	8	.	.	.	.	33	34	35	36	37	38	39	40
7432-1-TI	E2E	E2E	S2E	E2E	E2E	S2E	Gnd							S2E	E2E	E2E	S2E	E2E	E2E	Vcc
7433-1-TI	E2E	E2E	SOC	E2E	E2E	SOC	Gnd							SOC	E2E	E2E	SOC	E2E	E2E	Vcc
7437-1-TI	SOC	E2E	E2E	SOC	E2E	E2E	Gnd							E2E	E2E	SOC	E2E	E2E	SOC	Vcc
7438-1-TI	E2E	E2E	NC	E2E	E2E	S2E	Gnd							S2E	E2E	E2E	NC	E2E	E2E	Vcc
7439-1-TI	S2E	S2E	S2E	S2E	S2E	S2E	S2E	Gnd						S2E	S2E	S2E	E2E	E2E	E2E	Vcc
7440-1-TI	S2E	S2E	S2E	S2E	S2E	S2E	S2E	Gnd						S2E	S2E	S2E	E2E	E2E	E2E	Vcc
7442-1-TI	S2E	S2E	S2E	S2E	S2E	S2E	S2E	Gnd						S2E	S2E	S2E	E2E	E2E	E2E	Vcc

El Listado 7, es la sección del programa que asigna la flecha indicadora de entrada y salida en cada terminal del circuito, está basado en la tabla 14 que define cuales son entradas y las salidas, previamente fueron definidos los objetos tipo imagen en programación que toman la figura de la flecha indicadora. En la instrucción "PBox_IOs_Arrow(PinZIF40).Image= . . ." en el Listado 7 se asigna la imagen de la flecha según el tipo de entrada y la orientación que tiene, por ejemplo Norte, Sur, Este, oeste.

```

Dim IOsTypes() As String = {"E2E", "S2E", "SOC", "S3E", "S3Z", "SB3", "SBZ", "EB3", "Vcc", "Gnd", "NC"}
Dim PinType As String 'Variable temporal del tipo de pin, Entrada, Salida etc
Dim IOsOrientation() As String = {"N", "S", "E", "O"}
.
.
.
For PinNames = 1 To Grup_Total_Pins
    ColName = GrupNames + 4 + PinNames
    PinZIF40 = FindZIFPin(Vec_LocationPinNames(1, ColName))

    For Row = 0 To 10
        Temp1 = Vec_IO_Type(1, PinZIF40)
        Temp2 = IOsTypes(Row)
        If Vec_IO_Type(1, PinZIF40) = IOsTypes(Row) Then Exit For
    Next
    For Col = 0 To 3
        If Orientation = IOsOrientation(Col) Then Exit For
    Next

    PBox_IOs_Arrow(PinZIF40).Image = ArrowImagens(Row * 4 + Col)
    ArrowX = 25
    ArrowY = 30
    PBox_IOs_Arrow(PinZIF40).Width = ArrowX
    PBox_IOs_Arrow(PinZIF40).Height = ArrowY
    PBox_IOs_Arrow(PinZIF40).SizeMode = PictureBoxSizeMode.StretchImage
    'PBox_IOs_Arrow(PinZIF40).SizeMode = PictureBoxSizeMode.AutoSize
    PBox_IOs_Arrow(PinZIF40).Visible = True

```

Listado 7. Asignación de flechas según entradas y salidas.

El código mostrado en el Listado 8 determina la posición de cada una de las flechas indicadoras de entrada o salida, en este caso también se define solamente las coordenadas de dónde va a quedar la flecha indicadora, porque el objeto que contiene la imagen de la flecha ya está definido, solo resta determinar su ubicación, el cálculo es basado en la Tabla 14 que tiene las dimensiones de la imagen del circuito en Pixel, con lo que se determina la posición de cada una de las terminales como se muestra en

la Imagen 33 a), el código del Listado 8, es semejante al código del Listado 6 en el que se determinaba la ubicación de los nombres de las terminales.

```
Dim IOsTypes() As String = {"E2E", "S2E", "SOC", "S3E", "S3Z", "SB3", "SBZ", "EB3", "Vcc", "Gnd", "NC"}
Dim PinType As String 'Variable temporal del tipo de pin, Entrada, Salida etc
Dim IOsOrientation() As String = {"N", "S", "E", "O"}

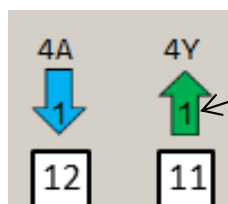
:
:
:

Select Case Orientation
Case "N"
    ' PinLabels(PinZIF40).Location = New Point(PinX, PinY)
    PinX = PBox_WorkChip.Location.X + PinIniX + Offset * (PinNames - 1) - ArrowX * 0.5
    PinY = PBox_WorkChip.Location.Y + PinIniY - ArrowY * 1.2
    LB_IO_Status(PinZIF40).Location = New Point(2 + (ArrowX - StatusX) / 2, ArrowY - StatusY - 2)
    PinLabels(PinZIF40).Location = New Point(PinX, PinY - PinLabels(PinZIF40).Height)
Case "S"
    PinX = PBox_WorkChip.Location.X + PinIniX + Offset * (PinNames - 1) - PBox_IOs_Arrow(PinZIF40).Width / 2
    PinY = PBox_WorkChip.Location.Y + PinIniY + PBox_IOs_Arrow(PinZIF40).Height * 0.2
    LB_IO_Status(PinZIF40).Location = New Point(2 + (ArrowX - StatusX) / 2, 2)
    PinLabels(PinZIF40).Location = New Point(PinX, PinY + ArrowY)
Case "E"
    PinX = PBox_WorkChip.Location.X + PinIniX + PBox_IOs_Arrow(PinZIF40).Width * 0.1
    PinY = PBox_WorkChip.Location.Y + PinIniY + Offset * (PinNames - 1) - PBox_IOs_Arrow(PinZIF40).Height / 2
    LB_IO_Status(PinZIF40).Location = New Point(6, 10)
Case "O"
    PinX = PBox_WorkChip.Location.X + PinIniX - PBox_IOs_Arrow(PinZIF40).Width - PBox_IOs_Arrow(PinZIF40).Height * 0.5
    PinY = PBox_WorkChip.Location.Y + PinIniY + Offset * (PinNames - 1) - PBox_IOs_Arrow(PinZIF40).Height / 2
End Select
PBox_IOs_Arrow(PinZIF40).Location = New Point(PinX, PinY)
' PBox_IOs_Arrow(PinZIF40).Visible = True

Next
GrupNames = GrupNames + Vec_LocationPinNames(1, GrupNames + 1) + 5
Orientation = Vec_LocationPinNames(1, GrupNames)
```

Listado 8, Código de ubicación de flechas indicadoras de entrada o salida.

En la imagen del circuito de prueba mostrado en el área de trabajo del software, se tiene un valor lógico 0 ó 1 en cada flecha indicadora de entrada o salida, como se muestra en la Imagen 36, el Listado 9, es un procedimiento de asignación del valor lógico en las terminales del circuito para efectos de visualización. Si la terminal es una entrada, el valor lógico es el que se está enviando al módulo probador de circuitos, en el caso de salida el valor lógico es leído como respuesta del circuito físico en el módulo probador.



Valor lógico de las
entras y salidas

Imagen 36. Valor lógico de las terminales del circuito.

```
Sub Display_IOs()
For X = 1 To 40
    'Despliega el valor de las entradas y salidas en el circuito en operacion
    If IOs_OutPuts(X) = 1 Then LB_IO_Status(X).Text = Mid(Str(IO_Out_Value(X)), 2, 1) ' si es salida
    If IOs_Inputs(X) = 1 Then LB_IO_Status(X).Text = Mid(Str(IO_In_Value(X)), 2, 1) ' si es entrada
Next
End Sub
```

Listado 9 Asignación del valor lógico en cada terminal del circuito.

Áreas de información complementarias de la interfaz gráfica

En el lado Superior derecho de la interfaz gráfica mostrada en la Imagen 27, se tiene un espacio de texto donde se muestra información del circuito seleccionado para prueba, se dispone de un enlace para verificar información en la hoja de datos, muestra el tipo de circuito, la función lógica, el fabricante, el empaquetado y el número de terminales, la información es tomada de la Tabla 11. Esta área es de información como apoyo del circuito que se está probando. La Imagen 37 a) es una vista ampliada del área de información.

Otra área de información en la interfaz gráfica parte inferior derecha es la localización, esta se refiere a cómo va colocado el componente en el probador para poder realizar el proceso de prueba. Los circuitos integrados tienen una muesca que señala la terminal 1 y de acuerdo a la Imagen 37 b) el componente siempre va colocado en la posición señalada en la base tipo ZIF, el nombre de la imagen que define la posición del circuito es señalada en la Tabla 12, columna 3.

El área de información complementaria señalada como descripción, muestra la descripción que el fabricante le da al componente indicando la función lógica, esta información se encuentra en la hoja de datos del componente y en la interfaz gráfica se localiza en la parte superior del área de trabajo de prueba del circuito. En la Imagen 37 se tiene una vista ampliada de esta área de información

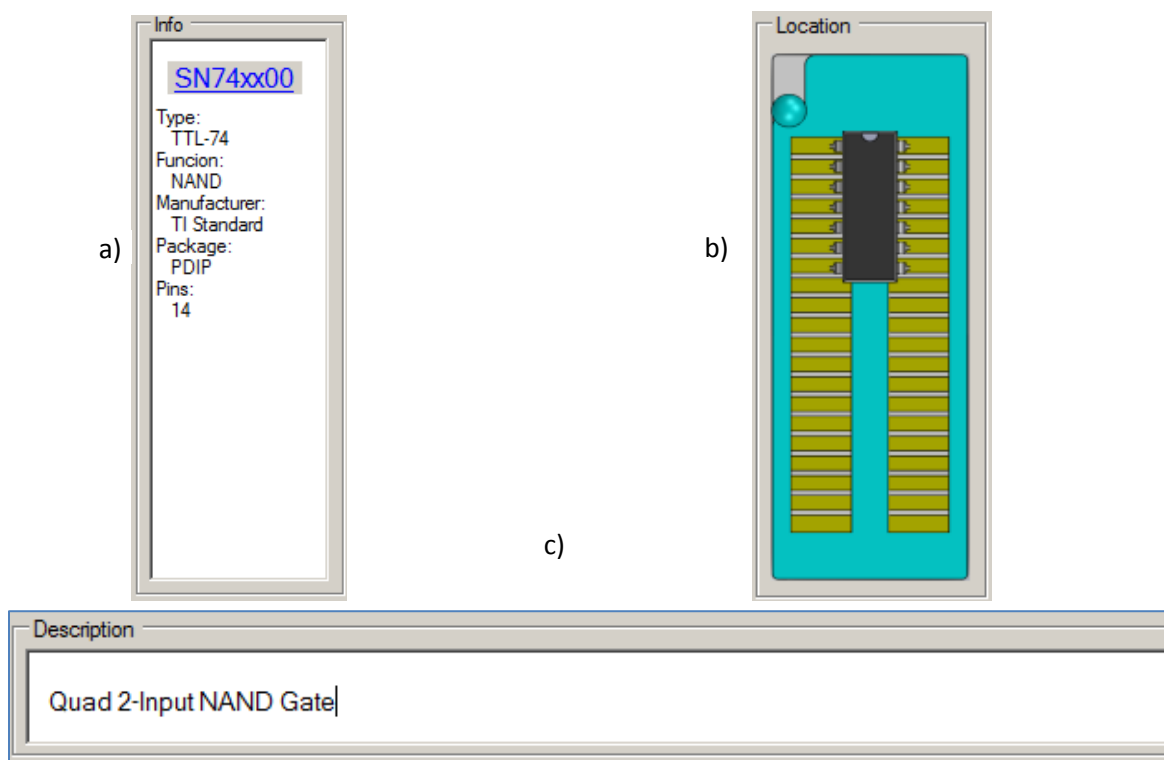


Imagen 37 Áreas complementarias. a) Vista ampliada del área de información.

b) Ubicación del circuito en la base ZIF para realizar la prueba.

c) Descripción de la función lógica del circuito.

-Programación de algoritmos de Pruebas combinacionales

En la Tabla 1 se tiene la clasificación de las funciones u operaciones lógicas de los circuitos digitales , en esa tabla se muestran 16 operaciones lógicas, lo cual indica que es un número limitado de operaciones diferentes, aun en el caso de que se hayan omitido algunas funciones. Esta delimitación favorece la programación de los algoritmos de pruebas porque también es un número limitado de algoritmos.

La ejecución de las pruebas tanto de circuitos combinacionales como secuenciales se basan en la tabla de pruebas, (Tabla 16), y se ejecutan bajo los siguientes pasos.

1. Leer el código de la prueba y sus parámetros.
2. Leer cuáles son las entradas y las salidas del circuito que le corresponden a ese tipo de prueba. según el número de circuito y la Tabla 13 de Nombres de terminales.
3. Aplicar las combinaciones posibles de unos y ceros según el número de entradas.
4. Comparar el resultado de salida con el resultado correspondiente a la función u operación lógica indicada en la prueba.
5. Marcar en el diagrama presentando en la pantalla de pruebas del software, si las terminales pasaron la prueba o tienen una falla.
6. Continuar con la siguiente prueba de la tabla.

Tabla 16, Condiciones iniciales y Secuencia de pruebas.

Circuito General	Prueba	Pins 40-32	Pins 32-25	Pins 24-17	Pins 16-9	Pins 8-1	Prueba	Tipo	Salida	Gate	Gate	Gate	Gate	Gate	Prueba	Tipo
7400-1-TI	CI	11111111	00000000	00000000	00000000	11111111	CD	NAND	0	1	2	3	4			
7401-1-TI	CI	11111111	00000000	00000000	00000000	11111111	CD	NAND	0	1	2	3	4			
7402-1-TI	CI	11111111	00000000	00000000	00000000	11111111	CD	NOR	0	1	2	3	4			
7403-1-TI	CI	11111111	00000000	00000000	00000000	11111111	CD	NAND	0	1	2	3	4			
7404-1-TI	CI	11111111	00000000	00000000	00000000	11111111	CD	NOT	0	1	2	3	4	5	6	
7405-1-TI	CI	11111111	00000000	00000000	00000000	11111111	CD	NOT	0	1	2	3	4	5	6	

En la Tabla 16 muestra que para cada circuito se tiene las condiciones de iniciales (CI) de las entradas (2ª columna), en la tabla se tienen los valores lógicos de las 40 terminales de la base ZIF, pero solo se aplican el valor binario a las entradas definida en la Tabla 15.

```

While Test <> ""
  Select Case Test
    Case "CI" 'CI=Condiciones iniciales
      Read_Initial_Condition(IndexTest)
      CMD_Write_All_Inputs()
      DataRX_to_IO_Pins()
      Display_IOs()
      IndexTest = IndexTest + 6
    Case "CD" 'CD=Compuerta digital AND,NAND,OR,NOR,BUFFER,NOT,XOR,NXOR
      GateType = Vec_Test_List(1, IndexTest + 1)
      Gate = 1

```

Listado 10 Condiciones iniciales de las entradas del circuito.

En el Listado 10 se muestra la sección del programa que envía al circuito probador el valor lógico de las entradas como condición inicial de las pruebas en el circuito digital.

Continuando con la Tabla 16, en la columna 8, se tiene el código de “CD” que significa “Prueba de compuerta digital”, y como parámetros se tiene: El tipo de la compuerta (AND, NAND, Or, Etc.), si la salida es común o individual de cada compuerta y por último el número de la compuerta donde se va a ejecutar la prueba.

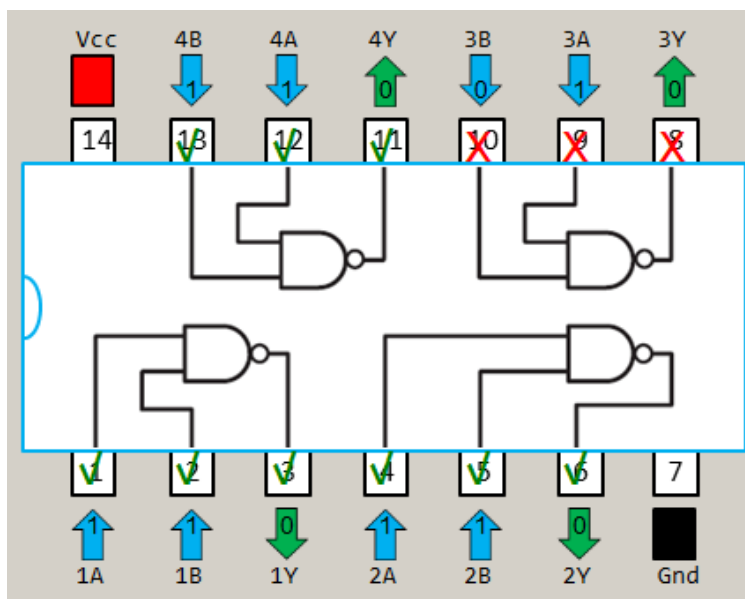


Imagen 38 Circuito 7400, Cuatro compuertas NAND.

Para el caso del circuito de la Imagen 38 el algoritmo busca las entradas de cada compuerta, por ejemplo, en la compuerta uno, el algoritmo busca todas las terminales de entrada numeradas uno, 1A, 1B, etc.(Excepto 1Y) y busca como salida únicamente la terminal 1Y, de esa manera, la compuerta puede tener dos o más entradas.

Existen algunos circuitos digitales como los CMOS familia 4000, que tienen diferentes compuertas en un mismo componente, de tal manera que la secuencia de pruebas se vería como en la Tabla 17, lo cual significa que se realizan pruebas en de compuerta 1 y 2 tipo NAND y en la compuerta 3 Tipo NOR.

Tabla 17 pruebas en circuito con diferentes compuertas en el mismo componente.

Prueba	Pins 40-32	Pins 32-25	Pins 24-17	Pins 16-9	Pins 8-1	Prueba	Tipo	Salida	Gate	Gate	Tipo	Prueba	Tipo	Salida
CI	11111111	00000000	00000000	00000000	11111111	CD	NAND	0	1	2	CD	NOR	0	3

El Listado 11, es la sección del programa que aplica las combinaciones de entrada en cada compuerta y de acuerdo a la compuerta que se está probando determina qué valor se debe obtener. En la parte inferior del listado se identifica las terminales que están funcionando correctamente o incorrectamente porque no cumplen con la lógica de la compuerta probada.

```

For i = 0 To Iterations
    DataTX(2) = i
    CMD_Write_Working_Test_Inputs() 'responde con resultado de operación lógica
    Receive_DataRX()
    CalculateGateValue = 0
    Select Case GateType
        Case "AND"
            If i = Iterations Then CalculateGateValue = 1
        Case "NAND"
            If i <> Iterations Then CalculateGateValue = 1
        Case "OR"
            If i <> 0 Then CalculateGateValue = 1
        Case "NOR"
            If i = 0 Then CalculateGateValue = 1
        Case "BUFFER"
            CalculateGateValue = i
        Case "NOT"
            CalculateGateValue = i Xor 1
        Case "XOR"
            If (i = 1) Or i = 2 Then CalculateGateValue = 1
        Case "NXOR"
            If (i = 0) Or i = 3 Then CalculateGateValue = 1
    End Select
    If CalculateGateValue = DataRX(2) Then
        Circuit_Operation("Working")
    Else
        Circuit_Operation("Fault")
        Bit = 1
        For X = 1 To Working_Inputs(0)
            IO_In_Value(Working_Inputs(X)) = 0
            If i And Bit Then IO_In_Value(Working_Inputs(X)) = 1
            Bit = Bit * 2
        Next
        IO_Out_Value(Working_Outputs(1)) = DataRX(2)
        Display_IOS()
        'mostrar la combinacion que tiene la falla
        Exit For
    End If
Next

```

Listado 11. Prueba de tipo compuerta digital AND,NAND,OR, Etc.

-Programación de Algoritmos de Pruebas Secuenciales

La programación de los algoritmos de pruebas secuenciales es semejante a los algoritmos de pruebas combinatoriales, se basa en los mismos pasos mencionados al inicio del tema de programación de algoritmos combinatoriales, con la diferencia de que los resultados obtenidos de un circuito secuencial dependen del orden o secuencia de las señales de entrada. Para mostrar este proceso se va a ejemplificar con el circuito 7474 que es un doble Flip-Flop tipo D como el que se muestra en la Imagen 39. Todos los circuitos con FlipFlop tipo D tienen como mínimo la entrada D, el Reloj (CLK) y la salida Q. la prueba de la entrada D es sistemática, se aplica un valor lógico en D y un pulso a la entrada CLK, y como resultado el dato en D aparece en Q. El algoritmo busca las entradas y salidas del flipflop, en este caso las entradas incluyen Reset (RST) y Borrado (CLR), así que, al detectar esas entradas se hace prueba en estas, el borrado de la salida del Flip-Flop se obtiene haciendo un pulso a cero en la entrada CLR, pero es importante que el valor previo en Q sea de 1 lógico. De otra manera no se detectaría el valor cero generado por CLR. La entrada RST genera un 1 en Q, pero es necesario que Q valga 0 para poder notar la operación de RST.

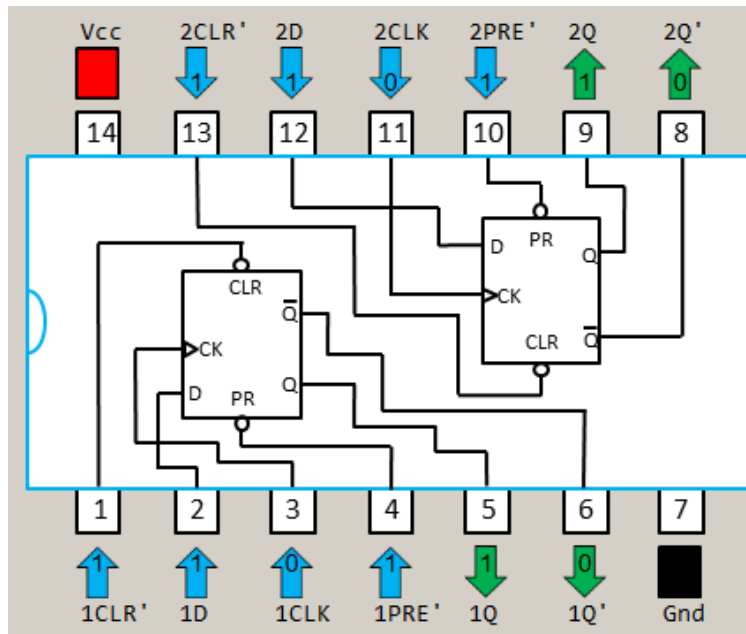


Imagen 39 Circuito TTL Flip-Flop 7474 con Reset y Clear.

El Listado 12 muestra la sección del programa que realiza las pruebas de un Flip-Flop tipo “D”, primero se detecta que la prueba es para un Flip-flop y además de que tipo es, como se mostradas en la Tabla 18. En este caso el Listado 12 ejemplifica al tipo “D” que corresponde a un circuito como el mostrado en la Imagen 39, la primera parte busca cuáles son las terminales de entrada, terminal D, reloj CLK, Reset RST y borrado CLR. Si alguna de esas terminales no existen en el circuito, no se toman en cuenta al realizar la prueba. Como salidas va a buscar Q y Q' en algunos circuitos tampoco existe Q' por lo tanto no se toman en cuenta en la prueba. Continuando con el Listado 12 hay un ciclo donde se aplican los dos valores 0 y 1 a la entrada D con un pulso en el reloj CLK, se compara Q y Q' con la lógica que debe dar como resultado y se decide si la entradas D y la salida Q están funcionando correctamente.

Tabla 18 Codificación de prueba de un Flip-Flop tipo D.

PinOut	Prueba	Pins 40-32	Pins 32-25	Pins 24-17	Pins 16-9	Pins 8-1	Prueba	Tipo	FF	FF
7474-1-TI	CI	11101111	00000000	00000000	00000000	11111011	FF	D	1	2

Finalmente, se realiza la prueba del borrado aplicando un pulso a 0 en la entrada CLR y verifica si la salida Q vale cero entonces señala que esa terminal está correcta, de igual manera sucede con la entrada Reset RST pero en este caso la salida Q debe ser un 1. Si el circuito tiene más elementos la prueba se repite para el siguiente FLIP-FLOP como lo muestra la tabla 17 en el extremo derecho, marcado que el circuito 7474 tiene 2 FLIP-FLOP.

```

Case "FF" 'Flip-Flpo tipo RS,JK,D,T
  FFType = Vec_Test_List(1, IndexTest + 1)
  Select Case FFType 'Tipo RS,JK,D
    Case "D" 'Flip-Flpo D
      Flip_Flop = 1
      Do
        FFNumber = Vec_Test_List(1, IndexTest + 2 + Flip_Flop)
        Find_FF_Inputs(FFNumber)
        InputsOnFF = DataTX(1)
        CMD_Working_Test_Inputs(InputsOnFF)
        Find_FF_Output(FFNumber)
        OutputsOnFF = DataTX(1)
        CMD_Working_Test_Outputs(OutputsOnFF)
        Iterations = (2 ^ 1) - 1
        For i = 0 To Iterations
          DataTX(2) = i
          CMD_Write_FF_Test_Inputs()
          CMD_FF_Pulse("CLK")
          If Working_Inputs(1) Then Q = DataRX(2)
          If Q = i Then
            FF_Circuit_Operation("Working", "Q")
          Else
            FF_Circuit_Operation("Fault", "Q")
          End If
          If Working_Inputs(1) Then Not_Q = DataRX(3)
          If Not_Q = i Then
            FF_Circuit_Operation("Working", "Not_Q")
          Else
            FF_Circuit_Operation("Fault", "Not_Q")
          End If
        Next
        CMD_FF_Pulse("CLR")
        Q = DataRX(2)
        If Q = 0 Then
          FF_Circuit_Operation("Working", "CLR")
        Else
          FF_Circuit_Operation("Fault", "CLR")
        End If
        CMD_FF_Pulse("RST")
        Q = DataRX(2)
        If Q = 1 Then
          FF_Circuit_Operation("Working", "RST")
        Else
          FF_Circuit_Operation("Fault", "RST")
        End If
        Flip_Flop = Flip_Flop + 1
      Loop While IsNumeric(Vec_Test_List(1, IndexTest + 1 + Flip_Flop))
      IndexTest = IndexTest + 1 + Flip_Flop
    Case "JK" 'Flip-Flpo tipo JK

```

Listado 12 Prueba de circuito combinacional tipo FLIP-FLOP D.

El resto de la programación de los algoritmos es semejante, pero es importante señalar que la estructura a través de las tablas y de algoritmos específicos a la función del circuito, permite que el software pueda ser actualizado fácilmente agregando librerías en las tablas y programando funciones lógicas que no se hayan considerado.

-Fabricación del circuito impreso PCB-ZIF

En la Imagen 40 a) se muestra el impreso en software de diseño de placas PCB, este diseño es el que se había presentado en la Sección del Desarrollo de la Placa de Acoplo y en la Imagen 40 b) se tiene la placa enviada a una empresa de fabricación de placas de circuitos PCB de manera profesional, la calidad de dicha placa PCB es con un acabado excelente en color verde y totalmente funcional, en esta, se monta la base ZIF para el probador de circuitos digitales.

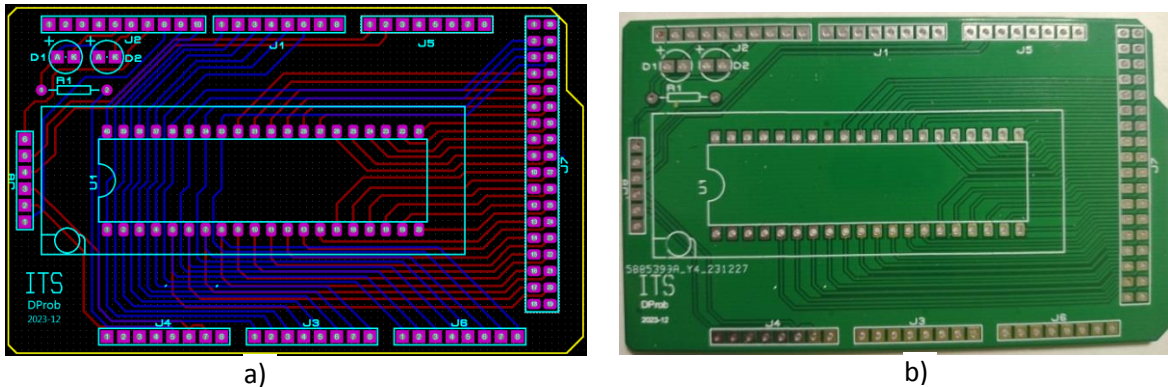


Imagen 40. Trazado de impreso de placa PCB-ZIF. a) Diseño en Software, b) Fabricación profesional de placas PCB-ZIF.

La imagen 33 muestra un lote de placas, dado que la fabricación de circuitos impresos PCB por parte de empresas dedicadas a ese ramo lo hacen por lotes de 5, 10, 15, etc. piezas, por cuestión de costo.



Imagen 41. Fabricación en lotes de 5, 10, 15, etc. Piezas.

Ensamble del circuito PCB-ZIF

La Imagen 42 corresponde al diagrama esquemático de la placa y la base ZIF, que también se mostró en la imagen 17, es el diseño de la placa de circuito PCB enviada a fabricar, este diseño es muy simple, ocupa muy pocos componentes porque ese es el propósito de este circuito probador, que sea de bajo costo. Requiere solamente de conectores tipo poste, la base ZIF, dos LEDs y una resistencia.

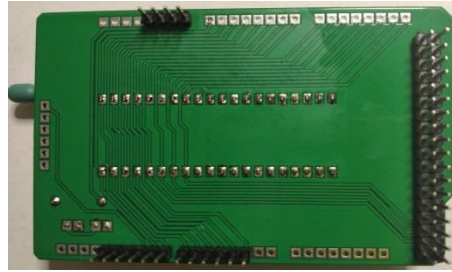


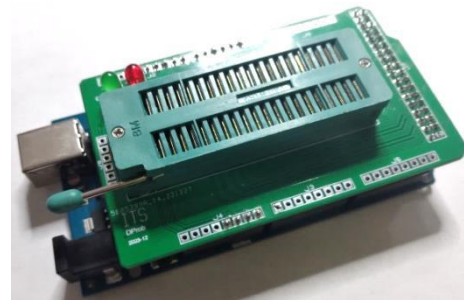
Imagen 44. Vista inferior de placa PCB-ZIF ensamblada.

Montaje del probador de circuitos

El montaje de la placa PCB con la base ZIF en el módulo del arduino Mega 2560 es realmente sencillo, solo se coloca en los conectores del módulo Arduino Mega y se conforma el probador de circuitos digitales, solo son dos piezas como se muestra en la Imagen 45. La imagen a) es el módulo de Arduino Megas 260 y la imagen b) es la placa montada directamente sobre el módulo al Mega 2560, con eso se obtiene completamente lo que es el probador de circuitos digitales.



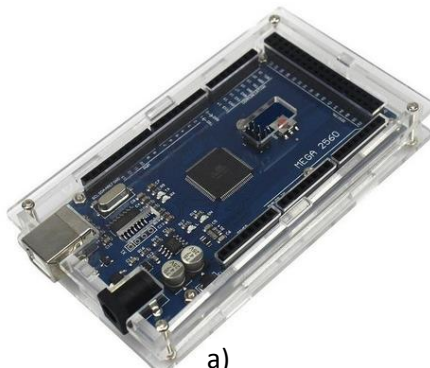
a)



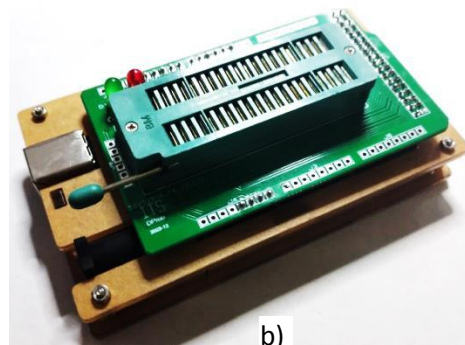
b)

Imagen 45. Montaje probador de circuitos. a) Arduino Mega 2560, b) Placa PCB-ZIF montada en Arduino Mega 2560.

Una forma más estética y manejable pero económica del probador de circuitos digitales es montar el módulo de 2560 en una carcasa como en la Imagen 46 a), que nos permitan darle mayor maniobrabilidad. Simplemente se coloca la placa PCB con la base ZIF y así se tiene un probador de circuitos digitales más manejable como se muestra en la Imagen 46 b).



a)



b)

Imagen 46. Montaje Práctico-Estético, a) Arduino mega 2560 con carcasa, b) Montaje práctico.

e) Pruebas y resultados

-Prueba manual de circuitos

Para realizar la prueba manual de un componente se ha seleccionado en el menú del software, el circuito TTL SN74LS00N como se muestra en la Imagen 47, si no aparece el número de parte exacto, entonces se puede seleccionar el número estándar 7400.

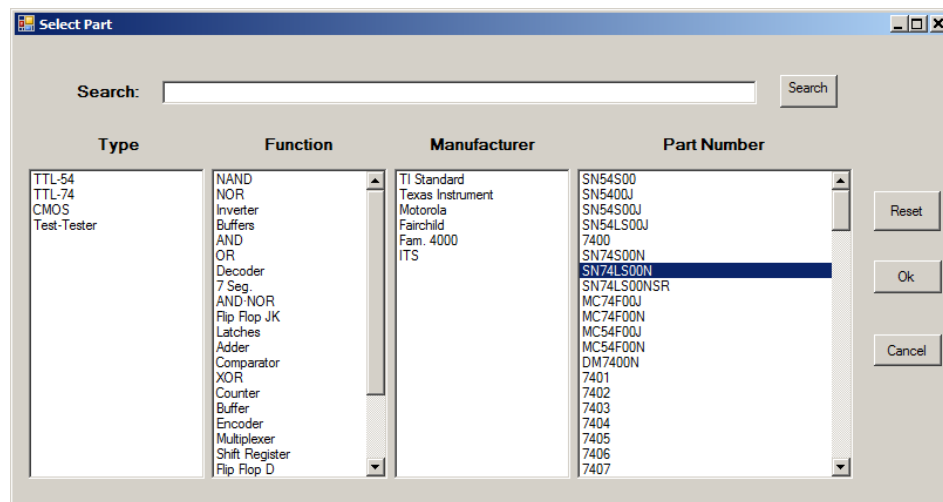


Imagen 47. Menú de selección del componente a probar.

La Imagen 48 muestra la pantalla después de haber seleccionado el circuito a probar, en esta imagen, se presenta: El diagrama de distribución de terminales, la descripción, la información adicional y en la parte baja derecha se muestra como se debe colocar el circuito en el probador, y en Imagen 49 se muestra físicamente la ubicación.

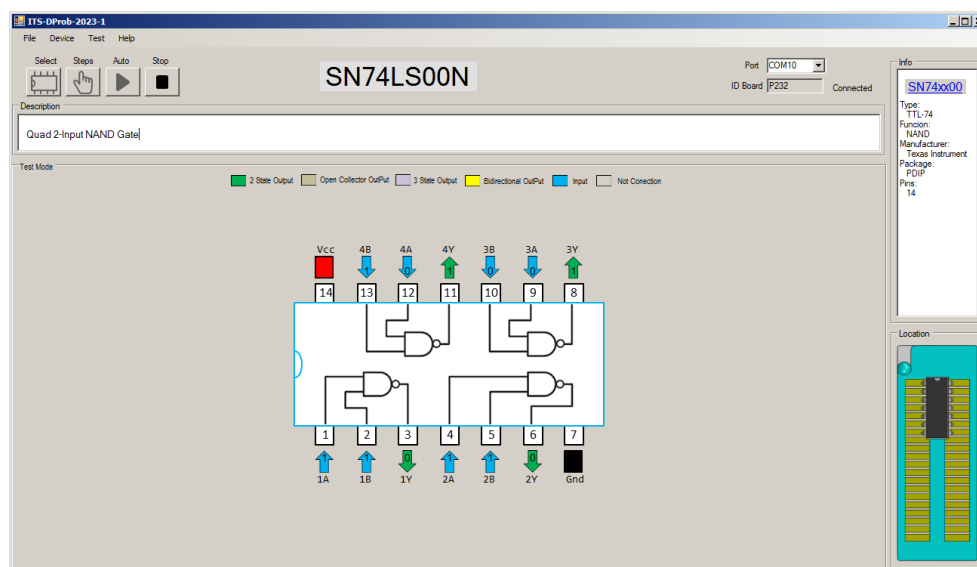


Imagen 48. Pantalla de prueba de circuito.



Imagen 49, componente de prueba SN74LS00N.

Dado que al seleccionar el circuito se aplican como valores iniciales 1's en cada entrada del circuito, se procedió a aplicar 0's de entrada en las terminales 9,10 y 12. Y en la Imagen 50 se observa claramente el resultado que da en las terminales de salida 8 y 11. Para una compuerta NAND con entradas 1,1 lógico, su salida es 0 lógico, en el caso de la terminal 11 la salida es un 0, porque la combinación de entrada es 1,0. En el caso de la terminal 8, la salida también vale 0, por la combinación de entrada es 0,0, al realizar todas las combinaciones posibles en cada compuerta NAND, se comprueba manualmente que el circuito 7400 está funcionando correctamente.

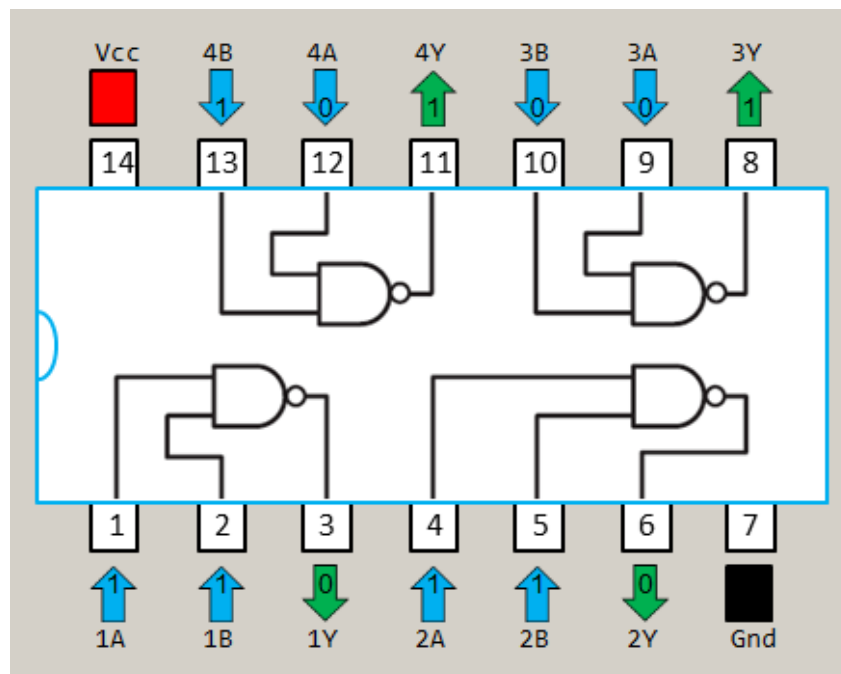


Imagen 50. Vista ampliada de Entradas y salidas del circuito mostrado en la Imagen 48.

-Prueba Automática de circuitos

La prueba automática se realizó al mismo circuito 7400 tomando la opción del menú “Auto”, la Imagen 51 muestra el resultado de la operación indicando el estatus o condición de funcionamiento de las terminales del circuito.

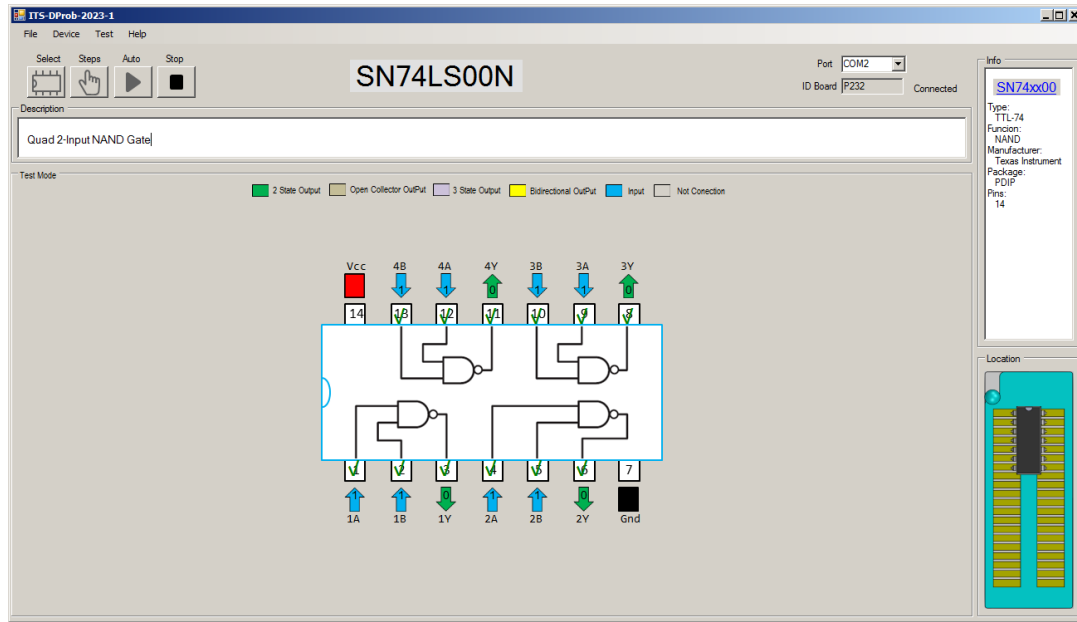


Imagen 51, Resultado prueba automática de circuito 7400.

En la Imagen 52 se tiene una vista ampliada del resultado de la prueba automática, se puede apreciar que cada terminal que pasa la prueba es palomeada con un ángulo en verde. Las terminales de Voltaje-Vcc y tierra-Gnd no están sometidas a prueba.

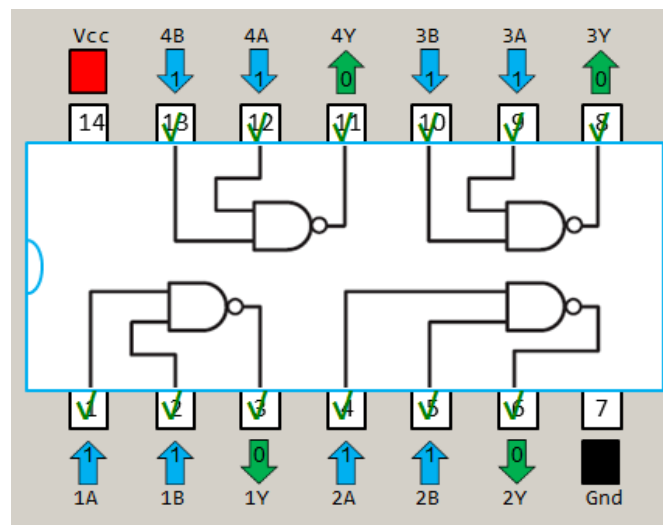


Imagen 52, Vista detallada del resultado de la prueba Automática.

Para evaluar el funcionamiento del probador de circuitos se ha desconectado una terminal del circuito SN74LS00N intencionalmente, como se muestra en la Imagen 53 y se somete a una prueba automática.

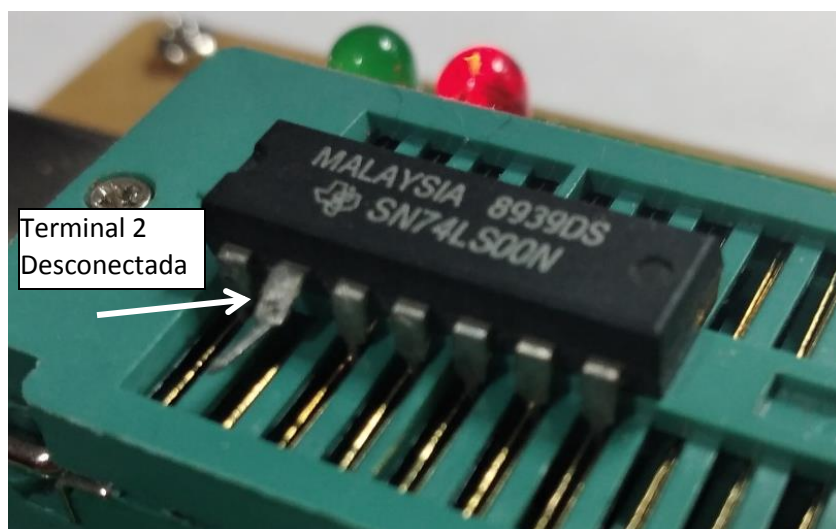


Imagen 53, Desconexión intencional de la terminal 2 del circuito SN74LS00N.

La Imagen 54 muestra el resultado de la prueba del circuito 74LS00 con la terminal desconectada de tal manera que se espera que se produzca una falla, como la prueba se hace por compuertas, en la imagen se muestra tachado en rojo las terminales de la compuerta que no cumplen con la lógica de una compuerta NAND, además se puede apreciar los valores de entrada en las terminales de entradas 1A y 1B que no están cumpliendo con la función lógica NAND, El resultado que se obtiene en la salida 1Y es un 0 lógico, pero el valor correcto debe ser un 1 lógico.

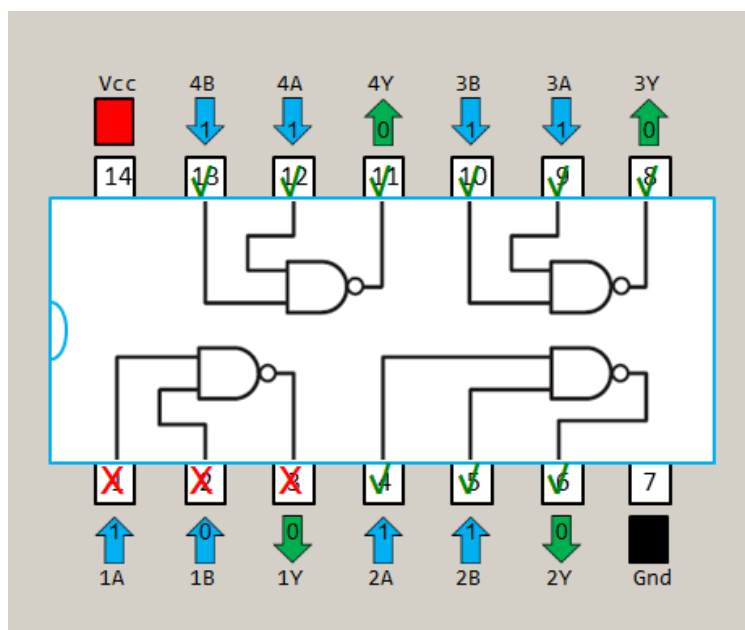


Imagen 54 Resultado de prueba automática con falla intencional.

-Prueba al Probador de circuitos digitales

Para evaluar las terminales del módulo de arduino 2560 que eléctricamente estén funcionando de manera correcta, se ha diseñado una prueba a través de emular un componente que solo tiene puentes entre una terminal y otra, como se muestran en la Imagen 55 a) y b), en este caso se tomó una base de circuito integrado de 40 terminales y se colocan puentes entre la parte superior e inferior.

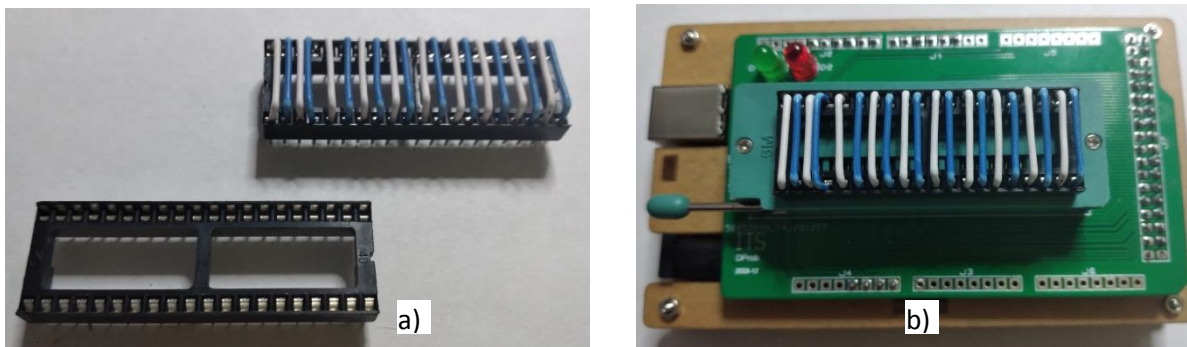


Imagen 55, Evaluación del módulo Arduino 2560, a) Componente Emulador, b) Montaje como componente de 40 terminales.

En el menú de Selección del software se han creado dos componentes nombrados Tester A-B y Tester B-A como se muestra en la Imagen 56, los cuales abren la pantalla de prueba como se muestran en la Imagen 57 e Imagen 58. La prueba se realiza manualmente conmutando los valores lógicos en las entradas y se debe obtener el mismo valor en la salida de cada puente, la evaluación se hace en los dos sentidos.

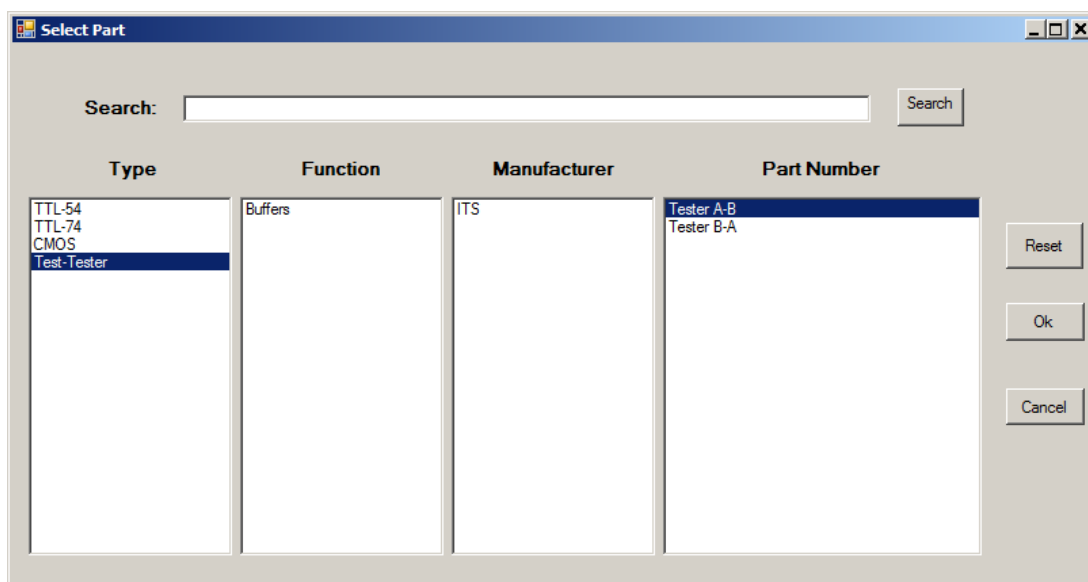


Imagen 56, Circuito de prueba del Módulo de arduino 2560.

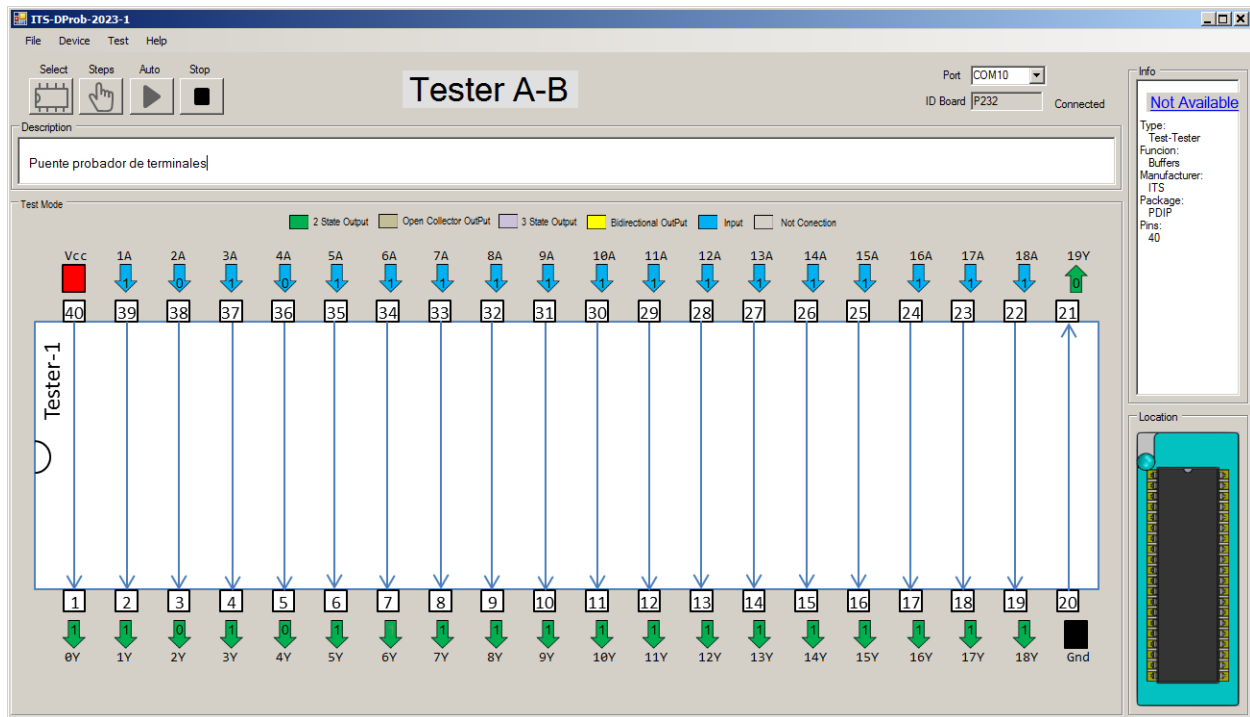


Imagen 57. Prueba de terminales del Arduino Mega 2560 de arriba hacia abajo.

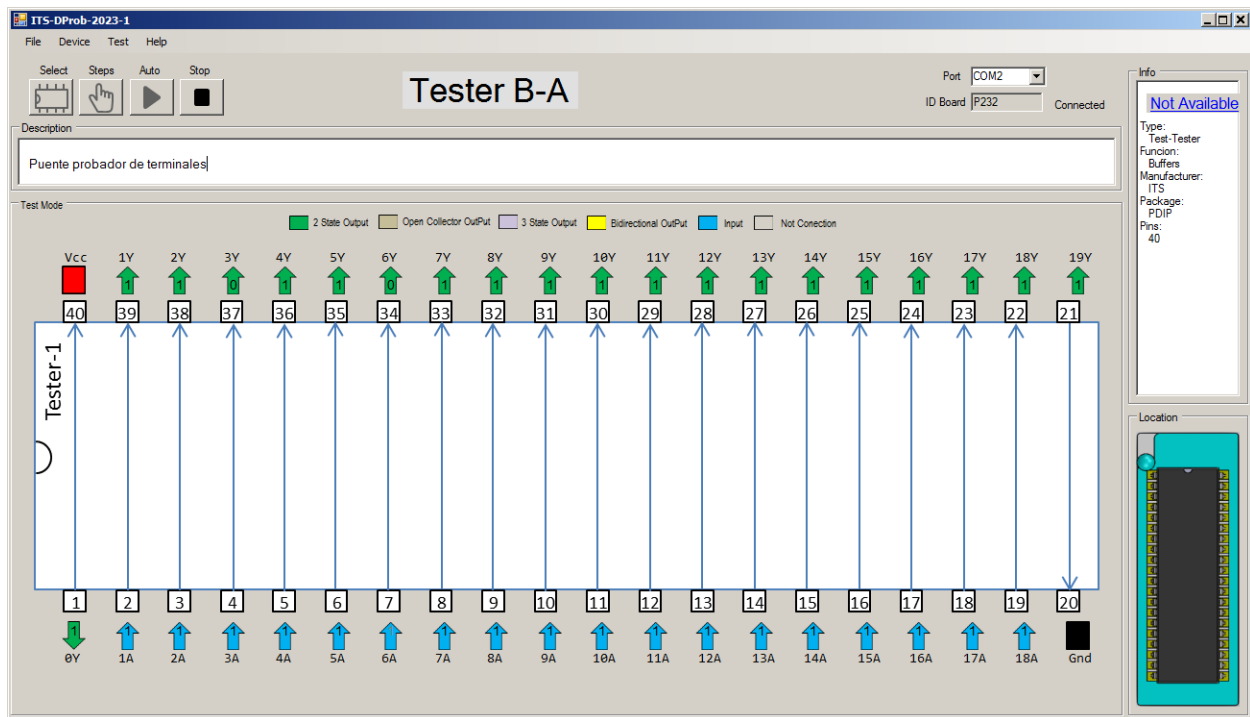


Imagen 58. Prueba de terminales del Arduino Mega 2560 de abajo hacia arriba.

f) Conclusiones y Observaciones

En el desarrollo del presente proyecto titulado “Probador de Circuitos Digitales TTL y CMOS de Bajo Costo” se tiene las siguientes conclusiones y observaciones.

-Conclusiones.

- Contar con un equipo de prueba de circuitos, portátil y de uso inmediato al realizar prácticas de electrónica definitivamente evita el retraso por fallas provocadas por componentes dañados.
- La interfaz gráfica orientada a propósito didáctico, permite dos tipos de prueba, manual y automática. En la prueba manual, el alumno maneja las señales de entrada y observa los resultados, con esto fortalece el conocimiento y manejo de los circuitos digitales. En la prueba Automática la interfaz señala claramente la parte o partes dañadas del componente, permitiendo determinar si el componente está en condiciones de uso.
- El desarrollo de equipo didáctico que se ajusta a necesidades del manejo de circuitos dan un aporte en el aprendizaje para los alumnos, como es el caso en el presente proyecto, Probador de circuitos Digitales con interfaz gráfica el cual presenta una característica Grafica-Didáctica que no se tiene en los equipos similares.

-Observaciones

La diversidad de componentes digitales que existen en el mercado es muy grande y aunque la librería de componentes incluida en esta primera versión del proyecto es suficiente para cubrir las necesidades de las materias con temas de electrónica digital, se sugiere actualizar dicha librería para evitar obsolescencia del proyecto.

Las primeras versiones del software como en este caso del Probador de Circuitos Digitales, es frecuente que requieren optimización para reducir el tiempo de ejecución en computadoras que no sean de gran capacidad, también es necesario compactar la base de datos para reducir el espacio requerido en memoria y poder tener un funcionamiento más eficiente al momento de hacer uso del probador.

g) Bibliografía

- [1] GWINSTEK. (2017), GuT-6000B Digital IC Tester, User manual, Good Will Instrument Co.
- [2] Xeltek, (2022), SUPERPRO 6100 Universal Programmers, Xeltek Tech-Support, Xeltek Inc.
- [3] Texas Instrument, 2007 Digital Logic data book , Texas Instruments Incorporated.
- [4] Motorola, (1991) CMOS Logic Data Book , Motorola Inc.
- [5] Fairchild Semiconductor (2000) MD7400 DataSheet. Fairchild Semiconductor Corp.
- [6] Microchip, (2014), ATmega640/1280/1281/2560/2561 datasheet, Microchip Technology Incorporated
- [7] EasyEDA, (2020), EasyEDA Tutorial V6.3.5.3, EasyEDA, China
- [8] Ndjountche T. (2020), Electrónica Digital 1, Circuitos lógicos combinatorios, ISTE Internacional, Reino Unido.
- [9] Ndjountche T. (2020), Electrónica Digital 2, Circuitos lógicos secuenciales y aritméticos, ISTE Internacional, Reino Unido.
- [10] D'Andrea, E. (2020). Programación Visual Basic.NET: Edición 2019. (n.p.): Amazon Digital Services LLC - Kdp.
- [11] Perez, M. (2014). Programación Orientada a Objetos Y Programación Estructurada. (n.p.): CreateSpace Independent Publishing Platform.

2. Objetivo del proyecto

-Objetivo General

Diseñar y construir un probador lógico de circuitos digitales TTL y CMOS con interface gráfica conectado a una computadora, con el propósito de eliminar el retraso en el desarrollo de experimentos electrónicos con circuitos digitales por causa de componentes dañados, el cual es sumamente útil en la materia de Electrónica Digital de la carrera de Ing. Mecatrónica.

-Objetivos específicos:

1. Diseñar un equipo de prueba lógica de componentes electrónicos digitales, que cubra las necesidades de prueba de componentes en la materia Electrónica digital de la carrera de Ing. Mecatrónica.
2. Incluir una librería de componentes electrónicos digitales de acuerdo los requerimientos de las materias con temas de electrónica digital y de los docentes.
3. Facilitar el desarrollo de experimentos o prácticas en temas de electrónica digital, al determinar previamente las condiciones eléctricas de los circuitos.
4. Desarrollar auto equipamiento en el área de la electrónica digital.
5. Diseñar y construir una herramienta que sea reproducible por los alumnos y de fácil uso.

3. Metas: cumplimiento de Metas

Metas Propuestas:

1-Desarrollar un equipo de prueba de circuitos digitales que tenga la funcionalidad como un equipo comercial, más las ventajas de una interface gráfica con propósito didáctico, que cubra en su totalidad la necesidad de pruebas de operación de al menos 150 componentes electrónicos digitales para uso en la materia electrónica digital de la carrera de Ingeniería Mecatrónica.

2-Obtener un probador de circuitos digitales que funcione como un equipo terminado (no como un prototipo), de fácil reproducción y bajo costo, para ser fabricado por los propios alumnos de los tecnológicos.

3-Eliminar en su totalidad el retraso al realizar las prácticas o experimentos de electrónica con componentes en mal estado, ya que este dispositivo por su bajo costo y fácil construcción, los alumnos lo va a poder portar como su equipo de trabajo personal.

4-Obtener un equipo con un software de control que trabaje con algoritmos de prueba estructurado para poder agregar nueva lista de componentes de una manera fácil y sencilla, como parte de actualización y evitar obsolescencia.

Metas Cumplidas:

1-Se logró desarrollar un equipo probador de circuitos digitales semejantes en funcionalidad a algunos equipos comerciales del rango económico, como se muestra en la Imagen 59.

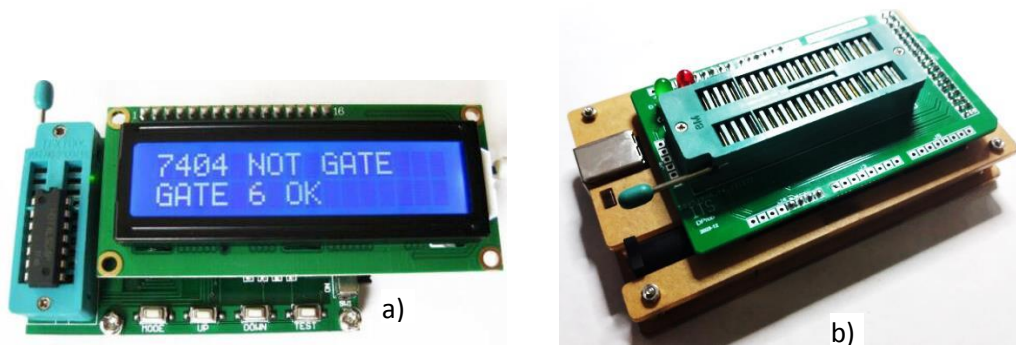


Imagen 59, Comparación de probadores, a) Probador de circuitos Comercial, b) Probador desarrollado en el presente proyecto. 11cm x 7cm x 4cm.

El probador desarrollado tiene una interfaz de propósito didáctico como se muestra Imagen 27. Pantalla de trabajo de prueba de circuitos., lo cual en los probadores comerciales no existe. Además, cuenta con una lista de componentes que cumplen completamente las necesidades de la materia de electrónica digital en la carrera de ingeniería electrónica cómo se muestra en la Tabla 2. Lista de compuertas TTL y CMOS a manejar por el probador de circuitos).

2-El probador desarrollado en el presente proyecto cuenta con la funcionalidad de un proyecto terminado (Imagen 59 b) dado que para su uso solo es necesario conectarlo a una computadora con el software de control. El bajo costo de aproximadamente \$354 MX lo hace apto para ser fabricado por los alumnos para su uso personal.

3- El resultado mostrado de la prueba de un circuito (Imagen 60) donde se determina con precisión la parte dañada del componente, elimina la pérdida de tiempo cuando los alumnos realizan prácticas o experimentos de electrónica con componentes dañados.

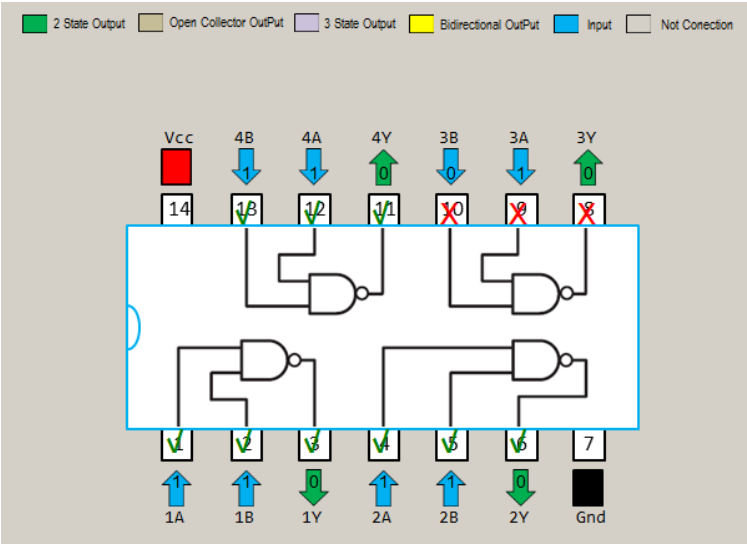


Imagen 60 Representación de Resultado de prueba Automática.

4-Se diseñó el software de tal forma que los algoritmos de pruebas trabajan con tablas que definen las pruebas a realizar a cada circuito y los parámetros necesario, como se muestra en la Tabla 19, lo cual permite actualizar la lista de componentes sin mayor dificultad.

Tabla 19 pruebas en circuito con diferentes compuertas en el mismo componente.

Prueba	Pins 40-32	Pins 32-25	Pins 24-17	Pins 16-9	Pins 8-1	Prueba	Tipo	Salida	Gate	Gate	Tipo	Prueba	Tipo	Salida
CI	11111111	00000000	00000000	00000000	11111111	CD	NAND	0	1	2	CD	NOR	0	3

5- Resultados concretos obtenidos en forma cuantitativa respecto a tesis desarrolladas, publicaciones trabajos de residencia profesional, patentes en trámite, participación en eventos, etc.

Se proyectó al menos enviar para publicación un artículo al Coloquio de investigación multidisciplinaria del tecnológico de Orizaba CIM programado para octubre del 2023, que coincide con el segundo

semestre de desarrollo del proyecto, fecha donde ya se cuenta con el avance suficiente para hacer una publicación sobre el proyecto.

Metas cuantificables	Cantidad programada	Cantidad lograda	Observaciones
Incorporación de estudiantes de licenciatura al proyecto (servicio social) :	1	1	Actividad: Desarrollo del software de control,
Artículos científicos publicados en revistas indizadas, enviados	1	1	Coloquio de Investigación Multidisciplinaria, CIM Orizaba (Ver Imagen 5, Evidencia)

Incorporación de estudiantes de licenciatura al proyecto (servicio social):	
Fecha	Nombre de los estudiantes incorporados y su Programa Educativo
22-May-2023	Leonardo Contreras Martínez Ing. En Sistemas Computacionales.

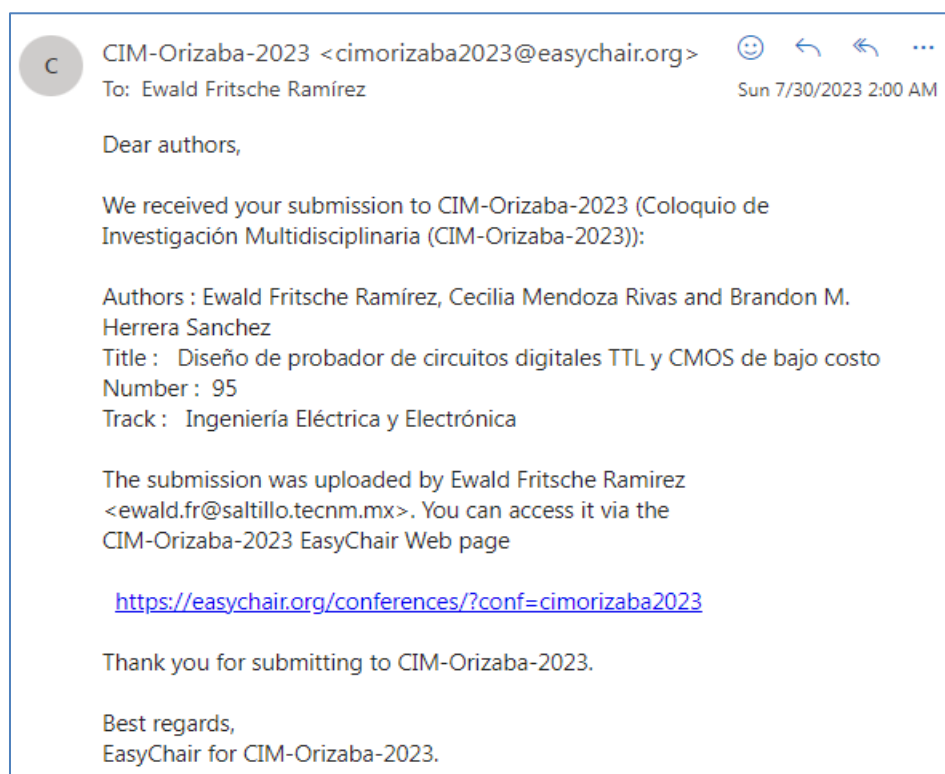


Imagen 61. Evidencia de envío de artículo a publicación a CIM del I.T. de Orizaba

Se proyectó la participación de un estudiante de servicio social del área de sistemas para trabajar con el desarrollo del software de control del probador de circuitos digitales.

Se asignó al alumno Leonardo Contreras Martínez, trabajar con la interface gráfica y estructura del área de trabajo del software,

4. Metodología:

- La parte inicial fue definir la lista de componentes porque está debe contener todos los circuitos disponibles en el laboratorio de electrónica digital, más el requerimiento especial de componentes electrónicos por los docentes que imparten la materia de electrónica digital para la carrera de Mecatrónica como se muestra en la Tabla 2.

- Se procedió a la recopilación de los datos técnicos de los componentes electrónicos los cuales son los elementos a someterse a prueba, se requirió de un análisis muy detallado de las características de los componentes digitales a probar, El análisis estuvo enfocado a determinar los siguientes aspectos:

- | | |
|---------------------------|------------------------------|
| - Función lógica | - Distribución de terminales |
| - Número de terminales | - Tipos de entrada y salida |
| - Forma física | - Secuencia de prueba |
| - Diagrama representativo | |

- Fue necesario determinar una clasificación de los componentes digitales porque define el tipo de prueba lógica a realizar (Tabla 1), así como en que terminales del circuito se lleva a cabo.

- La información se organizó en tablas que representa la base de datos del sistema probador, como se muestra en las tablas 4, 5, 6, 7 y 8.

- Se clasifico de manera estructurada porque impacta la etapa siguiente del proyecto la cual es la programación de los algoritmos de la lógica de prueba, además, permite agregar nuevos componentes que no estén previstos en la lista inicial como parte de la actualización del software.

- Las pruebas en los componentes se dividieron en dos pruebas principales,

-Pruebas para circuitos Combinacionales (Circuitos formados por combinación de compuertas digitales).

-Pruebas para circuitos Secuenciales (Circuitos formados con elementos de memoria Flip-Flops).

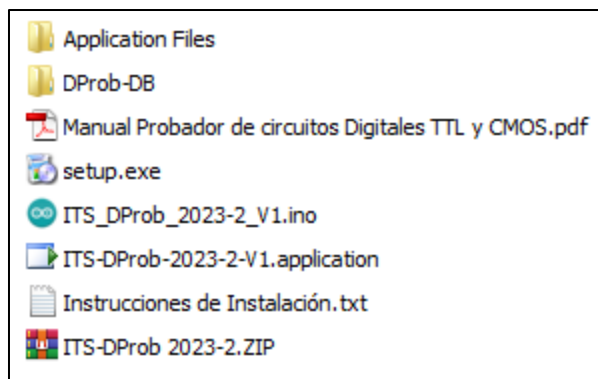
La secuencia de prueba es a través de 3 pasos fundamentales ,1- aplicar valores lógicos a las entradas , 2-leer los resultados en la salidas y 3- comparar con la lógica del circuito.

- La programación de los algoritmos de prueba, están directamente ligados a la estructura de la base de datos para los tipos de prueba qué requiere cada uno de los componentes electrónicos, dado que la base de datos definen la secuencia de activación de las terminales en cada circuito al momento de ser probados.

- Para lograr una interfaz gráfica se realizó el software de control de la computadora en un lenguaje Visual Orientado a Objetos y la contraparte del programa de control que va instalado en el módulo Arduino mega se realizó en lengua C.

5. Difusión

El equipo resultado del desarrollo de este proyecto, es un probador de circuitos digitales portátil de bajo costo y portátil como se planteó en el objetivo inicial, además, cuenta con la ventaja de que al tener la ingeniería de desarrollo, los alumnos lo pueden duplicar para su uso en prácticas o experimentos de electrónica. La difusión de la información del proyecto es a través de un enlace en la nube enviado **cada semestre** por correo a los alumnos de las materias de electrónica digital o bien, por medio de un enlace dentro de la plataforma educativa que los docentes manejen. Dicha información contiene los archivos mostrados en el Listado 13 para la instalación del software y construcción física del probador de circuitos.



Listado 13, Archivos de distribución del probador de circuitos.

En el Listado 13 se tiene los siguientes archivos:

Instrucciones de Instalación.txt. Archivo de texto con indicaciones, primer archivo a leer antes de instalar.

setup.exe. Programa de instalación.

Carpeta “DProb-DB. Es la base de datos y archivos complementarios.

ITS-DProb 2023-2.ZIP. Archivo comprimido que contiene los archivos en formato Gerber para la construcción profesional de la placa de circuito impreso PCB.

ITS_DProb_2023-2_V1.ino. Programa que requiere instalarse en el módulo de arduino Mega2560.

Manual Probador de circuitos Digitales TTL y CMOS.pdf. Por último, un manual de usuario y construcción del probador de circuitos digital TTL.

Al estar los archivos en un enlace en la nube, permite que los alumnos tengan acceso rápidamente, además, facilita que las actualizaciones estén a disposición de manera inmediata.

Enlace digital:

[Distribución de software, Probador de circuitos Digitales](https://tecnmsaltillo-my.sharepoint.com/:f/g/personal/ewald_fr_saltillo_tecnm_mx/Eo1s1IsU0GRlvrWPubI6AosBu7RyobgiNwAkAOgvCGeDqg?e=hjQv2x)

[https://tecnmsaltillo-](https://tecnmsaltillo-my.sharepoint.com/:f/g/personal/ewald_fr_saltillo_tecnm_mx/Eo1s1IsU0GRlvrWPubI6AosBu7RyobgiNwAkAOgvCGeDqg?e=hjQv2x)

[my.sharepoint.com/:f/g/personal/ewald_fr_saltillo_tecnm_mx/Eo1s1IsU0GRlvrWPubI6AosBu7RyobgiNwAkAOgvCGeDqg?e=hjQv2x](https://tecnmsaltillo-my.sharepoint.com/:f/g/personal/ewald_fr_saltillo_tecnm_mx/Eo1s1IsU0GRlvrWPubI6AosBu7RyobgiNwAkAOgvCGeDqg?e=hjQv2x)

6. Beneficios y Problemas

Beneficios:

El principal beneficio del probador de circuitos desarrollado en este proyecto, consiste eliminar el tiempo perdido al realizar prácticas o experimentos con circuitos digitales dañados por el uso frecuente de estos, lo que vuelve más eficiente el desempeño de los alumnos.

Aunque en los laboratorio de electrónica de los institutos tecnológicos se cuenta equipo de prueba de circuitos de circuitos digitales, su uso solo puede ser dentro del área de trabajo en los laboratorios ya que son equipos costosos, el presente proyecto facilita que el alumno avance en los preparativos de su experimento electrónico previo a presentar su resultado en el laboratorio ante el docente ya que es equipo portátil y económico de fabricar.

La creciente demanda de alumnos en la carrera de Mecatrónica también ha incrementado el tiempo invertido en realizar prácticas de electrónica, es difícil poder probar todos los componentes usados por un grupo de alumnos numeroso en el tiempo destinado para realizar una práctica o experimento, así que este proyecto permite que cada alumno pueda probar los circuitos en forma personal reduciendo el retraso..

El diseño didáctico del probador de circuitos refuerza la enseñanza en el aspecto práctico de temas de electrónica digital, así como, contribuye en el auto equipamiento con características orientadas a las necesidades propias de los programas de estudio.

Problemas:

El probador de circuitos desarrollado está diseñado para trabajar con software que incluye interface gráfica, para tal efecto se tiene que disponer de una computadora PC para su uso, lo cual puede ser un inconveniente si no se cuenta con equipó de cómputo.

La portabilidad del probador de circuitos con dimensiones pequeñas 11cm x 7cm x 4cm, lo expone a riesgos de accidentes como caídas por el manejo con las manos, lo que podría causar daños físicos y eléctricos.

La base ZIF del probador de circuitos, es un sistema mecánico de contacto por medio de palanca de sujeción del circuito, esta requiere un manejo adecuado para evitar daño.

7. Información adicional

Dado que el equipo desarrollado en este proyecto es controlado con un software, también se incluye un manual de instalación del software y el procedimiento construcción del equipo para que los alumnos lo fabriquen para su uso en clases, el manual se encuentra como archivo PDF dentro del software para distribuirse cada semestre a los docentes y alumnos.

Enlace digital:

[Distribución de software, Probador de circuitos Digitales](#)

<https://tecnmsaltillo->

my.sharepoint.com/:f/g/personal/ewald_fr_saltillo_tecnm_mx/Eo1s1IsU0GRlvrWPubI6AosBu7RyobgiNw

[AkAOgvCGeDqg?e=hjQv2x](#)

M.E.S. Teresita del Rosario Borrego Jiménez Subdirectora Académica	Ing. Ewald Fritsche Ramírez Docente del Área Eléctrica Electrónica
---	---