



**UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR**

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

MENCIÓN SISTEMAS INDUSTRIALES

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO ELECTRÓNICO

PROYECTO TÉCNICO:

**DISEÑO E IMPLEMENTACIÓN DE MÓDULOS DIDÁCTICOS PARA EL
DESARROLLO DE APLICACIONES HMI MEDIANTE EL USO DE
MICROCONTROLADORES SMD Y PANTALLAS NEXTION**

AUTORES:

FRANCK JOEL BANGUERA DEL CASTILLO

BORIS ALEXANDER MERA BAUTISTA

TUTOR:

ING. LUIS NEIRA CLEMENTE, MSc.

GUAYAQUIL – ECUADOR



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

MENCIÓN SISTEMAS INDUSTRIALES

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

Nosotros, Franck Joel Banguera Del Castillo con cédula de identidad N° 080331153-9 y Boris Alexander Mera Bautista con cédula de identidad N° 093041291-1, estudiantes de la carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana declaramos que el trabajo descrito aquí ha sido desarrollado respetando los derechos intelectuales de terceros cuyas fuentes se incorporan en la bibliografía. Los análisis realizados y conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Guayaquil, octubre del 2021

Franck Joel Banguera Del Castillo

C.I. 080331153-9

Boris Alexander Mera Bautista

C.I. 093041291-1



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

MENCIÓN SISTEMAS INDUSTRIALES

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR

A través del presente certificado, se ceden los derechos de propiedad intelectual correspondiente a este trabajo a la Universidad Politécnica Salesiana, según lo establecido por la ley de propiedad intelectual y por su normatividad institucional vigente.

Guayaquil, octubre del 2021

Franck Joel Banguera Del Castillo

C.I. 080331153-9

Boris Alexander Mera Bautista

C.I. 093041291-1



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

MENCIÓN SISTEMAS INDUSTRIALES

**CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE
TITULACIÓN**

Por medio de la presente constancia que los Señores Franck Joel Banguera Del Castillo y Boris Alexander Mera Bautista han desarrollado y elaborado satisfactoriamente el proyecto final de titulación, que se ajusta a las normas establecidas por la Universidad Politécnica Salesiana, por tanto, autorizo su presentación para los fines legales pertinentes.

A handwritten signature in black ink, appearing to read 'Luis Neira Clemente', is written above a horizontal line.

Ing. Luis Neira Clemente, MSc.

DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Quiero agradecer en primer lugar a Dios, por ayudarme a entender que en la vida podrán existir millón de obstáculos pero que está en mí el querer enfrentarlos y superarlos, que sin su ayuda, guía y bendición nada sería posible. A mi mamá por no solo haberme ayudado y guiado en esta etapa de mi vida, sino también por ser mi mayor motivación de superación y lucha para culminar de manera satisfactoria mi carrera, sin duda quiero darle mi mayor y más sincero agradecimiento a ella, porque nada de esto hubiera sido posible sin su bendición y amor.

Este ha sido un camino largo lleno de inconvenientes, baches e innumerables reproches, pero también ha sido uno lleno de emoción de júbilo, conocimientos y de mucho aprendizaje, que al lograr cumplir con la meta planteada y todos mis objetivos me llena de orgullo y mucha satisfacción por el deber cumplido porque esta es una meta más que con mucho esfuerzo y responsabilidad se logró llevar adelante. Esto es por y para ustedes.

Franck Joel Banguera Del Castillo

AGRADECIMIENTO

Toda persona se encuentra en un reto diario y en ciertos puntos esos retos son a corto o a largo plazo, en mi trabajo le agradezco en primer lugar a DIOS por bendecirme en este camino y porque me permitió hacer realidad este sueño anhelado.

A la Universidad Politécnica Salesiana por darme la oportunidad de estudiar y ser un profesional.

A mis padres y a mis docentes, MSc. Luis Neira Clemente, MSc. Luis Córdova Rivadeneira, MSc. César Cáceres Galán y MSc. Gianfranco Di Mattia Castro, que gracias a sus enseñanzas he superado con éxito toda esta aventura que llega a su fin. Agradezco a mis verdaderos amigos: Zully Peñafiel Arévalo y Jamil Delgado Ávila, que me dieron su apoyo y fidelidad en momentos difíciles, pero no imposibles en los que salí victorioso con la bendición de Dios.

En sí, quiero agradecerle a Dios por darme la oportunidad de haber conocido a personas maravillosas en este viaje para convertirme en un profesional.

Boris Alexander Mera Bautista

DEDICATORIA

Dedico esta tesis a mi mamá y hermana que confiaron siempre en mí, especialmente a mi mamá que no solo ahora en la universidad sino a lo largo de mi vida me ha ayudado a alcanzar mis logros y me ha apoyado para conseguirlos, sin quejarse ni mucho menos rendirse conmigo, no me cabe la menor duda de que ella es el soporte más importante en mi vida siempre y por querer que siempre haga algo nuevo y mejor en mí vida. También quisiera dedicarle esta tesis a mi amigo Gabriel Calero que sin duda fue una pieza fundamental en este maravilloso rompecabezas, demostrándome que todo es posible, con esfuerzo, dedicación y mucha responsabilidad podemos alcanzar grandes cosas en la vida.

Franck Joel Banguera Del Castillo

DEDICATORIA

Este proyecto se lo dedico en primer lugar a DIOS por ser la luz, la fuerza y la esperanza en esta larga aventura dándome sabiduría para sobresalir en cada obstáculo y siempre demostrándome su amor.

En segundo lugar, le dedico este proyecto a mis padres que me dieron la vida y me apoyaron en toda esta aventura; uno de mis pilares en esta aventura fue MSc. Belinda Cubero, Msc. Martha Rodríguez Farias y MSc. Susana Tamayo quienes se convirtieron en mi segunda madre, dándome consejos para superar momentos difíciles de mi vida, también debo dedicar este trabajo a mi Compañero de tesis Franck Joel Banguera del Castillo quien con su paciencia estuvo en momentos inolvidables en esta travesía.

Les agradezco con mi vida porque me apoyaron y alentaron cuando pensaba en rendirme, pero estuvieron ahí siempre para darme alientos y seguir con el sueño de ser Ingeniero.

Por último, a tres seres que demostraron que las adversidades te hacen más fuerte y que todo lo que sucede es por el bien de uno y voluntad de Dios, al MSc. Luis Córdova Rivadeneira, MSc. Luis Neira Clemente, MSc. César Cáceres Galán y MSc. Gianfranco Di Mattia Castro que fueron y son un ejemplo para mí, demostrando que los obstáculos en la vida siempre van a existir, pero con amor y fe en Dios todo se puede lograr.

Boris Alexander Mera Bautista

Índice General

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA	I
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR.....	II
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....	III
AGRADECIMIENTO	IV
AGRADECIMIENTO	V
DEDICATORIA	VI
DEDICATORIA	VII
Índice de Figuras.....	XII
Índice de Tablas	XVI
Resumen.....	XVII
Abstract.....	XVIII
Introducción	1
Capítulo 1: Descripción General.....	2
1.1. Importancia y Alcance	2
1.2. Justificación.....	2
1.3. Delimitación	3
1.3.1. Delimitación Temporal	3
1.3.2. Delimitación Espacial	3
1.3.3. Delimitación Académica.....	3
1.4. Objetivos	3
1.4.1. Objetivo General.....	3
1.4.2. Objetivos Específicos.....	4
1.5. Hipótesis.....	4
1.6. Metodología de Investigación	4
Capítulo 2: Marco Teórico.....	6
2.1. Estado del Arte.....	6
2.2. Buzzer.....	10
2.3. Relé.....	11
2.4. Led RGB	12
2.5. Sensor Infrarrojo	15

2.6.	LM324.....	16
2.7.	L293D.....	17
2.8.	Motor DC	19
2.9.	MOC3021.....	20
2.10.	Triac BT16.....	21
2.11.	LM35	22
2.12.	Pic 18F8720.....	23
2.13.	Teclado Matricial de 4X3	25
2.14.	LCD 4X20	26
2.15.	Pantalla NEXTION.....	28
2.16.	DS1307	33
2.17.	24LC512.....	34
Capítulo 3: Diseño y Construcción de Módulos Didácticos.....		36
3.1.	Consideraciones iniciales del diseño.....	36
3.1.1.	Funciones de cada tarjeta.....	36
3.1.2.	Descripción del Módulo principal	36
3.1.3.	Características de las plantas de trabajo.....	38
3.2.	Diseño Esquemático del Módulo Didáctico.....	39
3.2.1.	Diseño esquemático de la tarjeta principal	39
3.2.2.	Diseño esquemático de la Planta de Temperatura	40
3.2.3.	Diseño esquemático de la Planta de Velocidad	41
3.2.4.	Diseño esquemático de la Planta de Luminosidad.....	42
3.3.	Diseño del Circuito Impreso PCB del Módulo Didáctico.....	42
3.3.1.	Diseño del PCB de la tarjeta principal.....	42
3.3.2.	Diseño del PCB de la Planta de Temperatura.....	43
3.3.3.	Diseño del PCB de la Planta de Velocidad.....	44
3.3.4.	Diseño del PCB de la Planta de Luminosidad	45
3.4.	Implementación de las tarjetas Electrónicas	45
3.4.1.	Listado de elementos de cada tarjeta	45
3.4.1.1.	Tarjeta Principal	45
□	Comunicación Síncrona Serie	45

<input type="checkbox"/> Diseño de Relé.....	46
<input type="checkbox"/> Puerto RD07	46
<input type="checkbox"/> Entrada de funcionamiento.....	47
<input type="checkbox"/> Diseño de Buzzer.....	47
<input type="checkbox"/> Diseño de LCD	48
<input type="checkbox"/> Salidas.....	48
<input type="checkbox"/> Interruptores.....	48
<input type="checkbox"/> Diseño Teclado	49
<input type="checkbox"/> Comunicación Rx Tx.....	49
<input type="checkbox"/> Diseño de módulo principal.....	50
3.4.1.2. Planta de Temperatura.....	50
<input type="checkbox"/> Diseño de planta	50
3.4.1.3. Planta de Velocidad.....	50
<input type="checkbox"/> Diseño de planta	50
3.4.1.4. Planta de Luminosidad	51
<input type="checkbox"/> Diseño de planta	51
3.4.2. Ubicación y soldado de los elementos	51
3.5. Construcción de Módulos Didácticos	53
3.6. Pruebas de Funcionamiento	54
3.6.1. Comprobación de la programación del microcontrolador	54
3.6.2. Comprobación de los puertos de entrada y salida.....	55
3.6.3. Comprobación del funcionamiento de las pantallas gráficas LCD y NEXTION ...	55
Capítulo 4: Desarrollo de Prácticas Propuestas	59
4.1. Práctica # 1: Manejo de puertos del microcontrolador como salidas mediante la activación de las pantallas NEXTION.....	59
4.2. Práctica # 2: Manejo de puertos del microcontrolador como entradas mediante la activación de las pantallas NEXTION.	60
4.3. Práctica # 3: Manejo de herramientas de Nextion para elaboración de procesos industriales.	61
4.4. Práctica # 4: Elaboración de librerías de objetos para procesos industriales.	62

4.5. Práctica # 5: Manejo de pantallas para la elaboración de menú en diferentes aplicaciones.	
63	
4.6. Práctica # 6: Aprendizaje de manejo de simulador de pantallas NEXTION.	64
4.7. Práctica # 7: Manejo Led RGB mediante el uso de pantallas Nextion.	65
4.8. Práctica # 8:.....	66
Proceso Industrial #1: Control de luminosidad mediante seteo y visualización en pantalla NEXTION.	66
4.9. Práctica # 9:.....	67
Proceso Industrial #2: Control de temperatura mediante seteo y visualización en pantalla NEXTION.	67
4.10. Práctica # 10:	68
Proceso Industrial #3: Control de variación de velocidad de motor DC mediante seteo y visualización en pantalla NEXTION.....	68
Cronograma.....	69
Presupuesto	70
Conclusiones.....	72
Recomendaciones	73
Bibliografía.....	74
ANEXOS	77
ANEXO 1: Solución propuesta de la práctica #1.....	78
ANEXO 2: Solución propuesta de la práctica #2.....	84
ANEXO 3: Solución propuesta de la práctica #3.....	90
ANEXO 4: Solución propuesta de la práctica #4.....	95
ANEXO 5: Solución propuesta de la práctica #5.....	100
ANEXO 6: Solución propuesta de la práctica #6.....	105
ANEXO 7: Solución propuesta de la práctica #7.....	108
ANEXO 8: Solución propuesta de la práctica #8.....	113
ANEXO 9: Solución propuesta de la práctica #9.....	120
ANEXO 10: Solución propuesta de la práctica #10.....	127

Índice de Figuras

Fig. 1.- Módulo de control y medición de temperatura abiertos.....	6
Fig. 2.- módulos didácticos básicos basados en el microcontrolador atmega164	7
Fig. 3.- Diseño esquemático de sistema de control de tubos de rayos X.....	8
Fig. 4.- Entrenadores Didácticos.....	9
Fig. 5.- Componentes de un Buzzer.....	10
Fig. 6.- Buzzer con Oscilador Interno.....	10
Fig. 7.- Diseño 2D de Buzzer.....	11
Fig. 8.- Funcionamiento de un relé	11
Fig. 9.- Relé de 5 Vdc	12
Fig. 10.- Combinación RGB	13
Fig. 11.- LED RGB configuración común.....	14
Fig. 12.- Dimensiones del Paquete Led RGB.....	14
Fig. 13.- Sensor Infrarrojo de Herradura	15
Fig. 14.- Diagrama de conexión.....	16
Fig. 15.- Vista Lateral LM324	17
Fig. 16.- Vista Superior LM324.....	17
Fig. 17.- Integrado L293D	17
Fig. 18.- Conexionado para un motor con giro en ambos sentidos (lado izquierdo) y con motores con giro en sentido único en dos salidas (lado derecho).....	18
Fig. 19.- Distribución de pines de L293D.....	19
Fig. 20.- Motor DC	19
Fig. 21.- Descripción de Pines del MOC3021	20
Fig. 22.- Gráfica de Funcionamiento del MOC3021	21
Fig. 23.- Triac BT 16	22
Fig. 24.- Sensor LM35	23
Fig. 25.- Pic 18F8720 en Montaje Superficial.....	25
Fig. 26.- Teclado Matricial 4x3	26
Fig. 27.- LCD 20x4.....	27
Fig. 28.- Conexión de LCD 20x4.....	28
Fig. 29.- Pantalla Nextion	29

Fig. 30.- Interfaz principal Nextion Editor	33
Fig. 31.- Pines del módulo DS1307	34
Fig. 32.- Integrado 24LC512	35
Fig. 33.- Flujograma de funcionamiento.....	38
Fig. 34.- Esquemático de Tarjeta Principal.....	40
Fig. 35.- Esquemático de Planta de Temperatura	41
Fig. 36.- Esquemático de Planta de Velocidad	41
Fig. 37.- Esquemático de Planta de Luminosidad.....	42
Fig. 38.- PCB de Tarjeta Madre.....	43
Fig. 39.- PCB de Planta de Temperatura	44
Fig. 40.- PCB de Planta de Velocidad	44
Fig. 41.- PCB de Planta de Luminosidad.....	45
Fig. 42.- Comunicación Síncrona Serie	46
Fig. 43.- Relé.....	46
Fig. 44.- Puerto D.....	46
Fig. 45.- Entrada General.....	47
Fig. 46.- Buzzer.....	48
Fig. 47.- LCD	48
Fig. 48.- Salidas Generales	48
Fig. 49.- Interruptores	49
Fig. 50.- Teclado Numérico 4x3	49
Fig. 51.- Comunicación Rx Tx	49
Fig. 52.- Pic 18F8720.....	50
Fig. 53.- Placa de Soldado Principal.....	52
Fig. 54.- Elementos ubicados en la Tarjeta principal.....	52
Fig. 55.- Soldado de elementos en tarjeta principal.....	53
Fig. 56.- Diseño en INVENTOR	53
Fig. 57.- Medidas de estructura.....	54
Fig. 58.- Funcionamiento del Pic 18F8720.....	55
Fig. 59.- Práctica #2 funcionamiento de LCD y puertos	56
Fig. 60. Funcionamiento en tarjeta principal	56

Fig. 61.- Pantalla 1 Generador de Ozono.....	57
Fig. 62.- Pantalla 2 generador de ozono	57
Fig. 63.- Codificación de Pantalla 1.....	58
Fig. 64.- Codificación de Pantalla 2.....	58
Fig. 65.- Funcionamiento del Ozonizador mediante Placa Madre y Pantalla NEXTION	58
Fig. 66.- Cronograma de trabajo	69
Fig. 67.- Creación de proyecto en PIC C Compiler.....	78
Fig. 68.- Códigos predeterminados de inicialización.....	78
Fig. 69.- Interrupción Interna.....	79
Fig. 70.- Puerto como salida y Mensaje de Bienvenida.....	79
Fig. 71.- Creación de proyecto en Nextion	80
Fig. 72.- Creación de páginas de trabajo.....	80
Fig. 73.- Página 1 en Pantalla Nextion	81
Fig. 74.- Programación de botón Siguiente en Nextion.....	81
Fig. 75.- Pantalla de Puertos de Salida	82
Fig. 76.- Creación de proyecto en PIC C Compiler.....	84
Fig. 77.- Códigos predeterminados de inicialización.....	84
Fig. 78.- Codificación de envío de datos a la Pantalla Nextion.....	85
Fig. 79.- Puerto como entrada y Mensaje de Bienvenida	86
Fig. 80.- Creación de proyecto en Nextion	87
Fig. 81.- Creación de páginas de trabajo.....	87
Fig. 82.- Página 1 en Pantalla Nextion	88
Fig. 83.- Programación de botón Siguiente en Nextion.....	88
Fig. 84.- Pantalla de Puertos de Salida	89
Fig. 85.- Creación de proyecto en Nextion	90
Fig. 86.- Creación de páginas de trabajo.....	90
Fig. 87.- Elementos de página 1 en Práctica 3.....	91
Fig. 88.- Codificación de botón Configurar.....	91
Fig. 89.- Configuración QR código	92
Fig. 90.- Elementos página 2 de práctica 3	93
Fig. 91.- Creación de proyecto en Nextion.....	95

Fig. 92.- Creación de página de trabajo	95
Fig. 93.- Base de librería.....	96
Fig. 94.- Dimensiones en milímetros o pixeles.....	97
Fig. 95.- Submenú Toolbox	98
Fig. 96.- Creación de páginas de trabajo.....	100
Fig. 97.- Dimensiones de imágenes para pantalla HMI.....	100
Fig. 98.- Imágenes con dimensiones de pantalla HMI.....	101
Fig. 99.- Ambientes de casa.....	102
Fig. 100.- Codificación de botones en ambientes principales	102
Fig. 101.- Audios en librería principal.....	103
Fig. 102.- Inserción de audio en botón a través de atributos	104
Fig. 103.- Iconos de compilación y Carga de programación de pantalla HMI	104
Fig. 104.- Barra de herramientas.....	105
Fig. 105.- Menú de DEVICE en simulador	105
Fig. 106.- Icono Debug inicia la simulación.....	106
Fig. 107.- Ventana de Simulación.....	106
Fig. 108.- Datos como entrada y salida en la simulación	107
Fig. 109.- Creación de proyecto en Nextion	108
Fig. 110.- Creación de páginas de trabajo.....	108
Fig. 111.- Estructura de página 2	109
Fig. 112.- Creación de proyecto en Nextion	113
Fig. 113.- Creación de páginas de trabajo.....	113
Fig. 114.- Estructura de página 2	114
Fig. 115.- Creación de proyecto en Nextion	120
Fig. 116.- Creación de páginas de trabajo.....	120
Fig. 117.- Estructura de página 2	121
Ilustración 118.- Estructura de página 3	121
Ilustración 119.- Estructura de página 4	122
Fig. 120.- Creación de proyecto en Nextion	127
Fig. 121.- Creación de páginas de trabajo.....	127
Fig. 122.- Estructura de página 2	128

Fig. 123.- Estructura de página 3	128
Fig. 124.- Estructura de página 4	129

Índice de Tablas

Tabla 1.- Código Hexadecimal Led RGB.....	13
Tabla 2.- Presupuesto.....	70

Resumen

Año	Estudiantes	Tutor de Proyecto de Titulación	Proyecto de Titulación
2021	Franck J. Banguera D. Boris A. Mera B.	Ing. Luis Neira Clemente, MSc.	Diseño e Implementación de módulos didácticos para el desarrollo de aplicaciones HMI mediante el uso de microcontroladores SMD y pantallas Nextion

En el trabajo técnico o proyecto se estudia la importancia de la parte experimental para el éxito de un aprendizaje significativo, donde se relaciona el diseño e implementación de módulos didácticos, para el desarrollo de aplicaciones mediante el uso de microcontrolador SMD (dispositivo de montaje superficial) y de pantalla HMI (Interfaz Hombre Máquina) Nextion que en referencia con las demás es de bajo costo. La relación entre teoría y práctica permite a los estudiantes vivir la experiencia de brindar soluciones factibles y concretas para cualquier situación, ya sea en un proceso industrial o simplemente utilizar tecnología domótica para el control en una vivienda. Para el desarrollo de módulos didácticos se utilizan conocimientos importantes en ingeniería, por el hecho de no tener una estructura fija, se puede utilizar con diferentes enfoques para seguir investigando. Con la ayuda de herramientas de diseño, se puede demostrar que el conocimiento adquirido es relevante para los estudiantes. El uso de microcontroladores permite tener un abanico de opciones las cuales condescenderá que el trabajo no sea unidimensional sino que se ajuste de acuerdo con la propuesta del programador. Este trabajo promueve la oportunidad de realizar aportes mediante la investigación, motivando a la adquisición de mayor conocimiento, que permitirá mayores aplicaciones mediante la interpretación de diferentes lenguajes de programación basándose en la simulación y en la implementación de cada una de las ideas, generando soluciones desde el punto de vista técnico como el diseño de un PID discreto mediante salidas digitales apoyadas del PWM.

Palabras Claves: Pantalla Nextion, Microcontroladores, SMD, Aprendizaje significativo, PID discreto.

Abstract

Year	Students	Degree Project Tutor	Technical Degree Project
2021	Franck J. Banguera D. Boris A. Mera B.	Eng. Luis Neira Clemente, MSc.	Design and Imation of teaching modules for the development of HMI applications by using SMD microcontrollers and Nextion screens

This technical work or project studies the importance of the experimental part for the success of meaningful learning, where the design and implementation of teaching modules is related, to the development of applications through the use of SMD microcontroller (surface mount device) and HMI (Machine Man Interface) Nextion screens that in reference to others is inexpensive. The relationship between theory and practice allows students to live the experience of providing feasible and concrete solutions to any situation, whether in an industrial process or simply using home automation technology for home control. Important engineering knowledge is used for the development of teaching modules because it does not have a fixed structure, it can be used with different approaches to further research. With the help of design tools, you can demonstrate that the knowledge acquired is relevant to students. The use of microcontrollers allows to have a range of options which will condensing that the work is not one-dimensional but adjusted according to the programmer's proposal. This work promotes the opportunity to make contributions through research, motivating the acquisition of greater knowledge, which will allow greater applications by interpreting different programming languages based on the simulation and implementation of each of the ideas, generating solutions from the technical point of view such as the design of a discrete PID through digital outputs supported by the PWM.

Keywords: Nextion display, Microcontrollers, SMD, Meaningful learning, discrete PID.

Introducción

A través de este proyecto se pretende obtener una relación más directa entre la parte teórica y lo práctico, que sea eficaz en el diseño de programas y con un valor en el mercado d bajo costo, ya que este es el principal factor por el cual no todos los estudiantes implementas sus programas en casa ni hacen las pruebas respectivas. En efecto, a lo mencionado anteriormente, este trabajo se presenta el desarrollo de módulos didácticos que van enfocado a la comunicación de un microcontrolador SMD con una pantalla HMI que trabajan como entradas y salidas de acuerdo a lo que se pida en el diseño de prácticas, con esto los estudiantes podrán llevar un mayor estudio teórico y práctico en el desarrollo de prácticas de diferentes materias utilizando un microcontrolador y visualizándolas en dicha pantalla HMI sin tener el riesgo de pérdida de elementos o componentes en cada uno de sus diseños: también se puede observar cuales son las diferencias entre la implementación en su diseño simulado y su diseño implementado verificando su parte teórica.

El objetivo principal de este trabajo es ayudar a la comunidad educativa salesiana a que experimente sus propios programas o códigos en un módulo que permite imaginar cómo sería su funcionamiento en una empresa o institución para que así sus soluciones sean reales y eficientes dándole la oportunidad de que tenga experiencia vivida. La implementación de los módulos permite verificar que se cumplan los objetivos de estudio planteados, obteniendo resultados eficientes en las materias, que finalmente permiten verificar y validar los requisitos funcionales de prácticas. El comportamiento de los módulos, según las pruebas experimentales, ha cumplido con las características y funciones planteadas al inicio de este trabajo permitiendo una suficiente cobertura.

Consta de 4 secciones, siendo la primera la descripción del problema que se va a tratar en este proyecto, su importancia, alcance, delimitaciones y objetivos a cumplir. En la segunda sección (Marco Teórico) que abarca lo que es el Estado del Arte de proyectos vinculados a este y de los componentes que se utilizan en la implementación de los módulos. Para la tercera sección (Diseño Electrónico) se resume en cómo se diseñó dichos módulos y la ubicación de cada uno de ellos. La cuarta y última sección (Desarrollo de prácticas) se resumen el planteamiento de las diferentes prácticas que se formularon para este proyecto.

Capítulo 1: Descripción General

1. Planteamiento del Problema

1.1. Importancia y Alcance

Actualmente los estudiantes de diversas carreras se capacitan en diferentes áreas para mejorar sus conocimientos en base a los avances tecnológicos, sin olvidar que estos se ha convertido en uno de los productos fundamentales del consumo de la modernidad, es por ello que la Universidad Politécnica Salesiana en su Sede Guayaquil en el Campus Centenario cuenta con un laboratorio de Electrónica Digital, que se ve en la necesidad de incrementar módulos didácticos y de guías que contribuyan con el desarrollo de prácticas de diseño de plantas en simulación desde el punto de vista de la industria, que permitan reforzar los conocimientos teóricos impartidos por los docentes. Esta gran problemática es perjudicial para los estudiantes al momento de querer brindar una solución rápida y eficaz en este tipo de sistema ya sea en una empresa (pública o privada) o sino en un emprendimiento propio, es por ello por lo que se busca que los estudiantes puedan aplicar sus conocimientos mediante manuales sencillos y desarrollen habilidades aún más profundas en la experimentación.

1.2. Justificación

Mediante el uso de estos dispositivos los estudiantes de la Universidad Politécnica Salesiana de la Sede Guayaquil en el Campus Centenario, logran aprender a utilizar nuevas herramientas para la automatización y control de procesos industriales por medio de las pantallas NEXTION, que son de bajo costo en comparación a los HMI (interfaz hombre máquina) convencionales o tradicionales, su manejo y comunicación con los microcontroladores, logra que los estudiantes visualicen mediante gráficos y cuadros digitales el conocimiento teórico, impartido por los docentes de una forma eficaz y precisa. Estos módulos didácticos serán implementados con elementos modernos (el uso de componentes electrónicos en montaje superficial y utilización de módulos compuestos) y permiten su expansión de conocimientos en las demás asignaturas dando la posibilidad de

la conexión externa de elementos como una termorresistencia PT100 (sensor detector de temperatura por resistencia) o la comunicación con otros sistemas embebidos.

1.3. Delimitación

1.3.1. Delimitación Temporal

El proyecto técnico se efectuó en un tiempo de seis meses a partir de la fecha de aprobación de este, por los miembros de Consejo de la Carrera de Ingeniería Electrónica.

1.3.2. Delimitación Espacial

Este proyecto se realizó en las instalaciones de nuestro domicilio bajo todas las medidas de bioseguridad de la Ciudad de Guayaquil respetando las normativas vigentes del COE (Centro de Operaciones de Emergencia) cantonal.

1.3.3. Delimitación Académica

El proyecto implica en el diseño e implementación de módulos didácticos para aplicaciones HMI utilizando microcontroladores SMD (Surface Mounted Device o Dispositivos en Montaje Superficial) y pantallas Nextion, además de contener el desarrollo de 10 prácticas de simulación resueltas.

1.4. Objetivos

A continuación, se presenta el objetivo general de trabajo de titulación, seguido de los objetivos específicos que se determinaron para la investigación; parte fundamental para el desarrollo y experimentación de este.

1.4.1. Objetivo General

Diseñar e implementar módulos didácticos para el desarrollo de aplicaciones HMI a través del uso de microcontroladores SMD y pantallas Nextion.

1.4.2. Objetivos Específicos

- Diseñar módulos didácticos para la simulación de control en procesos industriales mediante microcontroladores SMD y pantallas Nextion.
- Diseñar e Implementar una planta para el control PID de temperatura monitoreada y controlada desde la pantalla Nextion.
- Diseñar e Implementar una planta para el control PID de velocidad para motores DC de baja potencia monitoreada y controlada desde la pantalla Nextion.
- Diseñar e Implementar una planta para el control PID de luminosidad utilizando Leds RGB monitoreada y controlada desde la pantalla Nextion.
- Desarrollar 10 prácticas en Pic C Compiler y en el IDE Nextion para el manejo de módulos.
- Elaborar guía detallada con la programación utilizada para el desarrollo de las prácticas.

1.5. Hipótesis

Con el desarrollo e implementación de estos módulos se estima lograr, que el entorno educativo pueda relacionar sus conocimientos teóricos con los prácticos y le permita la expansión de estos en las diversas áreas o asignaturas, visualizando que los HMI (interfaz hombre máquina) permitan plasmar nuestras creaciones. En caso de llegar a obtener los resultados esperados, el proyecto se implementa fácilmente a mayor número de módulos y se podrá utilizar en diversas asignaturas.

1.6. Metodología de Investigación

El diseño de esta guía de prácticas de laboratorio se basa bajo un concepto documental y experimental, que ante todo permitirá que los estudiantes que cursen la asignatura de Sistemas Microprocesados de la carrera de Electrónica y Automatización, de la Universidad Politécnica Salesiana Sede Guayaquil del Campus Centenario, complementen de forma exitosa la práctica y aprendizaje teórico para dar soluciones acopladas a la realidad. A través del uso del software Pic C Compiler se programa el microcontrolador permitiendo realizar pequeños cambios, esto mediante un dispositivo para la adquisición

de las señales que emitirá el microcontrolador para su conexión con las pantallas HMI (interfaz hombre máquina).

Por lo tanto, los métodos a utilizarse en este proyecto serán los siguientes:

- Método inductivo: Se aplica este método porque el diseño de módulos didácticos que ayuda a desarrollar aplicaciones HMI, por medio de microcontroladores y a monitorear la información sobre las gráficas y cuadros de mandos digitales de los procesos industriales.
- Método deductivo: Permite revisar datos en tiempo real de los procesos industriales mediante el desarrollo de las aplicaciones

Capítulo 2: Marco Teórico

2.1. Estado del Arte

(BENAVIDES CAGUASANGO & JIMÉNEZ BENALCÁZAR, 2016) En el proyecto técnico denominado **Diseño y Construcción de tres módulos didácticos para medición y control de temperatura ambiente e implementación de un SCADA en el Laboratorio de la Carrera de Ingeniería en Mantenimiento Eléctrico** de los Ingenieros Hugo Benavides y Jorge Jiménez detallan la importancia y necesidad de que los estudiantes puedan desarrollar habilidades reales a procesos industriales mediante el uso de los módulos y que se basa en la investigación tecnológica, descriptiva y documental sobre las tendencias en el campo de la automatización, para que el diseño de dichos módulos sea integral y lo más parecido a lo que se ve en una empresa. Explican cómo pueden generar las diferentes situaciones que se puedan vivir en una empresa y de su apreciación visual en el funcionamiento. El control de la lámpara usa un temporizador 555 y un convertidor AC/AC para ajustar el voltaje aplicado, cambiando así la temperatura mediante un sensor integrado, que se puede medir a temperaturas de hasta 60 ° C. El procesamiento de la señal física se realiza mediante el DAQ USB 6001 y el algoritmo de control PID es ejecutado por un programa desarrollado en el software LabVIEW. Para ajustar el controlador, se utiliza la técnica de Ziegler Nichols la cual se utiliza en un sistema de circuito cerrado y los ventiladores como una perturbación pueden verificar el correcto funcionamiento del controlador. El programa ejecutado funciona como un sistema SCADA y las pruebas muestran que el sistema es rápido, robusto e insensible a las interferencias externas, lo que es un resultado satisfactorio para los autores y revisores del proyecto, en este caso se relaciona con nuestro proyecto como parte de los procesos industriales y que buscan que el aprendizaje sea lo más real posible.



Fig. 1.- Módulo de control y medición de temperatura abiertos

(CARVAJAL YANNUZZELLI, 2018) En el trabajo de titulación denominado **Implementación de módulos didácticos básicos basados en el microcontrolador atmega164 para el laboratorio de microprocesadores de la Esfot** del Ingeniero Joel Carvajal indica que la asignatura de Microprocesadores tiene como objetivo que los estudiantes aprendan a diseñar sistemas de baja complejidad basados en un microcontrolador de estudio. Se reconocen los elementos básicos, la conexión y la arquitectura de los microcontroladores mediante la documentación técnica del fabricante, el cual el objetivo de su proyecto es renovar el laboratorio de microprocesadores, mediante la implementación de tablero didáctico básico utilizando un microcontrolador Atmega164PA y los dispositivos electrónicos más elementales al iniciar el estudio de la carrera electrónica como: resistencias, leds, pulsadores, interruptores, entre otros. Luego de revisar los temas de estudios, elementos electrónicos y la ficha técnica del microcontrolador, en donde el diseño del módulo sea amigable para los estudiantes. También permitieron que se pueda expandir si es necesario de acuerdo con los temas de estudio seleccionados, para instructores y estudiantes con la creación de guías de prácticas de laboratorio que sirven de referencia para demostrar el funcionamiento de los tableros didácticos. Además, se incluye un manual de usuario en el que se detalla: las conexiones que componen la placa didáctica, información sobre la instalación y uso del programador USB, y solución a posibles problemas; este tema de titulación se relaciona al nuestro en la creación de prácticas de laboratorios como guías para que los estudiantes puedan expandir más sus conocimientos y permitir el esparcimiento de los módulos con la conexión de diferentes elementos o componentes electrónicos.

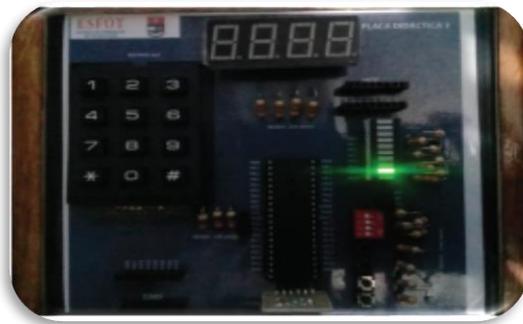


Fig. 2.- módulos didácticos básicos basados en el microcontrolador atmega164

(CRUZ SASIG, 2019) En el presente trabajo de titulación denominado **Implementación de sistema de control y supervisión HMI para relleno de tubos de rayos X** del Ingeniero Franklin Cruz se expone el desarrollo y puesta en marcha de un prototipo de máquina de control automático a fin de efectuar el relleno de tubos de rayos X con aceite dieléctrico visualiza en un HMI moderno y de bajo costo, con el propósito de ayudar al personal técnico encargado de realizar este proceso de forma manual, de esta manera reducir el tiempo de trabajo involucrado en cada etapa, desde que el tubo se encuentra vacío hasta que se llena. Aplicaron conocimientos de sistemas neumáticos e hidráulicos en la programación de Arduino Mega para que la programación de la pantalla se visualice mediante imágenes y palabras concretas según lo que esté realizando en cada momento. Los procesos se realizan de manera automática, cuenta con sensores de presión y temperatura según cada necesidad, esto ayuda al operador, evitando que tenga que permanecer en vigilia de que todo se vaya haciendo de manera correcta. La conclusión de este sistema es ayudar al técnico encargado a realizar este trabajo, de tal manera que pueda ocupar ese tiempo en cualquiera otra actividad. Este tema se relaciona al nuestro por el uso de Pantallas HMI (Interfaz Hombre Máquina) modernas y que manejen un IDE diferente a la programación de un sistema embebido ya sea este un Arduino, un raspberry o un PIC; los procesos industriales sean cual sean permiten que el estudiante visualice diferentes problemáticas en su entorno de trabajo (aunque sean ficticias) y que den una solución basada en el análisis profundo en el que se toma en cuenta cada una de sus acciones y sus posibles reacciones.

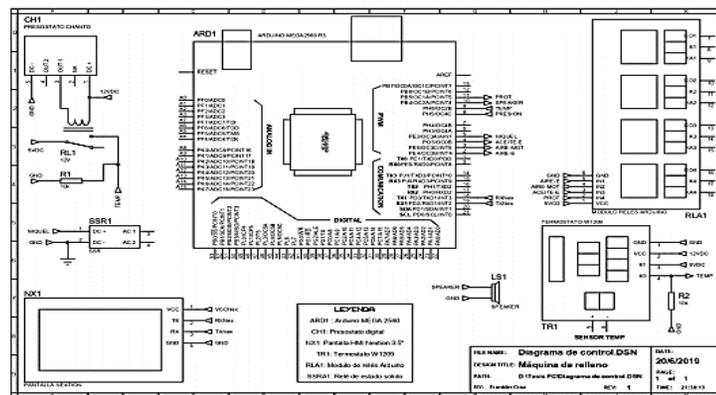


Fig. 3.- Diseño esquemático de sistema de control de tubos de rayos X

(Franco Reina & Monteseoca Paladines, 2014) Este proyecto denominado **Diseño y Construcción de cinco entrenadores didácticos con sistemas Microprocesados y desarrollo de una aplicación de control de velocidad para un motor de corriente alterna** de los Ingenieros Rafael Franco e Israel Montesdeoca surgió como necesidad de mejoramiento del nivel de enseñanza aprendizaje en la asignatura de Microprocesados de la carrera de ingeniería Electrónica de la Universidad Politécnica Salesiana sede Guayaquil, el cual lograron implementar cinco entrenadores didácticos, que favorecieran la interiorización eficiente de los conocimientos teóricos impartidos por los docentes. La metodología fue de carácter descriptivo y experimental, basada en: investigación de campo y una extensa investigación bibliográfica/experimental; por ende, la población muestra se determinó por estudiantes de la asignatura antes mencionada y de sus docentes. Es así, como el proyecto se estructuró desde el análisis de la situación problema con los parámetros pertinentes; los fundamentos teóricos para la construcción de los entrenadores didácticos; el diseño, construcción y montaje de entrenadores didácticos y aplicación de control de velocidad de motor en corriente alterna; el mantenimiento técnico correspondiente, las prácticas de laboratorio hasta el costo generado. Demostraron que, si los docentes de la asignatura utilizan los entrenadores didácticos en la aplicación teórico/práctica de sus clases basado en el plan analítico, se obtendrían beneficios inmediatos en el proceso de aprendizaje; lo cual reveló que de la correcta utilización técnica del entrenador didáctico dependió el éxito del aprendizaje. En conclusión, se ha contribuido a la innovación y prestigio institucional de la Carrera, este proyecto técnico se relaciona al nuestro porque buscamos que el aprendizaje del estudiante sea tanto experimental como técnico.



Fig. 4.- Entrenadores Didácticos

2.2. Buzzer

(Ingeniería Mecafenix, 2018) Es un transductor capaz de convertir o de transformar la energía eléctrica en sonido, es un componente pequeño pero eficiente que sirve como efecto piezoeléctrico en el cual se le aplica un voltaje y cambia ligeramente a un sonido. Este se compone de dos placas metálicas pequeñas y una de cerámica las cuales aprovechan el efecto de que solo con un clic cambia su forma pero no regresa a su estado natural hasta que le voltaje de alimentación sea retirado.

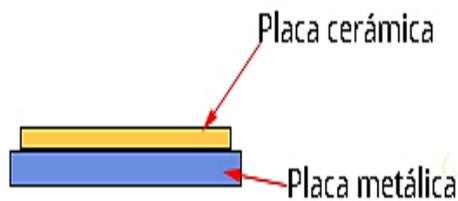


Fig. 5.- Componentes de un Buzzer

Dentro de los Buzzer piezoeléctricos se encuentran dos tipos muy comunes:

- Los que requieren de un voltaje y un oscilador externo, los cuales se denominan **Sin Osciladores** y los que cuentan con un oscilador interno que facilita el funcionamiento y que con solo aplicar el voltaje es suficiente, también son conocidos como **Con Osciladores**.

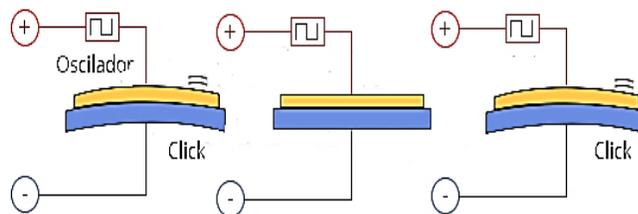


Fig. 6.- Buzzer con Oscilador Interno

(Componentes 101, 2017) Las características y especificaciones del zumbador o Buzzer son:

- Voltaje nominal de 6 voltios en corriente continua o corriente directa.

- Su voltaje de funcionamiento y de acuerdo con las protecciones para evitar su daño es de 4.8 voltios en corriente continua o corriente directa.
- La corriente nominal de este tipo de elementos es menor a los 30 microamperios.
- El tipo de sonido es de pitido continuo el cual se interrumpe en el momento que no recibe el voltaje de alimentación.
- Su frecuencia de resonancia es de aproximadamente 2300 Hz.

El diseño en 2D del Buzzer con sus dimensiones milimétricas con una tolerancia de +/- 0.5 mm se especifican en la Fig. 7:

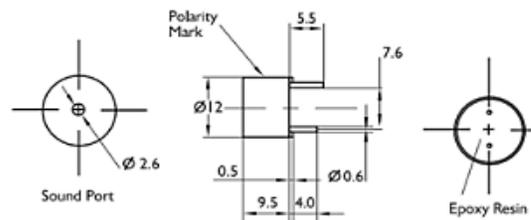


Fig. 7.- Diseño 2D de Buzzer

2.3. Relé

(Seas, 2019) Es un interruptor eléctrico el cual permite el paso de la corriente eléctrica cuando se encuentra en estado cerrado o la interrumpe cuando su estado es abierto y su accionamiento es únicamente eléctrico. Este interruptor eléctrico se compone de una bobina conectada a una corriente que cuando se conecta, produce un campo electromagnético que hace que el contacto pase con normalidad abierto a normalmente cerrado, esto se podría considerar como el funcionamiento del encendido de una lámpara o el arranque de un motor.

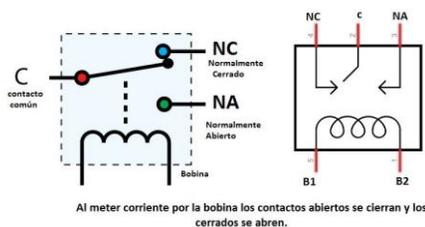


Fig. 8.- Funcionamiento de un relé

(Seas, 2019) Existen diversos tipos de relés los cuales tienen un uso específico en el campo de la electrónica:

- Los relés electromecánicos cuya variación está en el mecanismo de activación.
- Los relés de estado sólido que son fundamentales en su uso cuando existe una rutina continua de los contactos y se precisa una velocidad mayor en la conmutación.
- Los relés de corriente alterno en donde su funcionamiento principal es para gamas altas.
- Los relés temporizadores o también conocidos como acción retardada que es para temporizar su activación, más utilizados en procesos industriales.
- Los relés térmicos que fueron creados para proteger a los motores de sobrecargas y su funcionamiento es igual que los demás relés, pero con la diferencia de que trabajan con la deformación según el calor.



Fig. 9.- Relé de 5 Vdc

2.4. Led RGB

Es un led que combina los tres colores principales o primarios para formar más de 16 millones de otros colores de luz, el cual dependiendo de la tonalidad y un parámetro emite un color u otro. Las siglas RGB provienen del inglés de los colores rojo, verde y azul. Para ellos también se conoce le CMYK los cuales son del acrónimo de colores Cian, Magenta, Amarillo y Negro.

Estos leds RGB vienen representados por números comprendidos entre los valores de 0 a 255 en el cual se establece una terna ordenada para así poder representar el color que

deseamos. En la Fig. 10 se puede visualizar la composición de la terna ordenada, de acuerdo con los colores primarios y a los secundarios como es de la combinación de ellos:

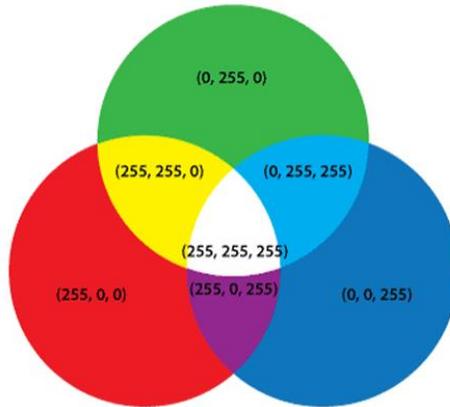


Fig. 10.- Combinación RGB

(Equipo Led Tecnología, 2019) Este tipo de leds también tiene su notación hexadecimal para su mejor composición en la programación de microcontroladores, su codificación se puede utilizar en cualquier tipo de lenguaje sea decimal o hexadecimal.

Tabla 1.- Código Hexadecimal Led RGB

COLOR	TERNA ORDENADA	CÓDIGO HEXADECIMAL
NEGRO	(0, 0, 0)	#000000
BLANCO	(255, 255, 255)	#FFFFFF
ROJO	(255, 0, 0)	#FF0000
VERDE	(0, 255, 0)	#00FF00
AZUL	(0, 0, 255)	#0000FF

Estos Leds RGB están comúnmente disponibles en tres disposiciones de los pines:

1. Un ánodo común para todos los leds
2. Un cátodo común para todos los leds
3. Cada ánodo y cátodos individuales se fijan en un total de seis pines

Su diseño más frecuente es de un ánodo común y cada uno de ellos con un cátodo para los LEDs rojo, verde y azul, resultando así cuatro pines los cuales se pueden visualizar en la Fig. 11 donde se detalla que el pin 2, es la conexión de la alimentación en un voltaje de no máximo de 5 voltios en corriente continua, visualizando su funcionamiento en el momento que se conecta el led a tierra.

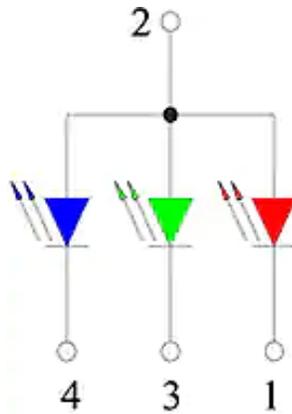


Fig. 11.- LED RGB configuración común

(Kitronik Ltd) Las dimensiones del paquete de este tipo de Led son en milímetros con una tolerancia de +/- 0.25 mm exceptuando el fabricante en cambios mínimos, en donde el pin común es el de mayor longitud de aproximadamente 25 mm, en comparación con los otros tres con una diferencia de 2 mm, el diámetro interior es de 5 mm y el exterior es de 5.8 mm que permite tener armonía y reflejar más la luz.

Su corriente máxima es de 20 mA a una temperatura de 25°C en los dispositivos ya establecidos del RGB, aunque el rango de trabajo o de operación es de -40 a 80°C, y su temperatura para soldar debe ser de 260 °C por 10 segundo para soldadura por reflujo y de 350°C por 3 segundo para soldadura manual.

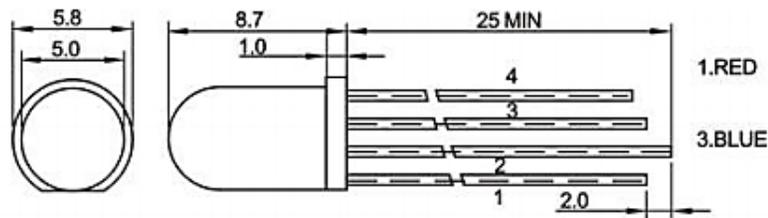


Fig. 12.- Dimensiones del Paquete Led RGB

2.5. Sensor Infrarrojo

(INFAIMON, 2018) Un sensor infrarrojo es un sensor fotoeléctrico o de fotocélula que responde al cambio en la intensidad de luz en donde uno de sus componentes llamado emisor es quien genera la luz y el componente receptor es quien lo percibe. Existen diseños diversos, pero siempre manteniendo el principio de funcionamiento; el que se utiliza es el diseño de herradura que permitirá el conteo de la velocidad de un motor en corriente continua. Su corriente de LED en directo no pasa de los 50 miliamperios y este infrarrojo es de tipo transmisivo el cual se podría considerar un voltaje en el colector emisor para su funcionamiento.

(FAIRCHILD, 2001) El sensor infrarrojo tiene características comunes entre sus fabricantes:

- El voltaje en inverso es de 6 voltios.
- El voltaje colector emisor es de 35 voltios
- Tiene una respuesta de 50 microsegundos
- La corriente del colector es de 20 miliamperios.
- Su salida es fototransistora y su tamaño es compacto.
- Las aplicaciones de este sensor es la detección de presencia o ausencia de objetos, detección de alguna posición o medición de revoluciones y como se mencionó anteriormente en este proyecto se realiza la medición de las revoluciones de un motor DC.
- Este sensor opera en un rango de temperatura de -55 grados Celsius a 100 grados Celsius.

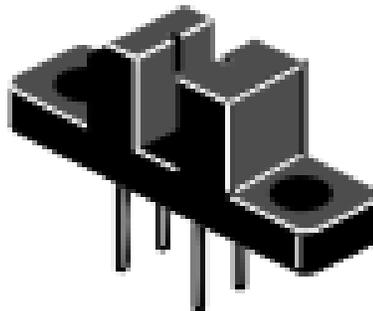


Fig. 13.- Sensor Infrarrojo de Herradura

El sensor infrarrojo que utilizaremos es el FC 03, permitirá obtener los pulso de su encoder óptico teniendo en cuenta que su salida de TTL en encendido siempre será un bloqueo, mientras está desbloqueado su salida no se activa.

2.6. LM324

(TEXAS INSTRUMENTS, 2015) Circuito integrado LM324 son cuatro amplificadores operacionales con verdaderas entradas diferenciales. Consta de cuatro amplificadores operacionales de alta ganancia, diseñados para funcionar a través de una fuente de alimentación simple. Sin embargo, también pueden funcionar con fuentes de alimentación dobles. Tiene ventajas sobre los amplificadores operacionales tradicionales en aplicaciones de suministro único, y puede manejar voltajes de suministro de 3 V a 32 V. El consumo de energía es muy bajo (aproximadamente 1/5 del consumo de energía tradicional del LM741).

Se puede utilizar en las siguientes aplicaciones:

- módulos de ganancia de CC,
- amplificadores
- y cualquier circuito típico con amplificadores operacionales.

Ahora, el uso de una fuente de alimentación simple puede implementar más fácilmente estas funciones. Por ejemplo, LM324 puede funcionar directamente a 5 V (un voltaje comúnmente utilizado en la electrónica digital) sin tener otra fuente de alimentación dual ± 15 V CC.

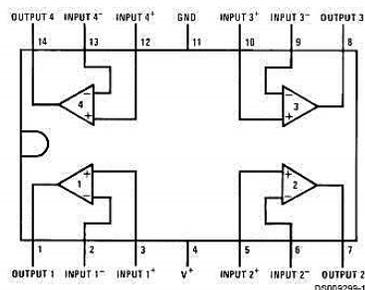


Fig. 14.- Diagrama de conexión

(ELECTRÓNICA PLUG AND PLAY, 2019) Entre las características más representativas; el ancho de banda es de 1MHz, trabaja en temperaturas de 0 hasta 70°C, su uso es variado y las aplicaciones más habituales son en los multímetros digitales. Las

interfaces transductores, los sistemas de adquisición de datos. Los procesos de control industriales los Home Theaters.

Se compone de 14 pines los cuales permiten en su conexión tener una baja corriente de polarización de entrada de 45 nA compensado con la temperatura en donde el bajo offset de voltaje de entrada es de 2mV y de corriente es de 5 nA.

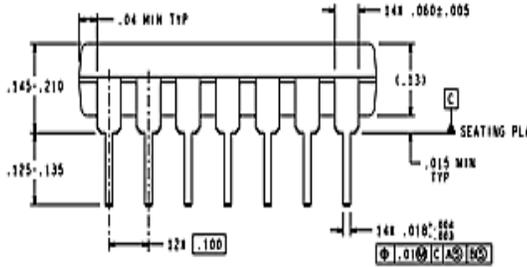


Fig. 15.- Vista Lateral LM324

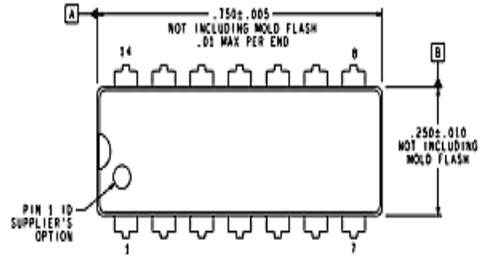


Fig. 16.- Vista Superior LM324

2.7. L293D

(Texas Instruments, 2016) El circuito integrado denominado L293D, es un controlador de motores que está construido en base a 4 mitades del puente H, el cual permite controlar distintas cantidades de motores en corriente continua ya sean unidireccionales o bidireccionales. Las potencias que maneja se consideran; potencias media en especial motores pequeños y cargas inductivas con una capacidad de corriente de 600 miliamperios con una tensión que varía entre los 4.5 voltios a los 36 voltios en corriente continua.

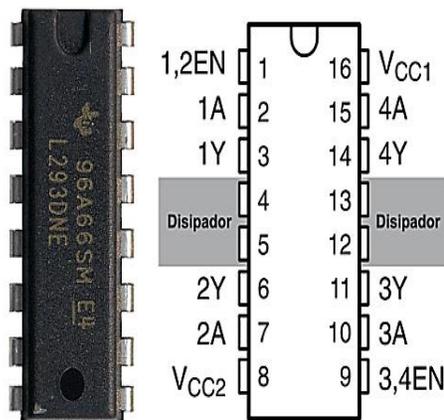


Fig. 17.- Integrado L293D

Este integrado que está formado por 4 medios puentes H, también permite formar dos puentes H completos para el manejo de 2 motores bidireccionales, se puede utilizar según se requiera de manera independiente, sus salidas tienen el diseño que permite el manejo de cargas inductivas tales como relés, solenoides y más, así que tiene en su composición diodos de protección para evitar la contracorriente de las cargas inductivas.

(Texas Instruments, 2016) Su estructura es de la configuración de transistores Darlington que conducen su salida a tierra, los cuales se habilitan por medio de una señal TTL y en sus entradas le permite controlar con facilidad el circuito, condicionalmente porque permite la regulación de velocidad de los motores por medio de una modulación de ancho de pulso, dando así una salida de flancos positivos de acuerdo con los sentidos o el sentido establecido de los motores o el motor como se puede visualizar en la Fig. 18.

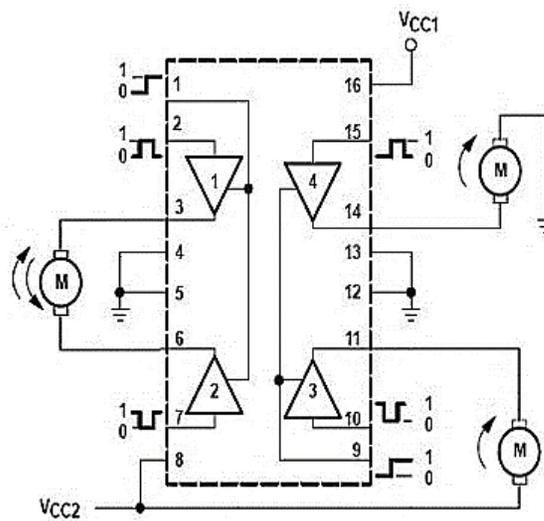


Fig. 18.- Conexionado para un motor con giro en ambos sentidos (lado izquierdo) y con motores con giro en sentido único en dos salidas (lado derecho)

(Robots Argentina, 2019) Los pines centrales del integrado están diseñados para proporcionar un contacto térmico con el gatillo o disipador, de modo que se pueda obtener la máxima potencia al manipular el integrado. En la Fig. 19 se muestra la distribución de pines afectada por esta disipación, y para ello se tiene una hoja de datos la cual aporta con curvas que permite una variación de los tamaños del disipador según la potencia a manejar.

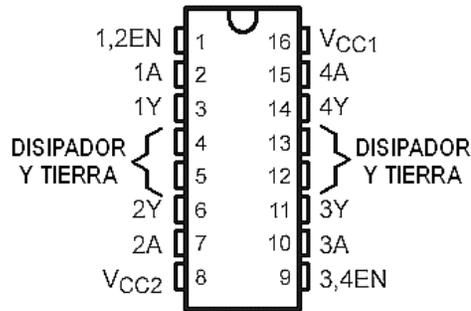


Fig. 19.- Distribución de pines de L293D

2.8. Motor DC

(MINEBEA MOTOR MANUFACTURING CORPORATION , 2010) Un motor DC o también conocido como motor en corriente continua o corriente directa, es una de las máquinas que convierte la energía eléctrica en energía mecánica provocando el movimiento rotatorio, todo por la acción que genera el campo electromagnético en el rotor y a su vez interactúa con el imán del estator que genera el campo magnético. Estas acciones se derivan en un rechazo entre los polos del imán del estator y del rotor, creando un par de fuerza donde el rotor gira en un sentido de forma permanente y estable. El sentido de giro de ese motor depende mucho de la polaridad que se le dé a la fuente principal ya sea una batería o una fuente de poder.



Fig. 20.- Motor DC

(PrestaShop, 2021) En este tipo de motores en sus revoluciones dependen del voltaje de entrada y para controlar su velocidad se debe realizar mediante la programación de a rotación en ángulos definidos por el usuario. Tiene un voltaje nominal de operación de 5 voltios en corriente continua y su rango de operación está entre 1 a 7 voltios, la velocidad de operación debe ser menor a 16000 rpm la cual se obtiene una corriente de carga de

aproximadamente 270 mA, pero la corriente de arranque podría llegar hasta las 1.3 amperios.

2.9. MOC3021

(ISOCOM COMPONENTS, 2008) Este tipo de integrado se los conoce como optoacopladores de fase aleatoria con fototriac, que le permite crear un tipo de aislamiento óptico entre la etapa de control y de potencia del circuito en su diseño, teniendo la posibilidad de trabajar con diferentes niveles de voltaje en dichas etapas. Normalmente su composición es de un diodo emisor de arseniuro de galio acoplado a la luz infrarroja del interruptor bilateral de silicio que activa el funcionamiento del Triac.

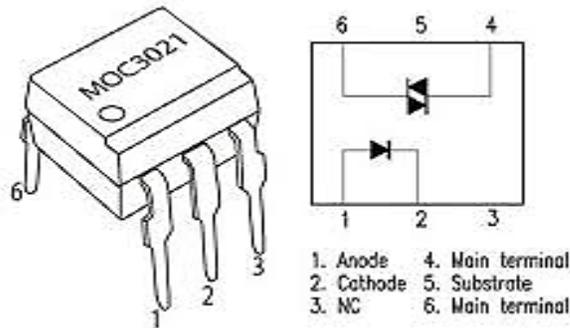


Fig. 21.- Descripción de Pines del MOC3021

Sus pines están establecidos en la Fig. 21 en la cual se describen de manera visual los pines de dicho integrado. Este tipo de integrado tiene características que permiten una mejor manipulación del elemento, entre ellas están:

- Su voltaje de aislamiento está entre los 7500 Vpico.
- Su encapsulado es de 6 pines.
- Tiene una corriente máxima del emisor de 60 mA.
- El voltaje inverso del emisor es de 3 Voltios.
- La corriente máxima del receptor es de 1 amperio.
- El rango de temperatura en el cual su funcionamiento está permitido es de -40 °C a los 85 °C.
- La corriente del disipador LED está entre los 15 mA.

(ISOCOM COMPONENTS, 2008) Las aplicaciones de este arquetipo de integrados son en los controles industriales, el control de válvulas solenoide, en los interruptores de alimentación de corriente alterna estático, en los relés de estado sólido, en el control de motores, etc....

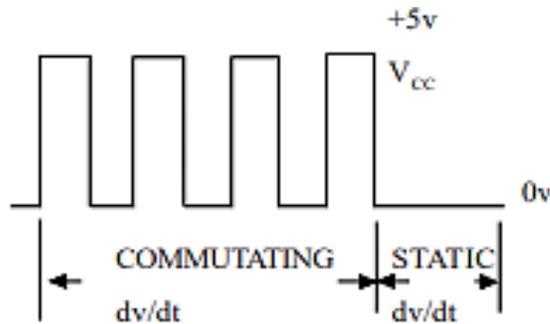


Fig. 22.- Gráfica de Funcionamiento del MOC3021

2.10. Triac BT16

(Semiconductor Components Industries, 2012) El Triac es un dispositivo semiconductor de tres terminales que se utiliza para controlar la corriente promedio que fluye hacia la carga. Su característica especial es, que conduce bidireccionalmente y puede invertir el voltaje o reducir la corriente por debajo de este valor. Para detener y mantener el Triac se puede disparar independientemente de la polarización de la puerta, es decir, disparado por una corriente de puerta positiva o negativa. Cuando se enciende el Triac, la trayectoria del flujo de corriente de resistencia de un terminal al otro terminal es muy baja, y su dirección de flujo depende de la polaridad del voltaje externo aplicado.

(SGS-THOMSON MicroElectronics, 1995) Este tipo de integrado es adecuado para la conmutación de corriente alterna de alimentación de red de usos generales, su uso puede ser en función de encendido, apagado o de ángulo de fase en aplicaciones, tales como los relés estáticos, reguladores de intensidad de luz y controladores de velocidad de motores. Entre las características o especificaciones las más relevantes son:

- El diseño es de 3 pines los cuales se divide en ángulo de fase (gate), ánodo 1 y ánodo 2.
- El estilo de su montaje es en Through Hole

- La temperatura de operación máxima es de 125 °C
- La temperatura mínima de funcionamiento o de operaciones de -40 °C
- La corriente de la compuerta máxima es de 100 mA.
- La corriente rms en funcionamiento o cuando se encuentra encendido es de 16 A.
- La potencia de pico en la puerta es de 1 W.
- La activación del funcionamiento en voltaje es de 1.55 voltios.

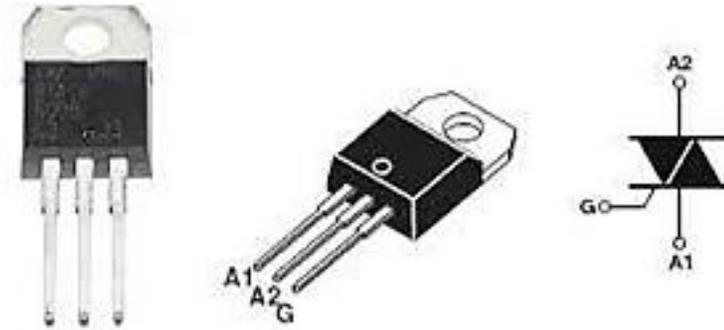


Fig. 23.- Triac BT 16

2.11. LM35

(TEXAS INSTRUMENTS, 2017) LM35 es un circuito de sensor electrónico que puede medir la temperatura. Su salida es analógica, es decir, suministra un voltaje proporcional a la temperatura. El rango de temperatura del sensor es de -55 °C a 150 °C. Su popularidad se debe a la fácil medición de la temperatura. Ni siquiera necesita un microprocesador o microcontrolador para medir la temperatura. Dado que el sensor LM35 es un sensor analógico, es suficiente medir el voltaje en el extremo de salida del sensor con un multímetro. Para convertir voltaje en temperatura, LM35 proporciona 10 mV por grado Celsius. Cabe destacar también que el sensor se puede utilizar sin offset, es decir, si medimos 20mV en la salida, medirá 2 °C.

Este tipo de integrados se usan ampliamente en dispositivos de temperatura, humedad, de humo, de luz; este es uno de los sensores más populares y utilizados en la electrónica. No todos los sensores están calibrados en grados Celsius, en este caso, este sensor si está preparado, con una variación de 1 °C.

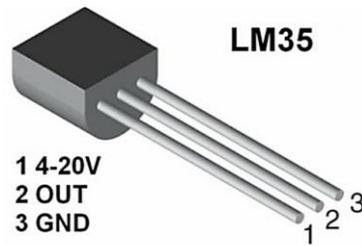


Fig. 24.- Sensor LM35

(Marmolejo, 2017) Este circuito no necesita circuitería externa para calibrarlo, pues su diseño permite tener una calibración directa y sin complicaciones. Este integrado tiene una variedad de encapsulados y todo es con respecto a la variación de temperatura.

- LM35-LM35A: Lectura de temperatura entre -55 a $+150^{\circ}\text{C}$.
- LM35C-LM35CA: Lectura de temperatura entre -40 a $+110^{\circ}\text{C}$.
- LM35D: Lectura de temperatura entre 0 a 100°C .

En este integrado se encuentra un buen equilibrio en la parte de sensibilidad, en los rangos de valores, en la precisión, en el tiempo de respuesta y en el valor nulo de referencia para considerar correcciones de medida y cálculos teóricos.

2.12. Pic 18F8720

(MicroChip, 2015) El Pic 18F8720 es un circuito integrado programable que contiene los componentes que realizan y controlan tareas el cual se denomina microcontrolador, en donde se puede planificar la manera de su funcionamiento y así adaptarlo a nuestras necesidades, es decir que el Pic es capaz de cambiar el comportamiento en función a las órdenes que se compartieron en su memoria inicialmente. Éste integrado está compuesto de varios bloques funcionales que cumple con una tarea específica, por lo tanto, es diferente a un microprocesador que depende de un circuito o algunos adicionales para su funcionamiento, mientras que el PIC está compuesta de:

- Contiene una memoria de solo lectura, se la conoce en el mercado como memoria ROM
- Contiene una memoria de acceso aleatorio, es conocida como memoria RAM y es el complemento de la memoria ROM.

- Tiene puertos o líneas de entradas y salidas que normalmente se las conoce como I/O.
- Existe una lógica de control que coordina la interacción entre los demás bloques funcionales.

(Ingeniería Mecafenix, 2017) Este tipo de integrado es completo en su estructura y su programación utiliza, dos tipos de arquitecturas internas las cuales permiten definir su funcionamiento y son; la de Von Neumann y de Harvard. La arquitectura de Von Neumann, es por donde circulan los datos y las instrucciones por el mismo bus y eso es porque están guardados en la misma memoria, como ventaja de esta arquitectura es el ahorro de líneas de entradas y salidas, pero con la desventaja de que exista disminución en la velocidad con la que se realizan los procesos. En la arquitectura de Harvard los datos e instrucciones tienen su propia memoria específica dando como resultado el uso de buses independientes, como ventaja esto logra trabajar con las memorias simultáneamente, pero con la consecuencia que se obtiene mucha velocidad en la ejecución de los programas sin poder corregir algo a menos de que el programa finalice.

(MicroChip, 2015) La gama de PIC que vamos a utilizar es la gama mejorada, utilizaremos un Pic 18F8720 el cual tiene como características:

- En comparación con las otras familias, este Pic tiene una memoria de programación de 120Kbytes, permite la extensión de instrucciones con un rango de 65536.
- Su memoria de datos es de 1024 bytes.
- Tiene 68 pines de entradas/salidas, el programador puede decir cuáles de ellas puede utilizar.
- De esos 68 pines que tiene el Pic, 16 son canales o pines analógicos los cuales permiten al programador expandir sus conocimientos en procesos analógicos.
- Trabaja con la comunicación SPI y la I2C lo cual permite expandir la comunicación con otros Pic de familias diferentes.
- Está compuesto de 80 pines los cuales se dividen en 7 de uso general (timers internos) o conexión externa con otros Pic, 68 entradas y salidas digitales y analógicas y 5 pines de PWM.

El Pic que usaremos no se encuentra en Through-hole sino más bien como montaje superficial, el cual permite que la manipulación con el programador sea nula, en especial en cambio constante en su programación. Este integrado como se visualiza en la fig. 25 permite el uso de programación directa entre el computador y la placa de diseño, evitando el contacto y su desgaste por mala manipulación o algo parecido; su unión se realiza en caliente a través de una técnica llamada sellado por horno en donde los 80 pines se adhieren a la PCB que se diseñó.



Fig. 25.- Pic 18F8720 en Montaje Superficial

Todo microcontrolador o Pic tiene una vida útil, en este caso de acuerdo con su uso o manejo puede durar hasta unos 10 años, este tipo de integrados puede evolucionar o mejorar para las futuras generaciones, permite la expansión y la solidificación de lo que deseamos construir y gracias a esto se puede concretar.

2.13. Teclado Matricial de 4X3

(I.Crisologo, 2011) El teclado matricial es ensamblado tal como lo dice su nombre en forma de matriz en especial para mejor entendimiento, al decir que utilizaremos un teclado matricial de 4x3 estamos indicando que dicha matriz se compone de 3 columnas y 4 filas o también conocidas como renglones. Cuando se oprime una tecla se realiza la conexión entre una columna y un renglón, en su interior ya está configurado para establecer su funcionamiento.

Como se visualiza en la Fig. 26 la configuración de su conexión por tecla es la denominada por matrices, utilizando para ello un decodificador de teclado matricial en el cual permite la conexión de los renglones y de las columnas. Las tareas básicas que realiza

un teclado matricial es la de supervisar si se presionó o no una tecla, si la respuesta es no, su funcionamiento queda ahí, pero en el caso de ser si, el microcontrolador realiza la tarea de preguntar que tecla fue la que se presionó y rastrea los renglones para así encontrar la columna de la tecla oprimida (esto se programa mediante el código del sistema de codificación de caracteres alfanuméricos más conocido como ASCII).

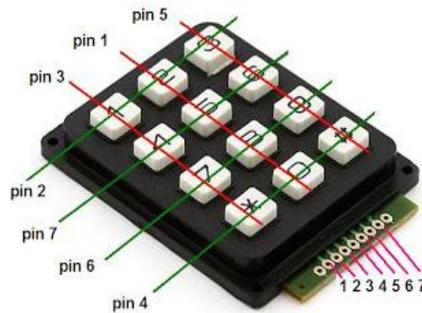


Fig. 26.- Teclado Matricial 4x3

(Llamas, 2016) Este tipo de teclado en los microcontroladores ya tiene su puerto definido, en el cual se establecerá su funcionamiento a través de las librerías proporcionadas por los creadores de dicho microcontrolador que se utiliza. Las 12 teclas solo requieren 6 pines del microcontrolador, en lugar de los 12 pines necesarios para conectar 12 teclas independientes. Para leer qué tecla se presionó, debe usar tecnología de escaneo, y no solo leer los pines del microcontrolador. Entre las características más frecuentes encontramos:

- Trabajo en un voltaje máximo de operación de 24 voltios en corriente continua
- Su corriente máxima de operación es de 30 miliamperios.
- Es mayor resistente al agua.
- La temperatura en la que trabaja es de 0 a 50 grados Celsius.

2.14. LCD 4X20

(Vishay, 2021) El LCD es una pantalla alfanumérica monocromática (un solo color en este caso, negro) de cristal líquido que utiliza las propiedades de modulación de la luz de los cristales cuando no emiten luz directamente. Este elemento es un dispositivo que permite la visualización de caracteres alfanuméricos y otros símbolos en un formato de 20

caracteres por línea en 4 líneas, es decir que está compuesta de 4 filas y 20 columnas en la visualización. Los LCD tiene infinitudes de características según su diseñador, entre las peculiaridades más comunes están:

- La alimentación es de 5 voltios en corriente continua, aunque existe LCD que solo necesitan 3 voltios.
- El voltaje máximo de operación es de 5.7 voltios comparado con los otros diseños.
- La corriente en directo cuando se aplica los 5 voltios es de 10 miliamperios, teniendo en cuenta que su voltaje máximo de operación.
- De acuerdo con el diseño de este tipo de pantallas, existe la condición por la temperatura normal en la que trabajan con un voltaje definido, las cuales son:
 - A una temperatura de -20 grados Celsius su valor de operación máxima es de 5.7 voltios.
 - A una temperatura de 0 grados Celsius su valor de operación máxima es de 5.2 voltios.
 - A una temperatura de 25 grados Celsius su valor de operación máxima es de 4.7 voltios.
 - A una temperatura de 50 grados Celsius su valor de operación máxima es de 4.5 voltios.
 - A una temperatura de 70 grados Celsius su valor de operación máxima es de 4.3 voltios.



Fig. 27.- LCD 20x4

Es decir que mientras la temperatura sube de voltaje de operación es inversamente proporcional a ello, teniendo en cuenta que el diseño de estos LCD se los construye o fábrica de manera estándar, pero respetando las condiciones climáticas de cada país. Este tipo de elemento electrónico tiene 16 pines que van conectados al microcontrolador en un puerto ya definido para su manejo tal como se muestra en la Fig. 28 recordando que no todos los pines o las líneas van conectadas, este tipo de visualizadores tiene su manipulación en el contraste el cual se realiza ubicando un potenciómetro en la que se puede subir y bajar la iluminación.

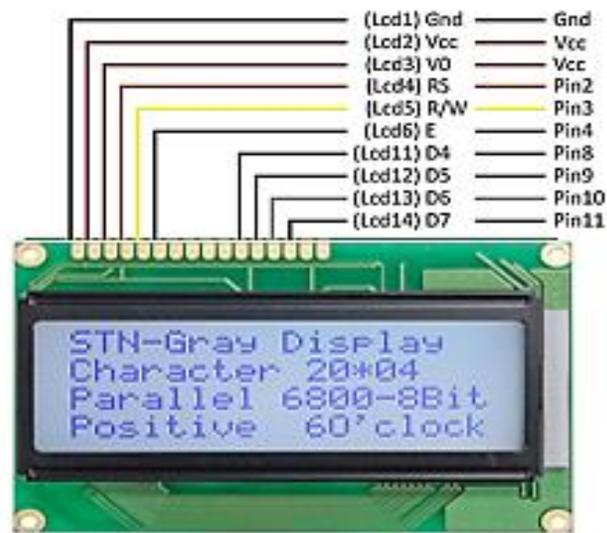


Fig. 28.- Conexión de LCD 20x4

2.15. Pantalla NEXTION

(Nextion Instruments, 2020) Nextion es una interfaz hombre-máquina (HMI) que combina un procesador integrado y una pantalla táctil de memoria con el software Nextion Editor para el desarrollo de proyectos HMI GUI. El software Nextion Editor le permite desarrollar rápidamente la interfaz gráfica de usuario HMI arrastrando y soltando componentes (gráficos, texto, botones, controles deslizantes, etc.) e instrucciones ASCII basadas en texto para codificar, como componentes en el costado de la pantalla para interactuar. La pantalla de Nextion HMI se conecta a la MCU periférica a través de TTL Serial para proporcionar notificaciones de eventos para que responda la MCU periférica.

La MCU periférica puede actualizar fácilmente el progreso y el estado en la pantalla Nextion, usando instrucciones ASCII simples basadas en texto.

El modelo que se utiliza en este proyecto es una Nextion NX1060P101-011R-I el cual es de pantalla táctil resistiva de 10.1 pulgadas sin carcasa, entre sus especificaciones está que su resolución es de 1024 x 600 pixeles, la iluminación desde el fondo es LED, el peso es de 505 gramos, el brillo está en intervalos de ajuste de 1% que se puede mover desde 0% a 100%, la vida útil de a luz de fondo es mayor a 30000 horas y puede recibir más de un millón de toques; esta información es de la pantalla Nextion como se visualiza en la Fig. 29.



Fig. 29.- Pantalla Nextion

Entre sus características electrónica están:

- La tensión de funcionamiento está en el intervalo o rango de 4.75 voltios a 6.5 voltios en corriente continua.
- El valor de funcionamiento de voltaje es de 5 voltios recomendados con una corriente de 2 amperios en corriente directa.
- Su corriente en funcionamiento depende del voltaje y del brillo en la pantalla
 - Si el brillo está al 100% y su voltaje está en 5 voltios su corriente 800 mA.

- Si el brillo es nulo y el voltaje está en 5 voltios su corriente es de 170 mA.
- La temperatura de trabajo está entre el intervalo de -20 °C a 70 °C.
- El puerto TTL está en el intervalo de 3.3 voltios a 5 voltios y contiene 4 pines de 2.54 milímetros entre ellos.
- Admite tarjeta microSD de máximo de 32 Gigabytes.

(Nextion Instruments, 2020) Nextion tiene su propio software de creación de interfaz para el dispositivo HMI el cual se conoce como GUI en donde se puede crear un sin número de simulaciones, este editor pasó un proceso exhaustivo en su creación y presentación para evitar fallas o errores en su desarrollo y es compatible con el sistema operativo Windows desde el XP o superior. En este editor se puede configurar el estilo de tema, de paneles y muchas cosas más, en su interfaz principal se puede editar subir imágenes y bloques, permite realizar una simulación de depuración y conexión a herramientas externas.

Entre lo que se puede realizar en este IDE está:

- El estilo que puede configurarse entre un temático azul o negro.
- Se puede ajustar los paneles según su tamaño como en su ubicación.
- Puede editar configuraciones básicas como el diseño de la venta, abrir algún archivo, eliminar registros actuales, no válidos o todos; permite el rastreo de los proyectos creados.
- El interfaz principal está compuesto de varias funciones las cuales se pueden visualizar en la Fig. 30 y su detalle a continuación (Nextion Instruments, 2020):
 - Barra de título

La barra de título contiene la ruta y el nombre de archivo del proyecto HMI cuando se carga un proyecto HMI.
 - Menú principal

El usuario puede tener varios menús en el cual puede crear, importar, subir, buscar, configurar y si es necesario solicitar ayuda en la creación de elementos en la herramienta de GUI de Nextion.

- Barras de herramientas
El usuario tiene los iconos en esta barra desde donde podrá abrir, guardar, depurar y hasta compilar sus proyectos además de contar con los iconos básicos de cualquier software en la actualidad como pegar, copiar, cortar, eliminar, deshacer, rehacer y hasta la configuración de contraseña de sus proyectos.
- Panel de página
Cada proyecto HMI debe contar al menos con una página, es por ello, el panel de página permite realizar las funciones de agregar, eliminar, copiar, insertar, o mover páginas en nuestro proyecto si se desea aumentar o disminuir.
- Panel de la caja de herramientas
Este panel es donde se puede agregar texto, números, botones, imágenes y demás en la página creada.
- Panel de atributos
Contiene una lista de componentes incluidos en la página de diseño actual.
- Panel de recursos de imagen
Este panel permite la importación directa de las imágenes al proyecto HMI que estamos diseñando.
- Panel de recursos de fuentes
Este recurso es importante porque todo depende de la visualización y armonía que exista para el programador, su asignación es a través de códigos establecidos y creados por el fabricante.
- Panel de recursos de audio
Este recurso permite importar los audios en el formato establecido por el fabricante (actualmente permite la importación de todos los formatos de audio).

- Panel de recursos de Gmov
Permite crear animaciones con los componentes y agregarlos a la página de nuestro proyecto HMI.
- Panel de recursos de video
Permite convertir videos y agregarlos a nuestro proyecto.
- Lienzo de diseño (componentes visuales)
Este es el espacio principal para el diseño de los componentes visuales y táctiles del proyecto en las páginas del HMI.
- Componentes no visuales
Los componentes no visuales de la página (variables, temporizadores, audio y FileStream) se enumeran en el área debajo del lienzo de diseño. Si no se utilizan componentes no visuales en la página, esta área no se muestra. Solo puede haber un máximo de 250 componentes (visuales y no visuales) por página.
- Código de evento de usuario
El código de evento del usuario puede contener cualquier declaración de Nextion válida. Esta sección no le enseñará conocimientos de programación, pero proporciona una descripción general rápida de los diversos eventos en los que puede insertar el código de usuario. El código de evento siempre es local a la página más no global.
- Salida
Contiene los detalles sobre el proceso de compilación cuando se selecciona la depuración, compilación y la carga.

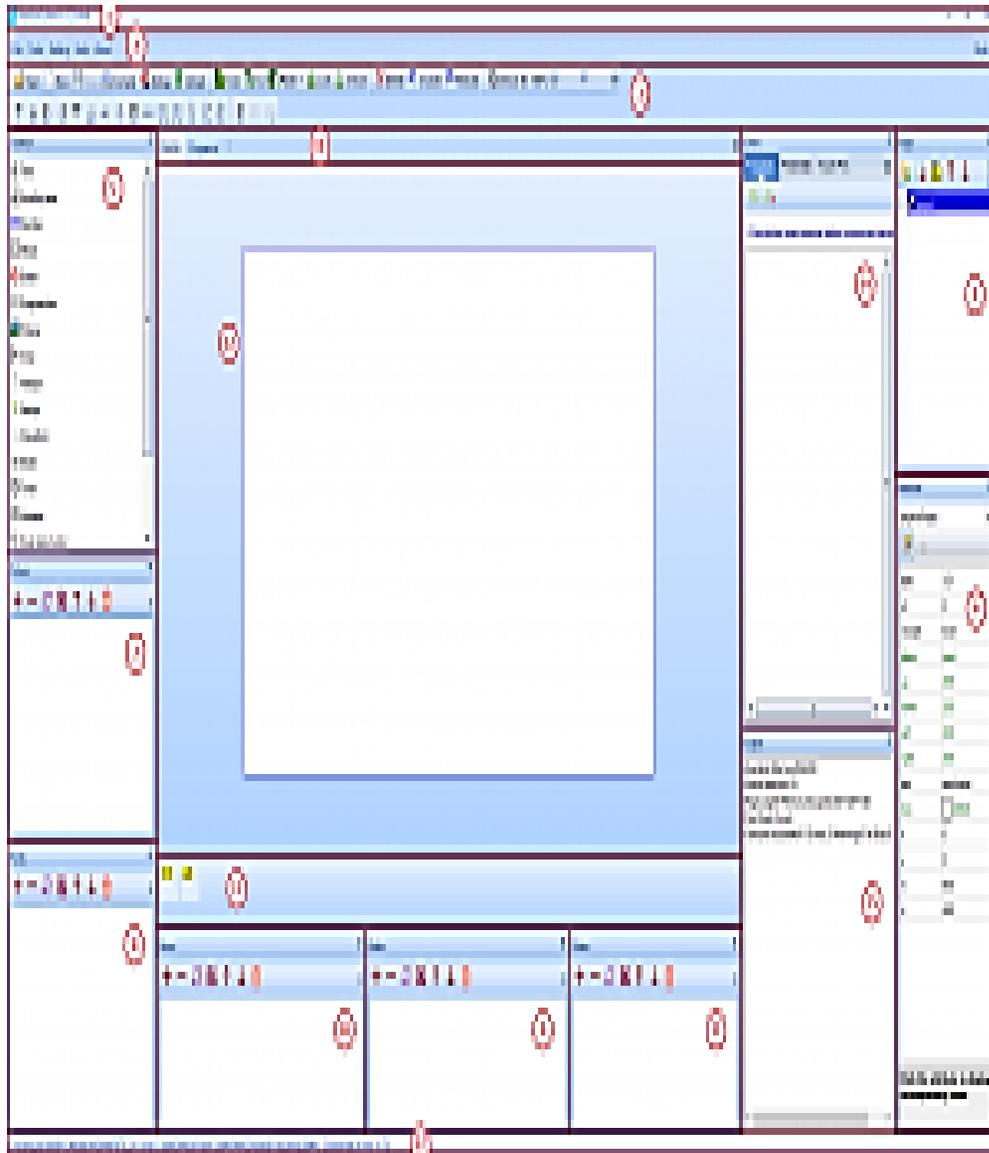


Fig. 30.- Interfaz principal Nextion Editor

- Existe un simulador para poder depurar errores del diseño.
- Finalizado el diseño se sube al dispositivo para visualizar, su carga es directa o con una tarjeta microSD.

2.16. DS1307

(Maxim Integrated, 2015) El módulo DS1307 se lo conoce como un reloj en tiempo real el cual funciona de manera decimal codificado en binario es decir BCD, es de potencia baja de un nivel de 56 bytes y la transferencia es a través de la comunicación I2C la cual es de

bus bidireccionales. Este tipo de módulo proporciona los segundos minutos, horas, día, mes y año; su funcionamiento es de acuerdo con el formato del tiempo ya sea 24 horas o 12. El módulo tiene incluido un circuito que le permite detectar fallas de energía y automáticamente cambia a su suministro de respaldo por precaución, es así como opera de manera normal mientras el circuito externo se encuentra apagado.

Trabaja en diferentes rangos o condiciones de temperatura, por tal motivo, los fabricantes especifican en la hoja técnica, de acuerdo con estos modelos el voltaje de operación es de 5.5 voltios en corriente continua, el cual permite que la variación de algún elemento externo conectado a él no ocasione una caída de tensión. Su funcionamiento es digital, solo recibe valores lógicos (0 o 1), la fabricación está compuesta de 8 pines los cuales se pueden visualizar en la Fig. 31.

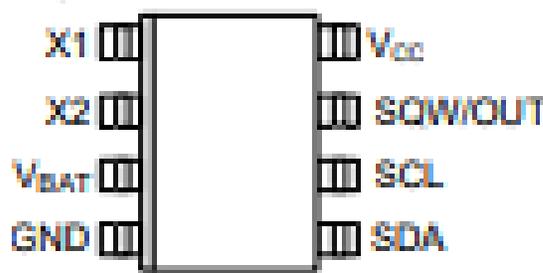


Fig. 31.- Pines del módulo DS1307

2.17. 24LC512

(MicroChip, 2015) El 24LC5112 es una memoria programable de solo lectura de borrrable eléctrica de 512 Kb, con la capacidad de funcionar en un rango o intervalo de voltaje de los 2.5 voltios a los 5.5 voltios, este integrado ha sido desarrollado para aplicaciones de bajo consumo como las comunicaciones personales y para la adquisición de datos, su capacidad para realizar lecturas aleatorias y las secuenciales, hasta un límite de 512K, las líneas de direcciones permite la conexión en el mismo bus de hasta ocho dispositivos sin inconvenientes, el espacio de direcciones es de hasta 4 megabits; su diseño contiene 8 pines los cuales permiten que su estructura sea más estándar y estética como se visualiza en la Fig. 32.

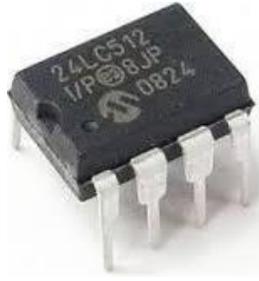


Fig. 32.- Integrado 24LC512

Entre las características que tiene este integrado está que su tiempo máximo de escritura por página es de 5 milisegundos, su programación inicial se puede editar, las lecturas de corriente están en 400 microamperios y su corriente en reposo es de 1 microamperio; su conexión puede ser en cascada la cual permitirá la conexión de hasta 8 dispositivos teniendo una frecuencia máxima de 400 KHz.

Capítulo 3: Diseño y Construcción de Módulos Didácticos

En este capítulo se realiza la creación de los módulos desde el diseño, la modelación, la implementación y pruebas para la validación de los requisitos funcionales basados en los objetivos específicos, los cuales fueron parte primordial para su desarrollo.

3.1. Consideraciones iniciales del diseño

Para iniciar el diseño y desarrollo de los módulos didácticos, se realiza una descripción general del proyecto técnico, indicando sus características principales y funciones, proporcionando una información global y específica de sí mismo, tanto de la placa o tarjeta madre como de sus complementos.

3.1.1. Funciones de cada tarjeta

La función que desempeña cada una de las tarjetas es la de experimentar un proceso industrial básico y el entrenamiento con las pantallas HMI Nextion, mediante los puertos de entrada y salidas del microcontrolador.

3.1.2. Descripción del Módulo principal

Se ha propuesto como caso este trabajo, el diseño de módulos didácticos para el entrenamiento en la parte práctica de la materia de Sistemas Microprocesados, que tendrá como finalidad la expansión del conocimiento y el uso en diferentes materias, dando como resultado la aplicación del conocimiento adquirido previo al conocimiento nuevo en materias afines. Otra característica de estos módulos es poder implementarlo con elementos de bajo costo con el fin de que sea accesible a la comunidad universitaria, a continuación se describe el funcionamiento general de los módulos.

El módulo se compone de una tarjeta madre, de tres plantas que se conectan de manera independiente a la tarjeta madre, una Pantalla HMI Nextion controlada directamente de la tarjeta madre y dos protoboard para circuitos o diseños externos propuestos y no propuestos en este proyecto. Estas placas serán de entrenamiento en donde se podrá visualizar o experimentar la materia de Sistemas Microprocesados

desde el punto de vista práctico, donde los estudiantes deberán ejecutar la programación establecida en la guía práctica y podrán experimentar con diseños de su autoría.

Se conecta a una fuente de 120 voltios y dicho valor entra a una fuente de conmutación, la cual convertirá el voltaje de entrada en 5 voltios DC (corriente continua o corriente directa) para que así la tarjeta madre y sus demás complementos funcionen correctamente (considerar que no existe pérdida ni caída de potencial ya que el diseño de las mismas se encuentra considerado cada uno de ellos), una vez energizada la tarjeta madre brinda la energía a la pantalla HMI, al protoboard y a las plantas adicional (la conexión de estas plantas es individualizada)

Toda la información que se envíe de manera externa se recibirá por medio del microcontrolador quien envía a los elementos lo que deben hacer, su funcionamiento será considerando al HMI (Interfaz Hombre-Máquina) como parte principal de lo que se realiza y también considerándolo como receptor, por ello se han establecido prácticas, en donde este aparato electrónico será el emisor y el microcontrolador el receptor y así mostrar su funcionamiento externo, luego de ello los roles cambian ya que la pantalla será la receptora y el microcontrolador será el emisor mediante los elementos externos como potenciómetros, switch y demás, demostrando así su funcionamiento interno en donde los estudiantes podrán visualizar la parte de la configuración de entradas y salida de puertos.

En un diagrama de flujo se visualiza el funcionamiento del sistema completo para diferenciar si su manejo de puertos es externo o interno, adicional a ello se maneja interrupciones, las cuales permitirán un manejo primordial en la programación del proceso industrial tal como se visualiza en la Fig. 33, que se desea realizar utilizando los softwares de cada uno de estos componentes o elementos para la pantalla NEXTION, se utiliza lo que es el IDE Guide de la misma compañía que lo proporciona para que se pueda trabajar en dicho interfaz gráfico; para la programación del microcontrolador utilizaremos el PIC C Compiler el cual se ha venido utilizando en

nuestro aprendizaje como estudiantes en las versiones permitidas por los creadores de dichos programas.

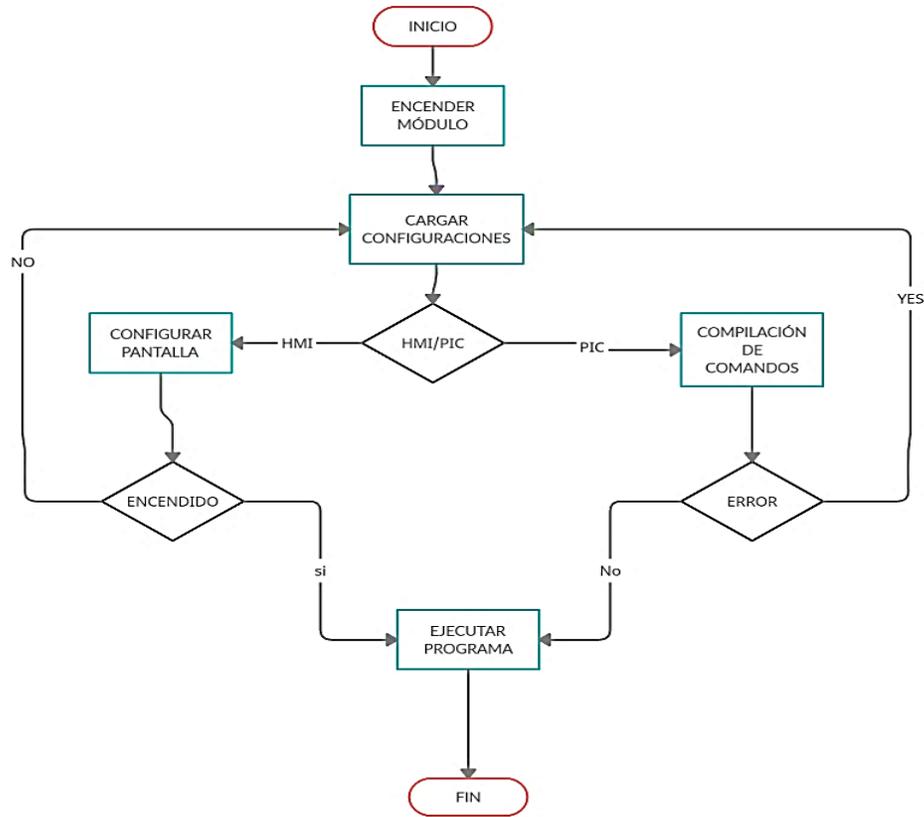


Fig. 33.- Flujograma de funcionamiento
Fuente: Autores

3.1.3. Características de las plantas de trabajo

En esta sección se especifican las características principales de funcionamiento de cada una de las plantas de trabajo, para ello la dividiremos de acuerdo con su control:

- **Planta de control de luminosidad**

La planta de control de luminosidad tiene como características:

- El voltaje nominal de operación es de 5 voltios
- Se puede trabajar con un rango de voltaje de los 3.5 voltios hasta los 7 voltios.

- La corriente máxima de operación está en 20 miliamperios.

- **Planta de control de velocidad**

La planta de control de velocidad tiene como características:

- El voltaje de operación nominal es de 5 voltios
- Trabaja en un rango de voltaje de 3 a 7 voltios.
- La corriente de operación es de 270 miliamperios como máximo
- El sensor infrarrojo envía un pulso por cada que la hélice del motor le prohíba el ingreso.
- La velocidad del motor será hasta máximo 15000 r.p.m.

- **Planta de control de temperatura**

La planta de control de temperatura tiene como características:

- Voltaje nominal de operación es de 5 voltios
- Se puede trabajar con un rango de voltaje de los 3.5 voltios hasta los 7 voltios.
- La corriente de operación es de 25 miliamperios.
- La temperatura mínima para trabajar está en un rango de 23° a 26 °C
- La temperatura máxima es de 60 °C

3.2. Diseño Esquemático del Módulo Didáctico

El diseño esquemático del módulo didáctico se dividirá de acuerdo con las PCBs, las cuales permiten tener un funcionamiento más directo, el software que hemos utilizado para el diseño de las tarjetas es Eagle en su versión más reciente.

3.2.1. Diseño esquemático de la tarjeta principal

Se realiza un diagrama esquemático de la tarjeta principal y los componentes en el software Eagle, el diseño esquemático de la tarjeta principal se realizó de acuerdo con

las necesidades de la asignatura de Sistemas Microprocesados en donde por medio del Pic se puede tener puertos de entradas y de salidas para ello el control se realiza por la Pantalla HMI Nextion.

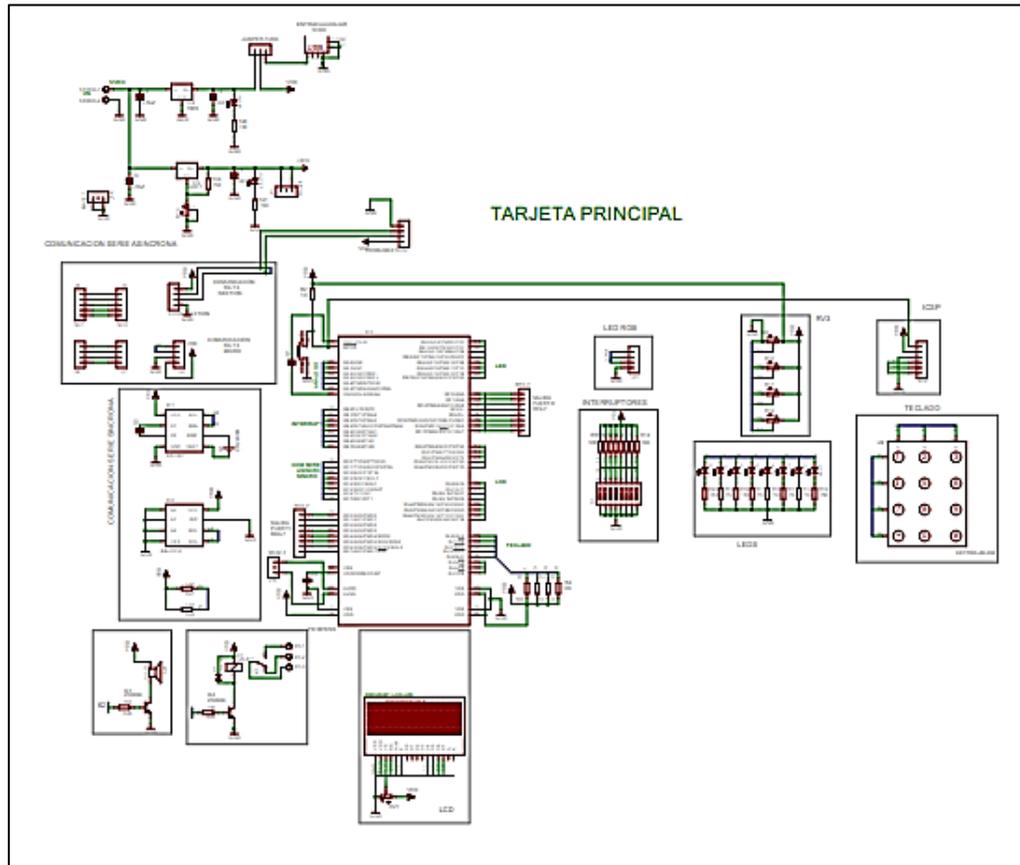


Fig. 34.- Esquemático de Tarjeta Principal
Fuente: Autores

3.2.2. Diseño esquemático de la Planta de Temperatura

Para realizar el control PID se diseñó una planta de temperatura en la cual se aplicaron conocimientos adquiridos en la carrera, en donde se ubicó protecciones y leds para verificar el funcionamiento. Estas protecciones se realizan a través de optoacopladores para evitar que el rendimiento baje en su eficiencia. Su diseño es de tierra común, en el momento que el calentador pase el nivel de temperatura descrita en la programación, el ventilador realiza el enfriamiento para llegar a un nivel de cero y así verificar su tiempo.

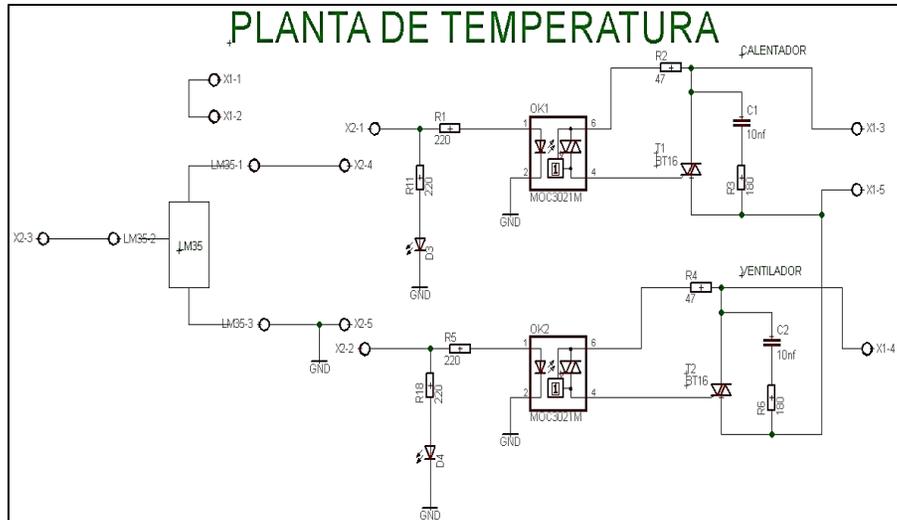


Fig. 35.- Esquemático de Planta de Temperatura
Fuente: Autores

3.2.3. Diseño esquemático de la Planta de Velocidad

Para la planta de control de velocidad su diseño esquemático es simple, ya que todo deberá registrarse a través de la programación es decir que el motor tendrá una hélice, que cuando pase por el sensor FC03 que es infrarrojo, envía al microcontrolador la cantidad de veces que ha realizado un giro de 360° o una vuelta, para determinar las revoluciones por minuto según el caso de programación de la práctica.

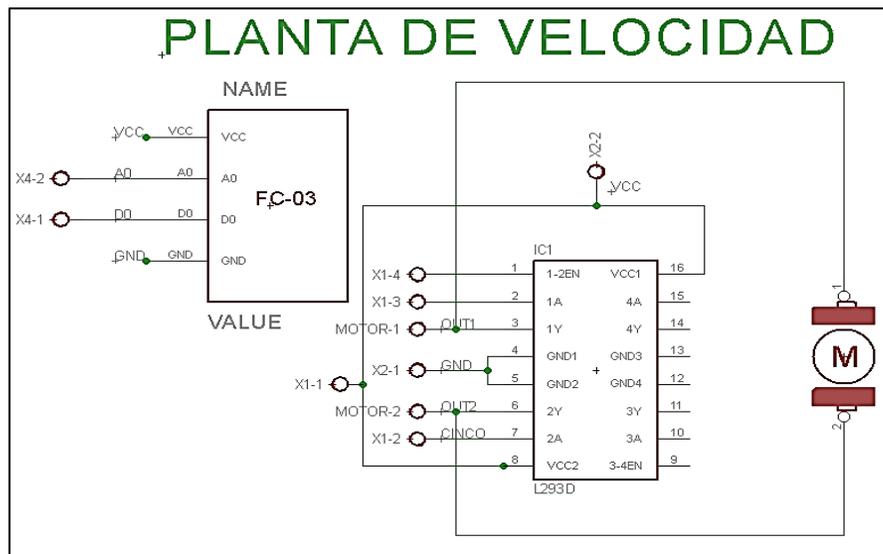


Fig. 36.- Esquemático de Planta de Velocidad
Fuente: Autores

3.2.4. Diseño esquemático de la Planta de Luminosidad

Para la planta de luminosidad, en su diseño esquemático, se tomó en cuenta los bajos voltajes que se receipta y para ello se utilizó un puente de Wheatstone, que permitirá un intervalo pequeño y a su vez se realiza una comparación con valor de voltaje ya definido (por el usuario) y ese valor se enviaré a la configuración de un amplificador diferencial.

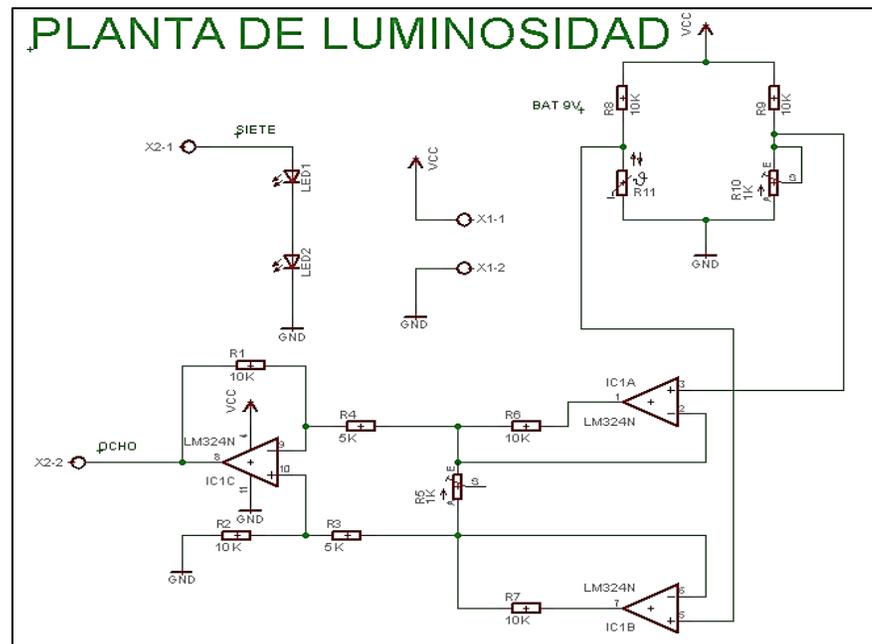


Fig. 37.- Esquemático de Planta de Luminosidad
Fuente: Autores

3.3. Diseño del Circuito Impreso PCB del Módulo Didáctico

De acuerdo con su diseño esquemático se realizó mediante el software Eagle el diseño del circuito impreso de las tarjetas que estan en el módulo didáctico.

3.3.1. Diseño del PCB de la tarjeta principal

El diseño del PCB (circuito impreso) de la tarjeta principal se realizó en doble capa (botón cooper y top cooper) para que así las conexiones de los elementos o componentes se encuentre en la cara superior los de montaje superficial y en la cara posterior los de Through-hole. Se dividió su diseño de acuerdo con el uso de los

componentes en la asignatura, como por ejemplo en el encendido de leds, el manejo o manipulación de interruptores, mando de teclado y LCD, comunicación serie síncrona y asíncrona, manipulación del Buzzer y puertos libres para conexiones externas que en este caso pueden ser variadas por el docente o por una asignatura diferente como instrumentación industrial.

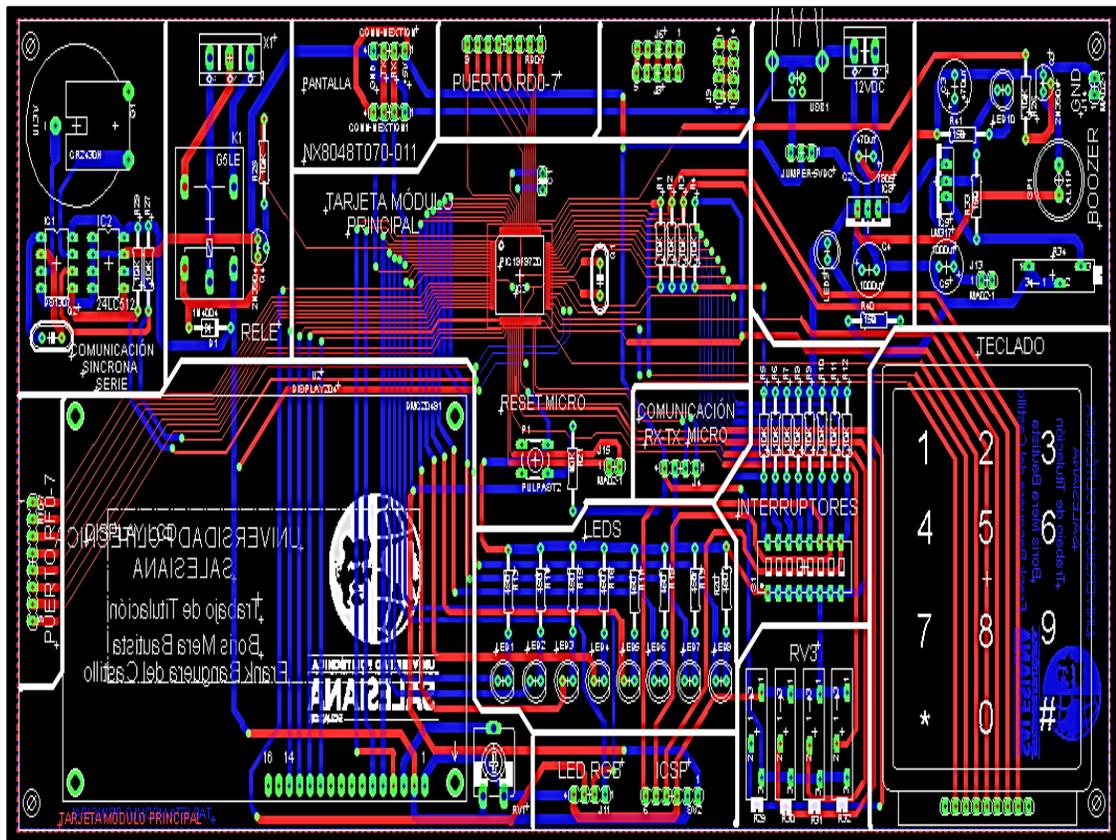


Fig. 38.- PCB de Tarjeta Madre
Fuente: Autores

3.3.2. Diseño del PCB de la Planta de Temperatura

En el diseño PCB de la planta de temperatura, la elaboración fue realizada en una capa en el software Eagle, ya que los elementos o componentes que se utilizan son Through-hole; en el caso de su diseño, se utilizan los optoacopladores como protección a cualquier cambio que exista en el voltaje de lo que se tiene ya diseñado, también el uso es particular en el funcionamiento de la elevación de la temperatura.

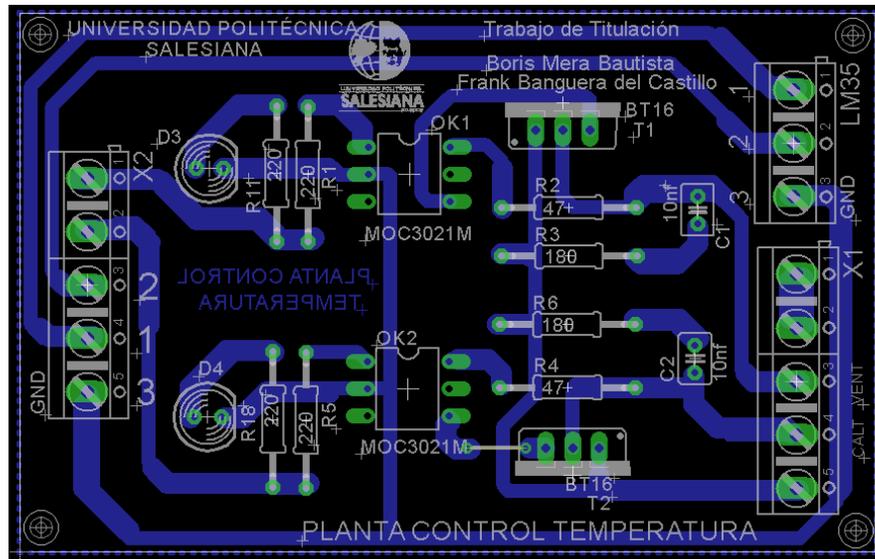


Fig. 39.- PCB de Planta de Temperatura
Fuente: Autores

3.3.3. Diseño del PCB de la Planta de Velocidad

La planta de control de velocidad se diseñó aplicando los conocimientos adquiridos en asignaturas previas, en especial en el uso de motor para corriente continua; por ello se ha utilizado un sensor infrarrojo y el integrado L293D que permitirán un manejo y control más estable de dicho motor. En el software Eagle de diseño con componentes en Through-hole el cual permite que se realice en una sola capa conocida como bottom cooper.

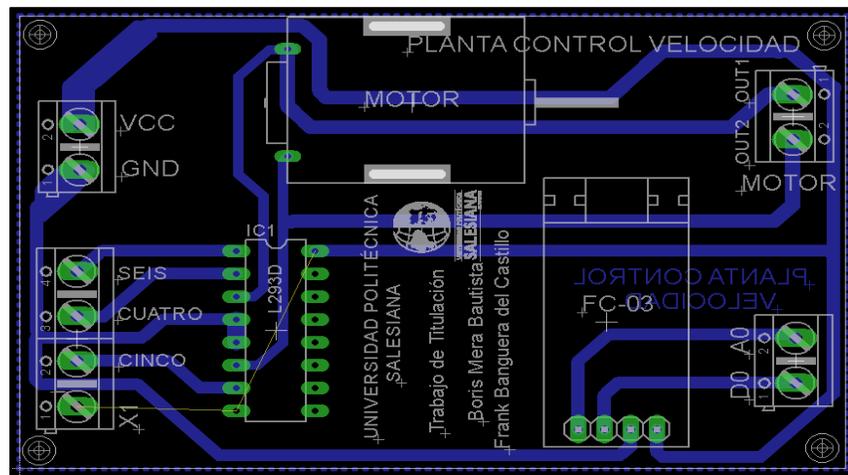


Fig. 40.- PCB de Planta de Velocidad
Fuente: Autores

3.3.4. Diseño del PCB de la Planta de Luminosidad

Para la planta de control de luminosidad se utilizaron elementos básicos los cuales permitieron que se utilice una sola capa para que la soldadura este en la parte posterior.

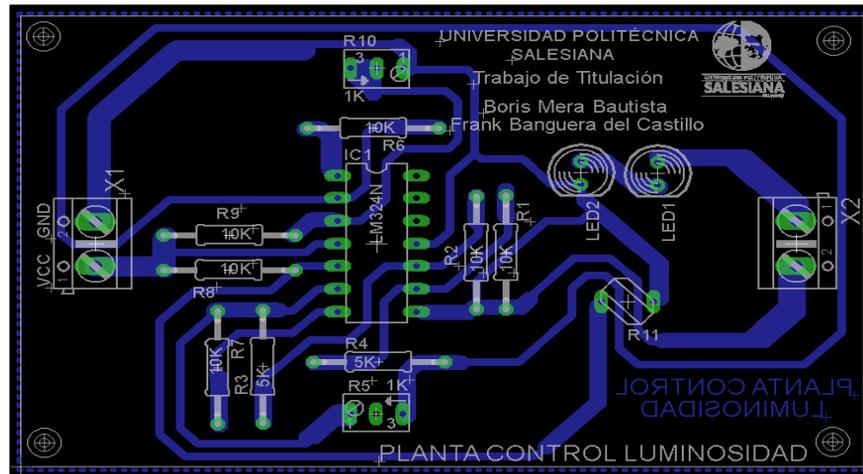


Fig. 41.- PCB de Planta de Luminosidad
Fuente: Autores

3.4. Implementación de las tarjetas Electrónicas

3.4.1. Listado de elementos de cada tarjeta

Se ha realizado la lista de elementos que se utilizan en cada una de las tarjetas de acuerdo con la circuitería:

3.4.1.1. Tarjeta Principal

- **Comunicación Síncrona Serie**
 - DS1307
 - 24LC512
 - 2 resistencias de 10k ohmios
 - 1 cristal de cuarzo
 - 1 fuente de 3 voltios (CR2430H)

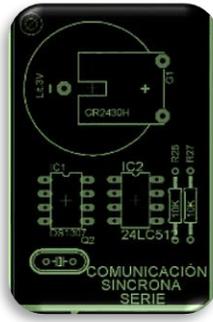


Fig. 42.- Comunicación Síncrona Serie
Fuente: Autores

- **Diseño de Relé**

1 diodo 1N4004

1 transistor NPN 2N3904

1 resistencia de 10k ohmios

1 relé de 5voltios

1 bornera de 3 entradas



Fig. 43.- Relé
Fuente: Autores

- **Puerto RD07**

8 pines macho



Fig. 44.- Puerto D
Fuente: Autores

- **Entrada de funcionamiento**

1 puerto USB

1 bornera de 2 entradas

1 regulador 7805

1 capacitor electrolítico de 470 uF

1 capacitor electrolítico de 1000 uF

1 diodo led

1 resistencia de 150 ohmios



*Fig. 45.- Entrada General
Fuente: Autores*

- **Diseño de Buzzer**

1 potenciómetro

1 capacitor de 1000 uF

2 resistencia de 150 ohmios

1 resistencia de 10K ohmios

1 regulador LM317

1 transistor NPN 2N3904

1 diodo led

1 Buzzer

1 capacitor electrolítico de 470 uF

2 bornera de 2 entradas



Fig. 46.- Buzzer
Fuente: Autores

- **Diseño de LCD**

1 Display de 20x4

1 potenciómetro para el contraste



Fig. 47.- LCD
Fuente: Autores

- **Salidas**

8 resistencias de 150 ohmios

8 diodos leds

1 bornera de 4 entradas para el led RGB

1 bornera para comunicación ICSP



Fig. 48.- Salidas Generales
Fuente: Autores

- **Interruptores**

- 1 bloque de switch de 8 pines
- 8 resistencias de 10k ohmios
- 4 potenciómetros

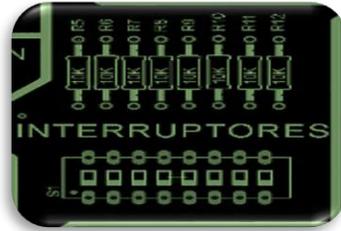


Fig. 49.- Interruptores
Fuente: Autores

- **Diseño Teclado**

Teclado numérico 4x3



Fig. 50.- Teclado Numérico 4x3
Fuente: Autores

- **Comunicación Rx Tx**

1 bornera de 4 entradas



Fig. 51.- Comunicación Rx Tx
Fuente: Autores

- **Diseño de módulo principal**

- 1 cristal de cuarzo
- 5 resistencias de 10k ohmios
- 1 capacitor cerámico 10nF
- 1 Pic 18F8720
- 1 pulsador
- 1 bornera de 2 entradas



*Fig. 52.- Pic 18F8720
Fuente: Autores*

3.4.1.2.Planta de Temperatura

- **Diseño de planta**

- 2 borneras de 3 entradas
- 2 borneras de 2 entradas
- 1 LM35
- 2 MOC3021
- 2 diodos leds
- 4 resistencias de 220 ohmios
- 2 resistencias de 180 ohmios
- 2 resistencias de 47 ohmios
- 2 capacitores cerámicos de 10nF
- 2 TRIAC BT16

3.4.1.3.Planta de Velocidad

- **Diseño de planta**

- 3 bornera de 2 entradas

1 bornera de 4 entradas
Integrado FC03
Integrado L293D
1 motor DC de 5 voltios

3.4.1.4.Planta de Luminosidad

- **Diseño de planta**
 - 2 borneras de 2 entradas
 - 2 diodos leds
 - 6 resistencias de 10k ohmios
 - 2 resistencias de 5k ohmios
 - Integrado LM324N
 - 2 potenciómetros de 1k ohmios
 - 1 fotorresistencia

3.4.2. Ubicación y soldado de los elementos

Los componentes que se utilizaron para la construcción del módulo didáctico se detallan a continuación:

Como se muestra en la Fig. 53 se visualiza el diseño de la placa sin las conexiones bottom o top del PCB y basado con este diseño se estableció la ubicación de soldado de cada uno de los elementos. Los elementos para su soldado fueron ubicados de manera estratégica la cual permitirán que el programador (docentes o estudiante) puedan realizar las prácticas de una forma simple, ordenada y compacta, es decir que las practicas establecidas se podrán mejorar o si se desea agregar una variable de entrada o salida externa.

Los elementos se los soldó de manera manual a través del uso de una estación, aplicando las técnicas aprendidas en la asignatura de Teoría del Diseño, y son:

- Paso 1: Calentar con el soldador los elementos a soldar.

- Paso 2: Aplicar estaño en su justa medida.
- Paso 3: No retirar el soldador hasta que el estaño se extienda.

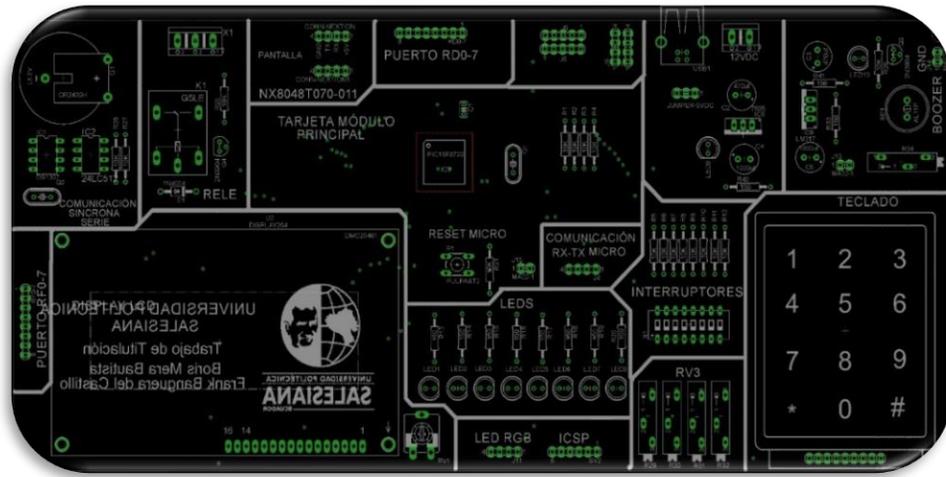


Fig. 53.- Placa de Soldado Principal
Fuente: Autores

Los elementos o componentes en su mayoría son de Through-hole y solo el Pic 18F8720 fue soldado a horno por ser de montaje superficial, en la Fig. 54 se puede observar que la técnica de soldado es básica y que, en el micro se soldaron dos cables para el funcionamiento de la comunicación de este con el Pick It 3.

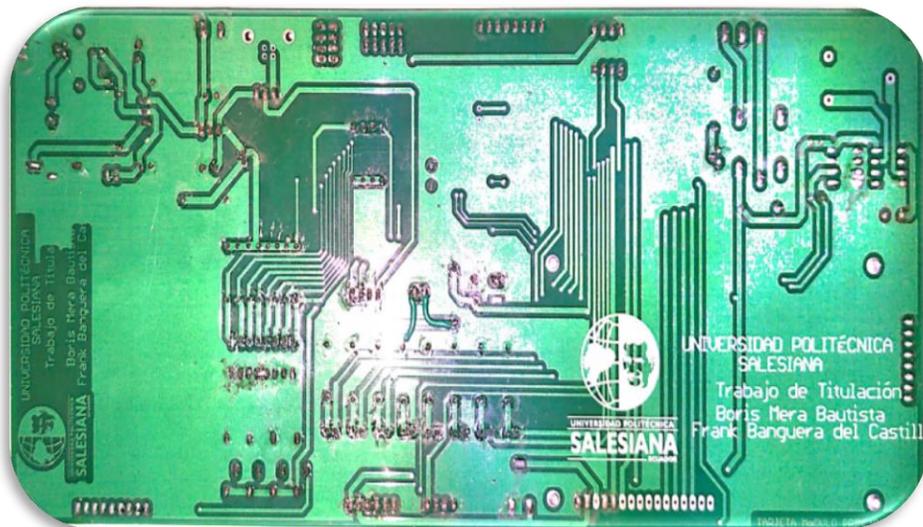
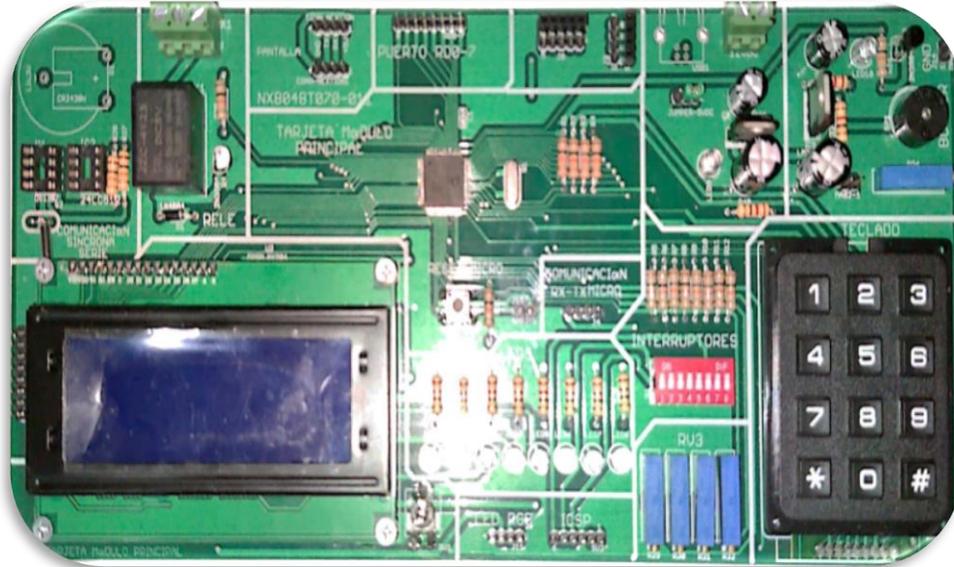


Fig. 54.- Elementos ubicados en la Tarjeta principal
Fuente: Autores

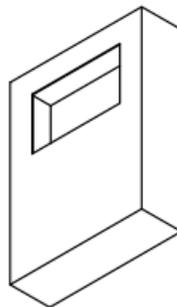
Los elementos de la Fig. 55 se han ubicado de manera que el manejo tanto del docente como del estudiante sea armonioso, así la enseñanza aprendizaje tenga su objetivo principal.



*Fig. 55.- Soldado de elementos en tarjeta principal
Fuente: Autores*

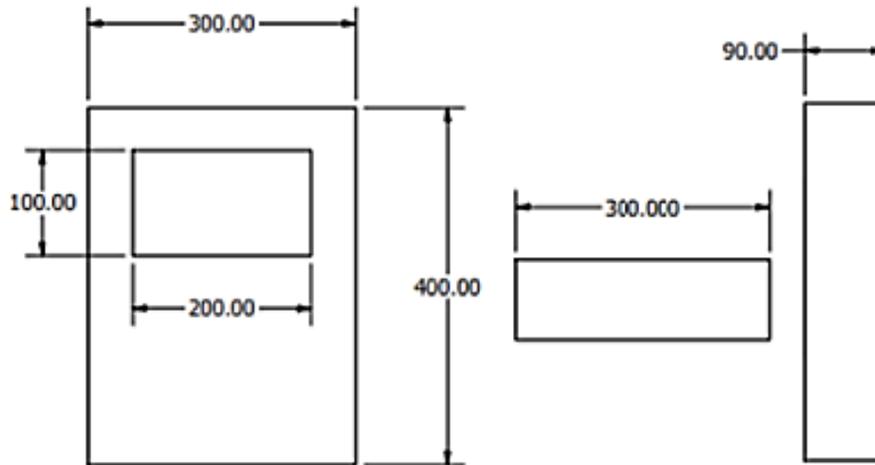
3.5. Construcción de Módulos Didácticos

En la estructura para el módulo didáctico se aplicaron los conocimientos adquiridos en la asignatura de AutoCAD la cual fue impartida en Primer Semestre, en donde lo único que se hizo fue actualizarnos con el software llamada INVENTOR pero con dichos conocimientos previos se pudo plasmar las ideas que se tenían para el módulo, como se visualiza en la figura 56.



*Fig. 56.- Diseño en INVENTOR
Fuente: Autores*

En la figura 57, se puede visualizar las medidas del módulo didáctico para que así se pueda enviar a elaborar la construcción con los materiales de metal galvanizado y acrílico negro; el diseño es compacto y pensando en todas las prácticas que se puedan realizar.



*Fig. 57.- Medidas de estructura
Fuente: Autores*

3.6. Pruebas de Funcionamiento

3.6.1. Comprobación de la programación del microcontrolador

Para la comprobación de la programación del microcontrolador Pic 18F8720 se realizó la compilación de una de las prácticas (Manejo de puertos del microcontrolador como entradas mediante la activación de las pantallas NEXTION), en donde se generó el archivo Hex para su funcionamiento virtual en el software Proteus y la transferencia de este al Pic con el programador.

Como se visualiza en la Fig. 58 el diseño en Proteus para el funcionamiento del Pic 18F8720, en donde se ubica el archivo Hex para su funcionamiento como el encendido de los leds (como salidas) conectados en el puerto E y activados por los switches del puerto B (como entradas).

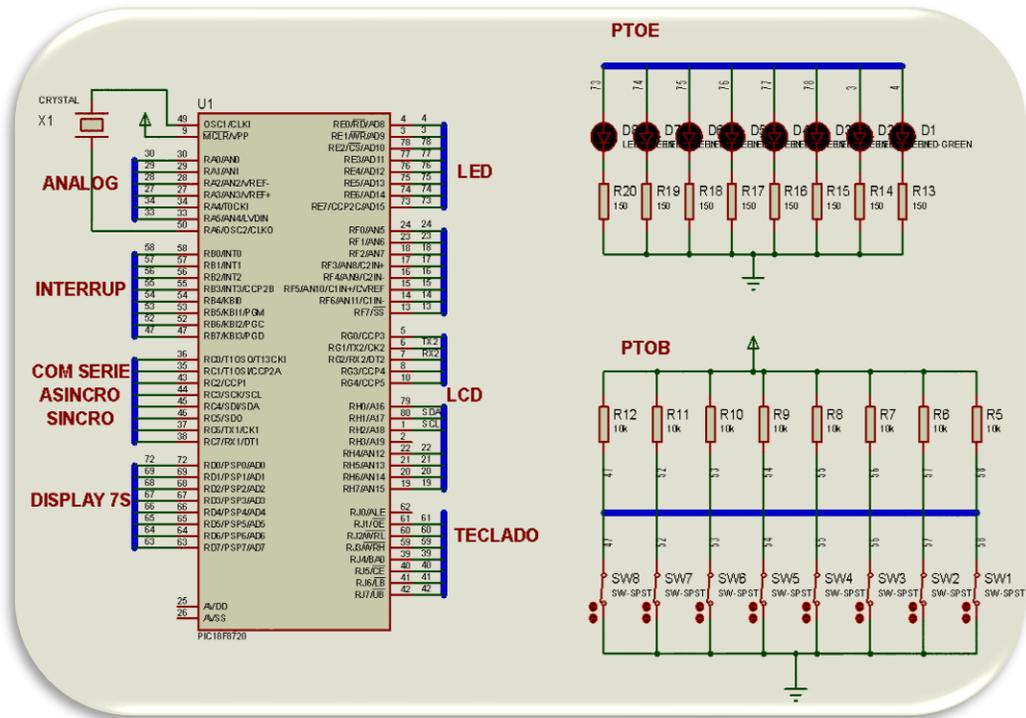


Fig. 58.- Funcionamiento del Pic 18F8720
Fuente: Autores

3.6.2. Comprobación de los puertos de entrada y salida

Para la comprobación de los puertos tanto como entrada y como salida, de acuerdo con la programación del Pic 18F8720 se realiza el encendido de los LEDS de manera automática (como salidas) y de manera manual (como entrada los interruptores), tanto en la parte de simulación como en el módulo ya implementado. Este funcionamiento se podrá visualizar en la Fig. 59 como parte de simulación mediante software y en la Fig. 60 la programación en la tarjeta principal del módulo.

3.6.3. Comprobación del funcionamiento de las pantallas gráficas LCD y NEXTION

El funcionamiento del LCD se realiza mediante simulación en el Software Proteus y su verificación en la implementación del módulo con la programación de encendido de leds y mensaje del programador ya diseñado con anterioridad. En la pantalla NEXTION se usa el IDE Guide que contiene un simulador incorporado y a su vez se realiza la prueba de manera real y física con un ozonizador.

Para verificar el funcionamiento correcto usaremos la práctica #2, en donde se realiza el encendido de los leds y de la pantalla gráfica LCD como salida y la entrada se maneja mediante los switch o interruptores tal como se visualiza en la Fig. 59

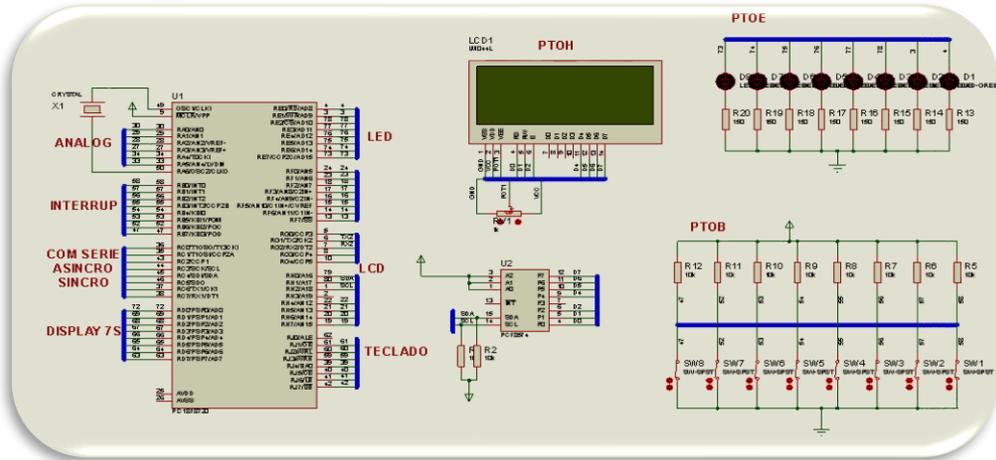


Fig. 59.- Práctica #2 funcionamiento de LCD y puertos
Fuente: Autores

En la Fig. 60 se puede comprobar dicho funcionamiento de la práctica #2 en donde en el LCD se puede visualizar el mensaje de bienvenida y con los interruptores el encendido de los leds de acuerdo con lo que se necesita.

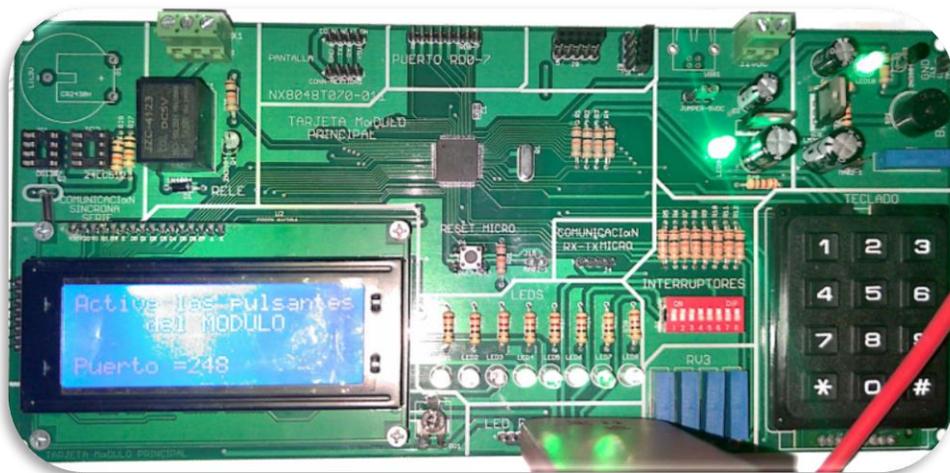


Fig. 60. Funcionamiento en tarjeta principal
Fuente: Autores

En este caso, el ozonizador que se diseñó en la pantalla NEXTION permitirá comprobar el funcionamiento, iniciamos diseñando en el IDE Guide de lo que vamos a efectuar; se utilizaron dos pantallas para que se pueda realizar de forma manual y automática, en la Fig. 61 se visualiza la pantalla 1 en donde se ubica varios textos (generador de Ozono y texto), varios botones (Configurar y activar) y una variable de contado (la cual muestra el tiempo establecido de funcionamiento).



Fig. 61.- Pantalla 1 Generador de Ozono
Fuente: Autores

En la pantalla 2, el diseño para la especificación de automática, si el usuario requiere de un tiempo de encendido y apagado, puede seguir trabajando todo el tiempo que se establezca. Se utilizaron 4 textos, dos botones y dos scroll para la manipulación del tiempo; como se visualiza en la Fig. 62 los scroll denominados h0 y h1 establecerán la configuración automática y una vez que se presione el botón **aceptar**, está configuración se guarda en la pantalla 1 y así enviar al microcontrolador para que realice el funcionamiento hexadecimal o binario de lo que se desea hacer.

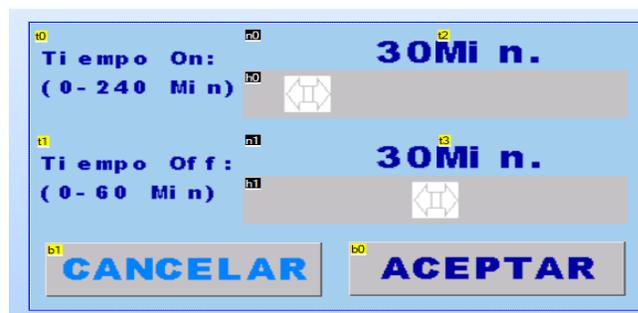


Fig. 62.- Pantalla 2 generador de ozono
Fuente: Autores

Como se visualiza en la Fig. 63 y Fig. 64, la programación de dichos botones se realiza a través de sus atributos y codifican en la ventana de eventos para que así el microcontrolador al recibir valores hexadecimales o binarios (de acuerdo con la necesidad del programado) pueda realizar la función de encendido y apagado del ozonizador.

```

Event
Timer Event(6)
// //x

va0.val=va0.val+1
if(va0.val==2)
{
print "CCCCCC"
tm0.en=0
}

```

Fig. 63.- Codificación de Pantalla 1

```

Event
Timer Event(2)
// //x

n0.val=h0.val
n1.val=h1.val

```

Fig. 64.- Codificación de Pantalla 2

En la fig. 65 se puede visualizar la implementación y funcionamiento de la pantalla HMI con el ozonizador, el cual se puede manipular tanto de manera manual como automática.

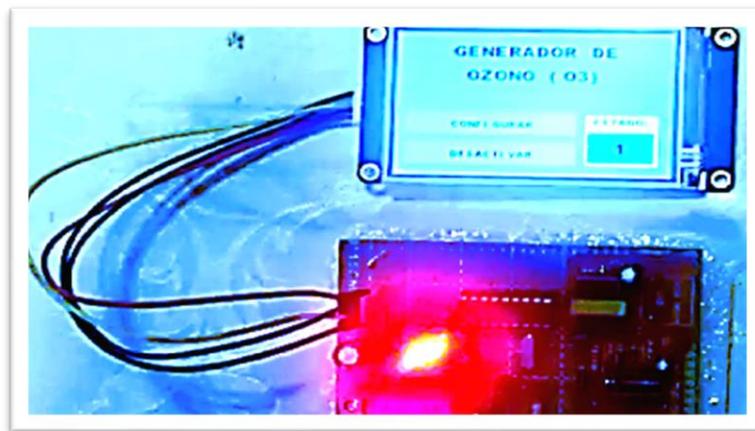


Fig. 65.- Funcionamiento del Ozonizador mediante Placa Madre y Pantalla NEXTION
Fuente: Autores

Capítulo 4: Desarrollo de Prácticas Propuestas

4.1. Práctica # 1: Manejo de puertos del microcontrolador como salidas mediante la activación de las pantallas NEXTION.

		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Manejo de puertos del microcontrolador como salidas mediante la activación de las pantallas NEXTION.
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> • Manejar puertos del microcontrolador como salidas para la activación de objetos en las pantallas NEXTION. OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> • Manejar puertos del microcontrolador como salidas. • Activar objetos en la pantalla NEXTION por medio de las salidas del microcontrolador. • Visualizar la activación de objetos en la pantalla NEXTION. 		
INSTRUCCIONES:		1. Establecer los puertos B y D como salidas en la programación del PIC C Compiler
		2. Diseñar en el IDE de NEXTION las salidas mediante el uso de formas geométricas
		3. Establecer la comunicación entre el microcontrolador y la pantalla HMI.
ACTIVIDADES POR DESARROLLAR		
1. Ejecutar el desarrollo del programa en PIC C Compiler para la declaración de puertos como salida.		
2. Compilar los programas para evitar errores o warning, y transferir la programación al microcontrolador con el Pick It 3 y a la pantalla HMI mediante puerto USB.		
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> • Podemos comprobar que los puertos establecidos están como salidas • Se aprecia el encendido de las formas geométricas. 		
CONCLUSIONES: <ul style="list-style-type: none"> • Se especifican los puertos como salida dentro de la programación para el control. • Se contrasta la correcta programación de las salidas digitales declaradas en el programa PIC C Compiler. 		
RECOMENDACIONES: <ul style="list-style-type: none"> • Verificar el encendido del módulo. • Determinar el espacio a trabajar. • Evitar manipulación errónea de objetos externos en los módulos. • Comprobar programa del Pic y de la pantalla NEXTION. • Compilar programas para evitar errores o daños. 		

4.2. Práctica # 2: Manejo de puertos del microcontrolador como entradas mediante la activación de las pantallas NEXTION.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: Manejo de puertos del microcontrolador como entradas mediante la activación de las pantallas NEXTION.	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> • Manejar puertos del microcontrolador como entradas para la activación de objetos en las pantallas NEXTION. OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> • Manejar puertos del microcontrolador como entradas. • Activar objetos en la pantalla NEXTION por medio de las entradas del microcontrolador. • Visualizar la activación de objetos en la pantalla NEXTION. 			
INSTRUCCIONES:		1. Establecer los puertos B y D como entradas en la programación del PIC C Compiler	
		2. Diseñar en el IDE de NEXTION las entradas mediante el uso de formas geométricas	
		3. Establecer la comunicación entre el microcontrolador y la pantalla HMI.	
ACTIVIDADES POR DESARROLLAR			
1. Ejecutar el desarrollo del programa en PIC C Compiler para la declaración de puertos como entrada.			
2. Compilar los programas para evitar errores o warning y transferir la programación al microcontrolador con el Pick It 3 y a la pantalla HMI mediante puerto USB.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> • Podemos comprobar que los puertos establecidos están como entradas. • Se aprecia el encendido de las formas geométricas. 			
CONCLUSIONES: <ul style="list-style-type: none"> • Se especifican los puertos como entrada dentro de la programación para el control. • Se contrasta la correcta programación de las salidas digitales declaradas en el programa PIC C Compiler. 			
RECOMENDACIONES: <ul style="list-style-type: none"> • Verificar el encendido del módulo. • Determinar el espacio a trabajar. • Evitar manipulación errónea de objetos externos en los módulos. • Comprobar programa del Pic y de la pantalla NEXTION. • Compilar programas para evitar errores o daños. 			

4.3. Práctica # 3: Manejo de herramientas de Nextion para elaboración de procesos industriales.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Manejo de herramientas de Nextion para elaboración de procesos industriales.	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> • Manejar herramientas de NEXTION a través de su IDE Guide para la creación de procesos industriales. OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> • Manejar herramientas de Nextion a través de su IDE Guide. • Utilizar toolbox para el desarrollo de procesos industriales. • Elaboración de procesos industriales por medio de la interfaz gráfica de Nextion. 			
INSTRUCCIONES:		1. Selección de objetos a trabajar.	
		2. Programación de objetos en pantalla HMI mediante atributos.	
		3. Establecer comunicación entre Pic y pantalla HMI.	
ACTIVIDADES POR DESARROLLAR			
1. Ejecutar el desarrollo del programa en PIC C Compiler para la declaración variables y comunicación entre dispositivos.			
2. Compilar los programas para evitar errores o warning y transferir la programación al microcontrolador con el Pick It 3 y a la pantalla HMI mediante puerto USB.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> • Podemos comprobar que el diseño de ozonizador es viable. • Se aprecia el encendido de placa principal de acuerdo con su programación. 			
CONCLUSIONES: <ul style="list-style-type: none"> • El manejo de herramientas en el IDE en NEXTION a través de la tarjeta principal. • Se contrasta la correcta programación y comunicación entre el Pic y la pantalla NEXTION. 			
RECOMENDACIONES: <ul style="list-style-type: none"> • Verificar el encendido del módulo. • Determinar el espacio a trabajar. • Evitar manipulación errónea de objetos externos en los módulos. • Comprobar programa de la pantalla NEXTION. • Compilar programa para evitar errores o daños. 			

4.4. Práctica # 4: Elaboración de librerías de objetos para procesos industriales.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	4	TÍTULO PRÁCTICA: Elaboración de librerías de objetos para procesos industriales.	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> • Crear librerías de objetos en NEXTION mediante el uso del IDE Guide para el diseño de procesos industriales. OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> • Utilizar librerías básicas de IDE Guide de NEXTION. • Diseñar librerías a partir de objetos básicos. • Manejar las diferentes herramientas del IDE Guide para la creación de librerías de objetos. 			
INSTRUCCIONES:		1. Insertar objetos establecidos por el docente	
		2. Establecer el funcionamiento de estos objetos de acuerdo con el proceso en cuestión.	
		3. Guardar las librerías de objetos creados.	
ACTIVIDADES POR DESARROLLAR			
1. Diseñar pantalla HMI de acuerdo con el proceso industrial.			
2. Establecer cambio de estados de imágenes para el proceso industrial.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> • Se comprueba funcionamiento de librería de acuerdo con el proceso industrial. • Cambio de estados de acuerdo con la señal digital enviada. 			
CONCLUSIONES: <ul style="list-style-type: none"> • El manejo de herramientas en el IDE en NEXTION a través de la tarjeta principal. • Se contrasta la correcta programación de la pantalla NEXTION. 			
RECOMENDACIONES: <ul style="list-style-type: none"> • Verificar el encendido del módulo. • Determinar el espacio a trabajar. • Compilar programas para evitar errores o daños. • Definir los objetos a utilizar en Nextion IDE. 			

4.5. Práctica # 5: Manejo de pantallas para la elaboración de menú en diferentes aplicaciones.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	5	TÍTULO PRÁCTICA: Manejo de pantallas para la elaboración de menú en diferentes aplicaciones.	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> • Manejar pantallas en NEXTION mediante la programación de menú para la elaboración de diferentes aplicaciones. OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> • Manejar el cambio de pantalla mediante el touch de la pantalla. • Manejar el cambio de pantallas mediante botones externos a la pantalla. • Desarrollar menús para aplicaciones a partir del manejo de cambio de pantallas. 			
INSTRUCCIONES:		1. Establecer áreas a diseñar.	
		2. Ubicar los componentes en el IDE de NEXTION.	
		3. Programar el funcionamiento de la programación de la Pantalla HMI.	
ACTIVIDADES POR DESARROLLAR			
1. Establecer comunicación entre la pantalla y los leds de la placa.			
2. Compilar los programas para evitar errores o warning y transferir la programación a la pantalla HMI mediante puerto USB.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> • Funcionamiento de pantalla por medio de aplicaciones regulares. • Comunicación entre pantalla y Pic mediante la intervención de leds. 			
CONCLUSIONES: <ul style="list-style-type: none"> • El manejo de herramientas en el IDE en NEXTION a través de la tarjeta principal. • Se contrasta la correcta programación de la pantalla NEXTION con el encendido de leds. 			
RECOMENDACIONES: <ul style="list-style-type: none"> • Verificar el encendido del módulo. • Determinar el espacio a trabajar. • Compilar programas para evitar errores o daños. • Definir la cantidad de pantallas a utilizar. • Establecer el nombre de estas. 			

4.6. Práctica # 6: Aprendizaje de manejo de simulador de pantallas NEXTION.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	6	TÍTULO PRÁCTICA: Aprendizaje de manejo de simulador de pantallas NEXTION.	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> Comprender el manejo del simulador de NEXTION mediante la programación de procesos industriales básicos para su aprendizaje. OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> Utilizar simulador del IDE de NEXTION con procesos básicos. Visualizar el simulador del IDE de NEXTION con procesos básicos. Manejar el simulador de IDE de Nextion como parte previa al funcionamiento real. 			
INSTRUCCIONES:		1. Selección de pantalla HMI	
		2. Diseño de acuerdo con el cambio de pantalla.	
		3. Envío de datos mediante teclado.	
ACTIVIDADES POR DESARROLLAR			
1. Diseñar las pantallas de manera manual y automática para el funcionamiento del simulador.			
2. Ejecutar el simulador de pantalla NEXTION una vez configurada la práctica.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> Funcionamiento correcto del simulador. Programación de pantalla HMI en funcionamiento correcto con el simulador. 			
CONCLUSIONES: <ul style="list-style-type: none"> El manejo del simulador en NEXTION mediante prácticas ya existentes. Diseño con diferentes pantallas de acuerdo a su necesidad previo a compra. 			
RECOMENDACIONES: <ul style="list-style-type: none"> Verificar el encendido del módulo. Determinar el espacio a trabajar. Diseñar de acuerdo con los objetos creados. Compilar programas para evitar errores o daños. Evitar errores antes de la simulación (compilar siempre). 			

4.7. Práctica # 7: Manejo Led RGB mediante el uso de pantallas Nextion.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	7	TÍTULO PRÁCTICA: Manejo Led RGB mediante el uso de pantallas Nextion.	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> • Manejar Led RGB mediante el uso de pantalla NEXTION para una mejor comprensión de los procesos industriales. OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> • Configurar iconos que representen LED RGB en la pantalla NEXTION. • Encender y Apagar Led RGB mediante la pantalla NEXTION. • Encender y Apagar Led RGB mediante interruptores externos a la pantalla NEXTION. 			
INSTRUCCIONES:		1. Establecer puertos aleatorios como entradas y salidas en la programación del PIC C Compiler	
		2. Diseñar en el IDE de NEXTION las salidas mediante el uso de librerías.	
		3. Establecer la comunicación entre el microcontrolador y la pantalla HMI.	
ACTIVIDADES POR DESARROLLAR			
1. Ejecutar el desarrollo del programa en PIC C Compiler para la declaración variables y comunicación entre dispositivos.			
2. Compilar los programas para evitar errores o warning y transferir la programación al microcontrolador con el Pick It 3 y a la pantalla HMI mediante puerto USB.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> • Podemos comprobar que la comunicación entre dispositivos es exitosa. • Se aprecia el funcionamiento correcto de la práctica mediante la pantalla HMI. 			
CONCLUSIONES: <ul style="list-style-type: none"> • Se especifican los puertos aleatorios como entradas o salidas de acuerdo con el programador. • Se contrasta la correcta programación y comunicación entre el Pic y la pantalla NEXTION. 			
RECOMENDACIONES: <ul style="list-style-type: none"> • Verificar el encendido del módulo. • Determinar el espacio a trabajar. • Evitar manipulación errónea de objetos externos en los módulos. • Comprobar programa del Pic y de la pantalla NEXTION. • Compilar programas para evitar errores o daños. 			

4.8. Práctica # 8:

Proceso Industrial #1: Control de luminosidad mediante seteo y visualización en pantalla NEXTION.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	8	TÍTULO PRÁCTICA: Proceso Industrial #1: Control de luminosidad mediante seteo y visualización en pantalla NEXTION	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> Controlar la planta de luminosidad mediante el seteo para la visualización en pantalla NEXTION OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> Controlar la planta de luminosidad Generar el PID de control en planta de luminosidad Visualizar en la pantalla NEXTION. 			
INSTRUCCIONES:		1. Establecer puertos aleatorios como entradas y salidas en la programación del PIC C Compiler	
		2. Diseñar en el IDE de NEXTION las salidas mediante el uso de librerías.	
		3. Establecer la comunicación entre el microcontrolador y la pantalla HMI.	
ACTIVIDADES POR DESARROLLAR			
1. Ejecutar el desarrollo del programa en PIC C Compiler para la declaración variables y comunicación entre dispositivos.			
2. Compilar los programas para evitar errores o warning y transferir la programación al microcontrolador con el Pick It 3 y a la pantalla HMI mediante puerto USB.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> Podemos comprobar que la comunicación entre dispositivos es exitosa. Se aprecia el funcionamiento correcto de la práctica mediante la pantalla HMI. 			
CONCLUSIONES: <ul style="list-style-type: none"> Se especifican los puertos aleatorios como entradas o salidas de acuerdo con el programador. Se contrasta la correcta programación y comunicación entre el Pic y la pantalla NEXTION. 			
RECOMENDACIONES: <ul style="list-style-type: none"> Verificar el encendido del módulo. Determinar el espacio a trabajar. Verificar formato de imágenes a importar. Compilar programas para evitar errores o daños. Probar el diseño en el módulo. 			

4.9. Práctica # 9:

Proceso Industrial #2: Control de temperatura mediante seteo y visualización en pantalla NEXTION.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	9	TÍTULO PRÁCTICA: Proceso Industrial #2: Control de temperatura mediante seteo y visualización en pantalla NEXTION.	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> Controlar la planta de temperatura mediante el seteo para la visualización en pantalla NEXTION OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> Controlar la planta de temperatura Generar el PID de control en planta de temperatura Visualizar en la pantalla NEXTION. 			
INSTRUCCIONES:		1. Establecer puertos aleatorios como entradas y salidas en la programación del PIC C Compiler	
		2. Diseñar en el IDE de NEXTION las salidas mediante el uso de librerías.	
		3. Establecer la comunicación entre el microcontrolador y la pantalla HMI.	
ACTIVIDADES POR DESARROLLAR			
1. Ejecutar el desarrollo del programa en PIC C Compiler para la declaración variables y comunicación entre dispositivos.			
2. Compilar los programas para evitar errores o warning y transferir la programación al microcontrolador con el Pick It 3 y a la pantalla HMI mediante puerto USB.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> Podemos comprobar que la comunicación entre dispositivos es exitosa. Se aprecia el funcionamiento correcto de la práctica mediante la pantalla HMI. 			
CONCLUSIONES: <ul style="list-style-type: none"> Se especifican los puertos aleatorios como entradas o salidas de acuerdo con el programador. Se contrasta la correcta programación y comunicación entre el Pic y la pantalla NEXTION. 			
RECOMENDACIONES: <ul style="list-style-type: none"> Verificar el encendido del módulo. Determinar el espacio a trabajar. Evitar manipulación errónea de objetos externos en los módulos. Comprobar programa del Pic y de la pantalla NEXTION. Compilar programas para evitar errores o daños. 			

4.10. Práctica # 10:

Proceso Industrial #3: Control de variación de velocidad de motor DC mediante seteo y visualización en pantalla NEXTION.

		GUÍA DE PRÁCTICA DE LABORATORIO	
CARRERA: Electrónica y Automatización		ASIGNATURA: Sistemas Microprocesados	
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Proceso Industrial #3: Control de variación de velocidad de motor DC mediante seteo y visualización en pantalla NEXTION.	
OBJETIVOS: OBJETIVO GENERAL <ul style="list-style-type: none"> Controlar la planta de velocidad mediante el seteo para la visualización en pantalla NEXTION OBJETIVOS ESPECÍFICOS <ul style="list-style-type: none"> Controlar la planta de velocidad Generar el PID de control en planta de velocidad Visualizar en la pantalla NEXTION. 			
INSTRUCCIONES:		1. Establecer puertos aleatorios como entradas y salidas en la programación del PIC C Compiler	
		2. Diseñar en el IDE de NEXTION las salidas mediante el uso de librerías.	
		3. Establecer la comunicación entre el microcontrolador y la pantalla HMI.	
ACTIVIDADES POR DESARROLLAR			
1. Ejecutar el desarrollo del programa en PIC C Compiler para la declaración variables y comunicación entre dispositivos.			
2. Compilar los programas para evitar errores o warning y transferir la programación al microcontrolador con el Pick It 3 y a la pantalla HMI mediante puerto USB.			
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> Podemos comprobar que la comunicación entre dispositivos es exitosa. Se aprecia el funcionamiento correcto de la práctica mediante la pantalla HMI. 			
CONCLUSIONES: <ul style="list-style-type: none"> Se especifican los puertos aleatorios como entradas o salidas de acuerdo con el programador. Se contrasta la correcta programación y comunicación entre el Pic y la pantalla NEXTION. 			
RECOMENDACIONES: <ul style="list-style-type: none"> Verificar el encendido del módulo. Determinar el espacio a trabajar. Evitar manipulación errónea de objetos externos en los módulos. Comprobar programa del Pic y de la pantalla NEXTION. Compilar programas para evitar errores o daños. 			

Presupuesto

Tabla 2.- Presupuesto
Fuente: Autores

		DISEÑO E IMPLEMENTACIÓN DE MÓDULOS DIDÁCTICOS PARA EL DESARROLLO DE APLICACIONES HMI MEDIANTE EL USO DE MICROCONTROLADORES SMD Y PANTALLAS NEXTION		
N°	DESCRIPCIÓN	CANTIDAD	VALOR POR UNIDAD	VALOR TOTAL
1	DS1307	2	\$ 4,00	\$ 8,00
2	24LC512	2	\$ 2,00	\$ 4,00
3	CR2430H	2	\$ 2,50	\$ 5,00
4	Diodo 1N4004	2	\$ 0,20	\$ 0,40
5	Transistor NPN 2N3904	2	\$ 1,50	\$ 3,00
6	Resistencia 10Kohmios	44	\$ 0,02	\$ 0,88
7	Relé 5V	2	\$ 3,00	\$ 6,00
8	Borneras De 3 Entradas	6	\$ 1,80	\$ 10,80
9	Pines Macho	16	\$ 0,30	\$ 4,80
10	Puerto USB	2	\$ 4,00	\$ 8,00
11	Regulador 7805	2	\$ 0,50	\$ 1,00
12	capacitor Dieléctrico de 470 uF	4	\$ 0,20	\$ 0,80
13	capacitor dieléctrico de 1000 uF	4	\$ 0,25	\$ 1,00
14	Diodo led	28	\$ 0,10	\$ 2,80
15	Resistencia 150 ohmios	18	\$ 0,05	\$ 0,90
16	Potenciómetro	16	\$ 1,00	\$ 16,00
17	Regulador LM317	2	\$ 0,90	\$ 1,80
18	Buzzer	2	\$ 1,00	\$ 2,00
19	Borneras De 2 Entradas	20	\$ 1,20	\$ 24,00
20	Display LCD 20x4	2	\$ 11,00	\$ 22,00
21	Borneras De 4 Entradas	4	\$ 2,90	\$ 11,60
22	Bloque De Switch	2	\$ 0,90	\$ 1,80
23	Teclado Numérico 4x3	2	\$ 4,50	\$ 9,00
24	Cristal De Cuarzo	4	\$ 0,25	\$ 1,00
25	PIC 18F8720	2	\$ 17,80	\$ 35,60
26	Pulsador	2	\$ 0,95	\$ 1,90
27	LM35	2	\$ 1,00	\$ 2,00
28	MOC 3021	4	\$ 0,85	\$ 3,40
29	Resistencia 220 Ohmios	8	\$ 0,05	\$ 0,40
30	Resistencia 180 Ohmios	4	\$ 0,05	\$ 0,20
31	Resistencia 47 Ohmios	4	\$ 0,05	\$ 0,20

32	Capacitor cerámico 10nF	6	\$	0,10	\$	0,60
33	Triac BT16	4	\$	2,50	\$	10,00
34	Integrado FC03	2	\$	1,30	\$	2,60
35	Integrado L293D	2	\$	2,20	\$	4,40
36	Motor DC 5v	2	\$	1,00	\$	2,00
37	Resistencia 5Kohmios	4	\$	0,05	\$	0,20
38	Fotorresistencia	2	\$	0,56	\$	1,12
39	Porta Fusible De Panel	2	\$	3,50	\$	7,00
40	Protoboard	2	\$	8,00	\$	16,00
41	Pantalla HMI Nextion	2	\$	89,00	\$	178,00
42	Impresión De Placas PCB (Principal)	2	\$	34,60	\$	69,20
43	Impresión De Placas PCB (Plantas)	6	\$	2,90	\$	17,40
44	Importación De Tarjetas	1	\$	75,00	\$	75,00
45	Estructura Del Módulo	2	\$	250,00	\$	500,00
46	Control De Señales De Entradas	2	\$	2,50	\$	5,00
47	Fuente De Poder	2	\$	10,00	\$	20,00
48	Pick It 3	2	\$	40,00	\$	80,00
49	Módulo Bluetooth	2	\$	6,00	\$	12,00
50	Plugs Hembras	16	\$	0,70	\$	11,20
51	Plugs Machos	16	\$	0,85	\$	13,60
52	Fusible Enroscable Y Base	2	\$	2,15	\$	4,30
53	Enchufe	2	\$	0,85	\$	1,70
54	Interruptor	2	\$	1,20	\$	2,40
55	Ventilador De Tablero	2	\$	12,00	\$	24,00
56	Cable Flexible #20	1 rollo	\$	10,00	\$	10,00
57	Estructura Plantas	6	\$	25,00	\$	150,00
58	Jumpers (Por Juegos)	4	\$	4,50	\$	18,00
VALOR FINAL					\$	1.426,00

Conclusiones

Mediante esta tesis de grado o proyecto técnico se amplió el aprendizaje complementario de lo teórico y práctico de la asignatura de Sistemas Microprocesados, mediante el uso de la electrónica en conjunto con la interfaz de programación del PIC C Compiler con la pantalla Nextion para la implementación de prácticas industriales.

Los conocimientos adquiridos en la Universidad Politécnica Salesiana Sede Guayaquil en la Carrera de Ingeniería Electrónica se afianzaron con el diseño de estos módulos, además de la investigación de diferentes temas con la finalidad de dar una solución eficaz y concreta.

Los módulos didácticos permiten procesar datos, consultar y a la vez mostrarlos a través de las pantallas LCD y/o HMI, lo cual permite que el docente pueda consolidar los conocimientos que se imparten de una manera mucho más dinámica y practica en la asignatura.

Por el desarrollo de prácticas ya establecidas y la comprensión del uso correcto de los módulos para el laboratorio de Electrónica Digital, el estudiante logra consolidar los conocimientos teóricos y podrá implementar cada uno de ellos cumpliendo el objetivo principal de este proyecto que es de dar un aprendizaje significativo en donde el estudiante pueda dar soluciones reales y veraces.

Recomendaciones

El diseño en la pantalla HMI tiene un lenguaje gráfico muy claramente definido, el cual se debe seguir para que los lectores interesados puedan entenderlo completamente, se recomienda utilizar el programa de software Nextion que incluye todos los símbolos e interacciones para este propósito.

El uso de módulos didácticos (como los que se utilizan en este trabajo) es muy útil, no solo por su bajo costo, sino también porque son de mucha ayuda en la parte práctica, y además son dispositivos con muchos recursos y alto capacidades de procesamiento.

El lenguaje de programación CC ++ es fundamental para el desarrollo del código de programación, se puede implementar correctamente el modelado detallado, reduciendo así los errores causados por problemas de lógica.

Es muy importante definir sistemáticamente el proceso de prueba experimental del equipo de acuerdo con el método formal, lo que puede hacer que los resultados obtenidos cumplan con los estándares establecidos por el método científico para demostrar su efectividad.

Bibliografía

- BENAVIDES CAGUASANGO , H. A., & JIMÉNEZ BENALCÁZAR, J. D. (2016). “*DISEÑO Y CONSTRUCCIÓN DE TRES MÓDULOS DIDÁCTICOS PARA MEDICIÓN Y CONTROL DE TEMPERATURA AMBIENTE E IMPLEMENTACIÓN DE UN SCADA EN EL LABORATORIO DE LA CARRERA DE INGENIERÍA EN MANTENIMIENTO ELÉCTRICO*”. Ibarra.
- CARVAJAL YANNUZZELLI, J. S. (2018). *Implementación de módulos didácticos básicos basados en el microcontrolador atmega164 para el laboratorio de microprocesadores de la Esfot*. Quito.
- Componentes 101. (25 de Septiembre de 2017). *Componentes 101*. Obtenido de <https://components101.com/buzzer-pinout-working-datasheet>
- CRUZ SASIG, F. O. (2019). *Implementación de sistema de control y supervisión HMI para relleno de tubos de rayos X*. Quito.
- ELECTRÓNICA PLUG AND PLAY. (ENERO de 2019). HOJA TÉCNICA LM324. COLOMBIA. Obtenido de <https://www.electronicaplugandplay.com/circuitos-integrados/product/480-amplificador-operacional-lm324>
- Equipo Led Tecnología. (15 de Enero de 2019). *Led Tecnología*. Obtenido de <https://www.ledtecnologia.com/que-es-rgb/>
- FAIRCHILD. (05 de FEBRERO de 2001). *ELETRÓNICA PLUG AND PLAY*. Recuperado el 27 de ENERO de 2021, de <https://www.electronicaplugandplay.com/:https://www.electronicaplugandplay.com/component/edocman/?task=document.viewdoc&id=146&Itemid=>
- Franco Reina, R. C., & Monteseoca Paladines, I. (2014). *Diseño y Construcción de cinco entrenadores didácticos con sistemas microprocesados y desarrollo de una aplicación de control de velocidad para un motor de corriente alterna*. Guayaquil.
- I.Crisologo. (14 de Julio de 2011). Teclado Matricial 4x3. *Arduino y teclado 3x4*. Rio de Janeiro, Brasil.
- INFAIMON. (12 de Abril de 2018). *INFAIMON.COM*. Recuperado el 27 de ENERO de 2021, de INFAIMON.COM: <https://blog.infaimon.com/detector-infrarrojo-funcionamiento-aplicaciones/>

Ingeniería Mecafenix. (06 de Abril de 2017). *Ingeniería Mecafenix*. Recuperado el 26 de Enero de 2021, de Ingeniería Mecafenix: <https://www.ingmecafenix.com/electronica/microcontrolador-pic-partes-aplicaciones/>

Ingeniería Mecafenix. (15 de Octubre de 2018). *Ingeniería Mecafenix*. Obtenido de <https://www.ingmecafenix.com/electronica/el-buzzer/>

ISOCOM COMPONENTS. (17 de Julio de 2008). *ELECTRÓNICOS CALDAS*. Obtenido de https://www.electronicoscaldas.com/datasheet/MOC3020X-MOC3021X-MOC3022X-MOC3023X-MOC3020-MOC3021-MOC3022-MOC3023_Isocom.pdf

Kitronik Ltd. (s.f.). Kitronik Ltd – 5mm RGB LED Common Cathode. *TECHNOLOGY DATA SHEET & SPECIFICATIONS*. UK: DataSheet.

Llamas, L. (02 de Octubre de 2016). *luisllamas.es*. Recuperado el 26 de Enero de 2021, de Ingeniería, informática y diseño: <https://www.luisllamas.es/arduino-teclado-matricial/>

Marmolejo, D. R. (25 de noviembre de 2017). *HETPRO*. (HETPRO) Recuperado el 25 de enero de 2021, de HETPRO: <https://hetpro-store.com/TUTORIALES/lm35/>

Maxim Integrated. (26 de Enero de 2015). *Maximintegrated.com*. Recuperado el 26 de Enero de 2021, de Maximintegrated.com: <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>

MicroChip. (24 de Junio de 2015). *MicroChip*. Recuperado el 25 de Enero de 2021, de MicroChip: <https://ww1.microchip.com/downloads/en/DeviceDoc/39609C.pdf>

MicroChip. (24 de Junio de 2015). *MicroChip.com*. Recuperado el 26 de Enero de 2021, de MicroChip.com: <https://ww1.microchip.com/downloads/en/DeviceDoc/21754M.pdf>

MINEBEA MOTOR MANUFACTURING CORPORATION . (07 de Diciembre de 2010). *MINEBEA MOTOR MANUFACTURING CORPORATION* . Obtenido de https://www.electronicoscaldas.com/datasheet/PPN7PA12C1_NMB-Technologies.pdf

Nextion Instruments. (25 de Octubre de 2020). <https://nextion.tech/>. Obtenido de <https://nextion.tech/datasheets/nx1060p101-011r-i/>

Nextion Instruments. (25 de Octubre de 2020). <https://nextion.tech/>. Obtenido de https://nextion.tech/editor_guide/#e5

PrestaShop. (Enero de 2021). *TostaTronic Shop*. Obtenido de <https://tostatronic.com/store/es/motores/101-motor-corriente-directa-cd-a-5v.html>

Robots Argentina. (05 de Noviembre de 2019). *Robots Argentina*. Obtenido de <http://robots-argentina.com.ar/didactica/?s=Manejo+de+potencia+para+motores+con+el+integrado+L293D>

Seas. (22 de Agosto de 2019). *BlogSeas*. Recuperado el 27 de Enero de 2021, de BlogSeas.com: <https://www.seas.es/blog/automatizacion/el-rele-para-que-es-para-que-sirve-y-que-tipos-existen/>

Semiconductor Components Industries. (03 de Septiembre de 2012). *Farnell*. Recuperado el 25 de Enero de 2021, de Newark: <http://www.farnell.com/datasheets/1708247.pdf>; <http://onsemi.com>

SGS-THOMSON MicroElectronics. (1 de Marzo de 1995). *alldatasheet.es*. Recuperado el 25 de Enero de 2021, de alldatasheet.es: <https://pdf1.alldatasheet.es/datasheet-pdf/view/22033/STMICROELECTRONICS/BTA16-600B.html>

TEXAS INSTRUMENTS. (ENERO de 2015). LM324 DATASHEET. *LMx24-N, LM2902-N Low-Power, Quad-Operational Amplifiers*. TEXAS, ESTADOS UNIDOS. Obtenido de <https://www.electronicaplugandplay.com/component/edocman/?task=document.viewdoc&id=131&Itemid=>

Texas Instruments. (Enero de 2016). L293x Quadruple Half-H Drivers. Dallas, Texas.

TEXAS INSTRUMENTS. (1 de DICIEMBRE de 2017). *ALLDATASHEET.ES*. Recuperado el 25 de ENERO de 2021, de ALLDATASHEET.ES: https://www.ti.com/lit/ds/symlink/lm35.pdf?HQS=TI-null-null-alldatasheets-df-pf-SEP-ww&ts=1611590436583&ref_url=https%253A%252F%252Fpdf1.alldatasheet.com%252F

Vishay. (01 de Enero de 2021). *VISHAY.com*. Recuperado el 26 de Enero de 2021, de VISHAY.com: <https://www.vishay.com/docs/37314/lcd020n004l.pdf>

ANEXOS

ANEXO 1: Solución propuesta de la práctica #1

Creamos un proyecto en Pic C Compiler y le asignamos un nombre que describa la práctica.

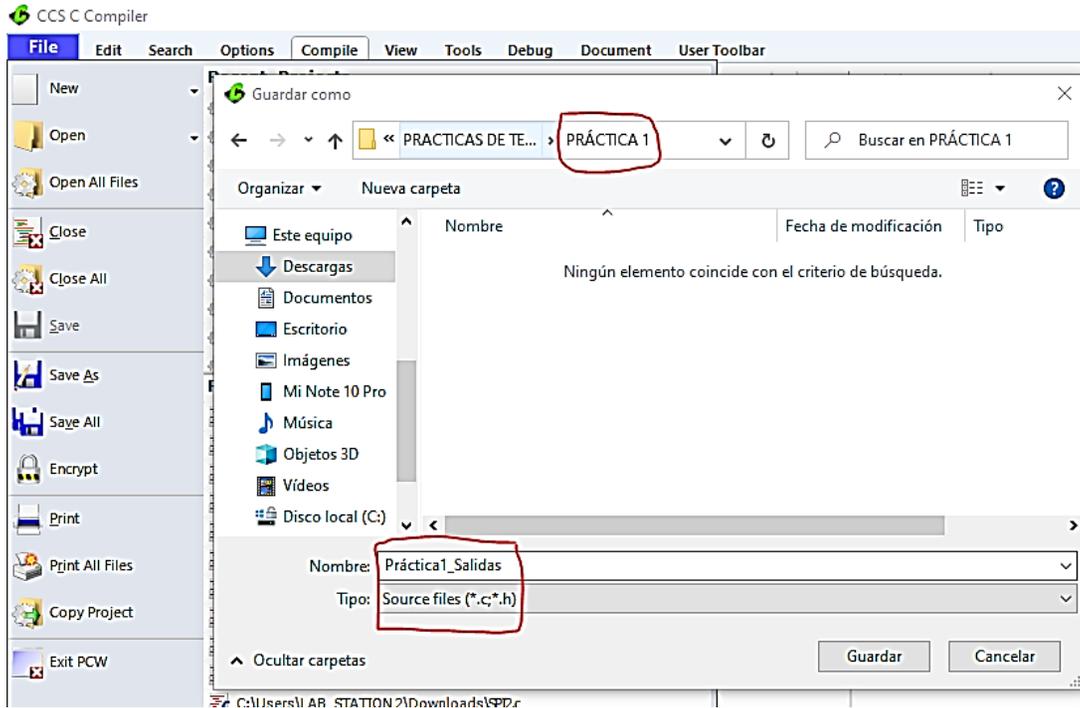


Fig. 67.- Creación de proyecto en PIC C Compiler
Fuente: Autores

Una vez creado el proyecto, detallamos las líneas de código predeterminadas las cuales permiten la inicialización del microcontrolador.

```
1 #include <18f8720.h>
2 #fuses xt, NOWDT, NOPROTECT, NOBROWNOUT, NOLVP, NOCPD, NOPUT, NOWRT
3 #use delay(clock=4000000)
4 #use i2c(Master, Fast=100000, sda=PIN_H1, scl=PIN_H2, force_sw)
5 #use rs232(baud=9600, xmit=PIN_G1, rcv=PIN_G2, TIMEOUT=500) //C6 -- C7
6 #include <i2c_Flex_LCD.c>
7
```

Fig. 68.- Códigos predeterminados de inicialización
Fuente: Autores

En la figura anterior se visualiza la línea de código 4 y 5 que permitirán la comunicación con la pantalla NEXTION y de ahí debemos ubicar las líneas de códigos que permitirán recibir los códigos enviados por la pantalla al micro y visualizarlos en los LEDs, estas líneas se establecerán como función en Pic C Compiler conocida como Interrupción interna.

```

#INT_RDA2
void RDA_isr2()
{
    disable_interrupts(INT_RDA2);
    s7=getc();
    s6=getc();
    s5=getc();
    s4=getc();
    s3=getc();
    s2=getc();
    s1=getc();
    s0=getc();

    s7=s7-48; s6=s6-48; s5=s5-48; s4=s4-48; s3=s3-48; s2=s2-48; s1=s1-48; s0=s0-48;
    valor=s7+(s6*2)+(s5*4)+(s4*8)+(s3*16)+(s2*32)+(s1*64)+(s0*128);
    porte=valor;
    printf("page1.n0.val=%u",valor); printf("yy\u000d");
    lcd_gotoxy(1,4);
    lcd_putc("      ");
    lcd_gotoxy(1,4);
    printf(lcd_putc,"Puerto=%u\u000d\u000d\u000d\u000d\u000d\u000d\u000d\u000d",s6,s5,s4,s3,s2,s1,s0,s7);
    enable_interrupts(INT_RDA2);
    // delay_ms(50);
    // output_low(PIN_J0);
}

```

Fig. 69.- Interrupción Interna
Fuente: Autores

Luego de ella se establecerá el main que permitirá activar los puertos como salida y visualizar en el LCD un mensaje de bienvenida mientras que entrará en un ciclo infinito hasta que se reinicie el proceso o se apague.

```

39 void main()
40 {
41     set_tris_b(0xFF);
42     set_tris_e(0x00);
43     lcd_init(0x4E,20,4);
44     enable_interrupts(GLOBAL);
45     enable_interrupts(INT_RDA2);
46
47     lcd_putc("    UNIVERSIDAD");
48     lcd_gotoxy(1,2);
49     lcd_putc("    POLITECNICA");
50     lcd_gotoxy(1,3);
51     lcd_putc("    SALESIANA");
52     lcd_gotoxy(1,4);
53     lcd_putc("    GUAYAQUIL");
54     delay_ms(1000);
55     lcd_putc("\fActive los pulsantes");
56     lcd_gotoxy(1,2);
57     lcd_putc("    en la NEXTION");
58     porte=255;
59     delay_ms(500);
60     porte=0;
61     while(true)
62     {
63     }
64 }
65 }

```

Fig. 70.- Puerto como salida y Mensaje de Bienvenida
Fuente: Autores

Una vez establecido el proyecto en PIC C Compiler, ahora programaremos la Pantalla Nextion para ello abriremos un proyecto en el software Nextion y ubicaremos un nombre representativo de la práctica.

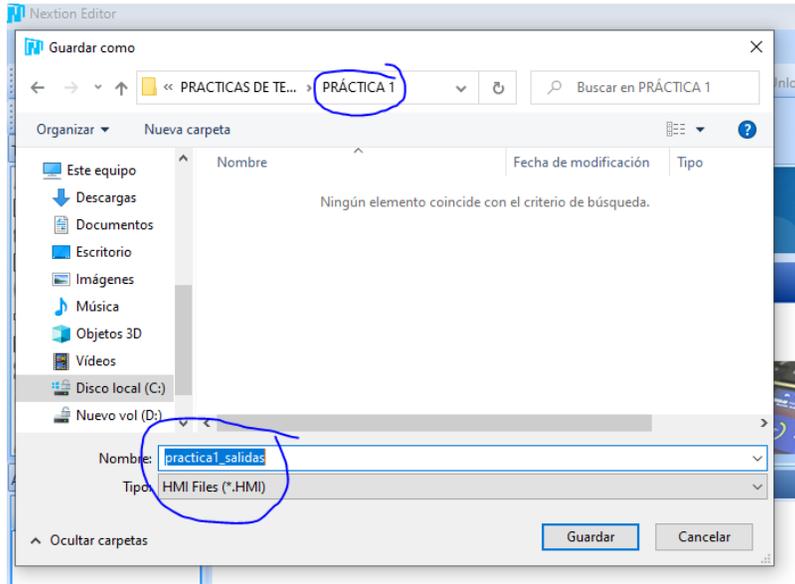


Fig. 71.- Creación de proyecto en Nextion
Fuente: Autores

Estableceremos dos páginas para trabajar en ellas, página 1 la presentación de la práctica y página 2 el desarrollo visual de la misma.

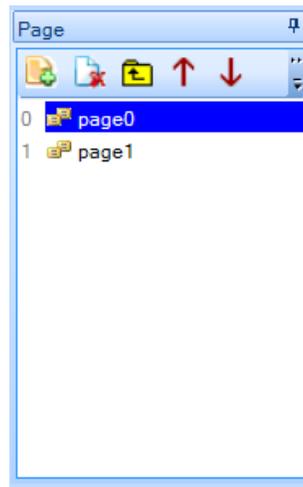


Fig. 72.- Creación de páginas de trabajo
Fuente: Autores

En la página 1 se establecerá la presentación del proyecto en donde se ubica un botón para ir a la siguiente página.

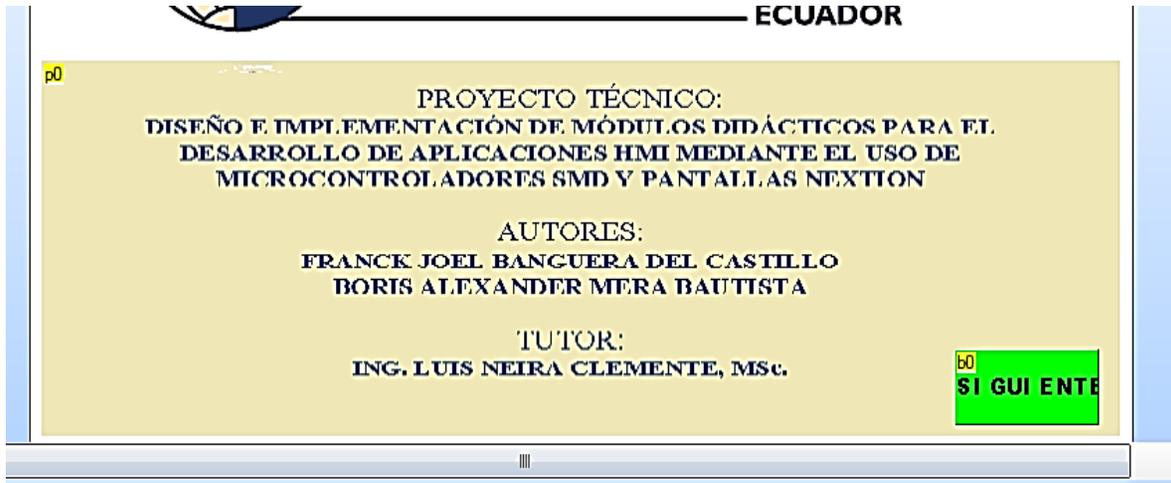


Fig. 73.- Página 1 en Pantalla Nextion
Fuente: Autores

En esta figura, realizamos la programación para que el botón pueda realizar la acción de enviarnos a la página 2.



Fig. 74.- Programación de botón Siguiete en Nextion
Fuente: Autores

Luego de haber configurado el botón siguiente en la página 1, nos toca ubicar los elementos en la página 2 que vamos a utilizar en esta pantalla para realizar la práctica como puertos de salida. Para esta figura se establecerán componentes como texto y botones los cuales se programarán para enviar valores al microcontrolador, los textos no necesitan programación ya que en los

atributos se puede modificar sus nombres, mientras que los botones se establecen los valores que enviarán al microcontrolador.

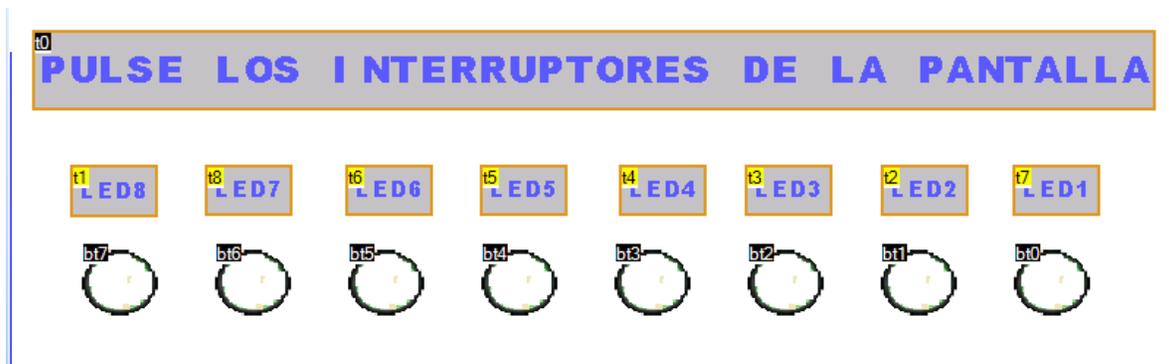


Fig. 75.- Pantalla de Puertos de Salida
Fuente: Autores

Al ubicarse cada botón se ubicará un timer para que pueda enviar la señal al micro, mediante la codificación siguiente:

```
if(bt7.val==0)
{
  print "0"
}else
{
  print "1"
}
if(bt6.val==0)
{
  print "0"
}else
{
  print "1"
}
if(bt5.val==0)
{
  print "0"
}else
{
  print "1"
}
if(bt4.val==0)
{
  print "0"
}else
{
  print "1"
}
if(bt3.val==0)
{
  print "0"
```

```
}else
{
  print "1"
}
if(bt2.val==0)
{
  print "0"
}else
{
  print "1"
}
if(bt1.val==0)
{
  print "0"
}else
{
  print "1"
}
if(bt0.val==0)
{
  print "0"
}else
{
  print "1"
}
```

Una vez presionado cada botón enviará al microcontrolador dicho evento y si es positivo enviará el 1 hasta que se indique que hay que apagar.

ANEXO 2: Solución propuesta de la práctica #2

Creamos un proyecto en Pic C Compiler y le asignamos un nombre que describa la práctica.

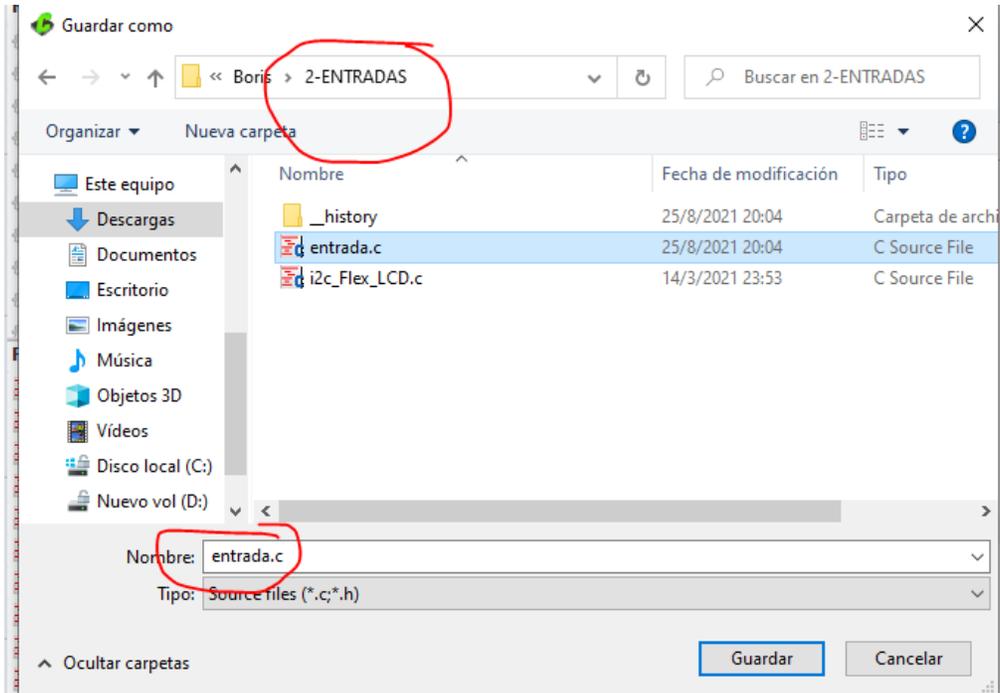


Fig. 76.- Creación de proyecto en PIC C Compiler
Fuente: Autores

Una vez creado el proyecto, detallamos las líneas de código predeterminadas las cuales permiten la inicialización del microcontrolador.

```
1 #include <18f8720.h>
2 #fuses xt, NOWDT, NOPROTECT, NOBROWNOUT, NOLVP, NOCPD, NOPUT, NOWRT
3 #use delay(clock=4000000)
4 #use i2c(Master, Fast=100000, sda=PIN_H1, scl=PIN_H2, force_sw)
5 #use rs232(baud=9600, xmit=PIN_G1, rcv=PIN_G2, TIMEOUT=500) //C6 -- C7
6 #include <i2c_Flex_LCD.c>
7
```

Fig. 77.- Códigos predeterminados de inicialización
Fuente: Autores

En la figura anterior se visualiza la línea de código 4 y 5 que permitirán la comunicación con la pantalla NEXTION y de ahí debemos ubicar las líneas de códigos que permitirán recibir los códigos enviados por el micro y visualizarlos en los LEDs como en la pantalla NEXTION.

```

19 void envio(void)
20 {
21     int cero=0;
22     int uno=1;
23     if(input(PIN_B0))
24         { printf("page1.bt0.val=%u",uno); printf("yÿÿ");}
25     else
26         { printf("page1.bt0.val=%u",cero);printf("yÿÿ");}
27     if(input(PIN_B1))
28         { printf("page1.bt1.val=%u",uno); printf("yÿÿ");}
29     else
30         { printf("page1.bt1.val=%u",cero);printf("yÿÿ");}
31     if(input(PIN_B2))
32         { printf("page1.bt2.val=%u",uno); printf("yÿÿ");}
33     else
34         { printf("page1.bt2.val=%u",cero);printf("yÿÿ");}
35     if(input(PIN_B3))
36         { printf("page1.bt3.val=%u",uno); printf("yÿÿ");}
37     else
38         { printf("page1.bt3.val=%u",cero);printf("yÿÿ");}
39     if(input(PIN_B4))
40         { printf("page1.bt4.val=%u",uno); printf("yÿÿ");}
41     else
42         { printf("page1.bt4.val=%u",cero);printf("yÿÿ");}
43     if(input(PIN_B5))
44         { printf("page1.bt5.val=%u",uno); printf("yÿÿ");}
45     else
46         { printf("page1.bt5.val=%u",cero);printf("yÿÿ");}
47     if(input(PIN_B6))
48         { printf("page1.bt6.val=%u",uno); printf("yÿÿ");}
49     else
50         { printf("page1.bt6.val=%u",cero);printf("yÿÿ");}
51     if(input(PIN_B7))
52         { printf("page1.bt7.val=%u",uno); printf("yÿÿ");}
53     else
54         { printf("page1.bt7.val=%u",cero);printf("yÿÿ");}
55 }

```

Fig. 78.- Codificación de envío de datos a la Pantalla Nextion
Fuente: Autores

Luego de ella se establecerá el main que permitirá activar los puertos como entrada y visualizar en el LCD (Liquid Cristal Display lo que en español es representación visual por cristal líquido) un mensaje de bienvenida mientras que entrará en un ciclo infinito hasta que se reinicie el proceso o se apague.

```

57 void main()
58 {
59     set_tris_b(0xFF);
60     set_tris_e(0x00);
61     lcd_init(0x4E,20,4);
62     enable_interrupts(GLOBAL);
63     enable_interrupts(INT_RDA2);
64     lcd_putc("    UNIVERSIDAD");
65     lcd_gotoxy(1,2);
66     lcd_putc("    POLITECNICA");
67     lcd_gotoxy(1,3);
68     lcd_putc("    SALESIANA");
69     lcd_gotoxy(1,4);
70     lcd_putc("    GUAYAQUIL");
71     delay_ms(1000);
72     lcd_putc("\fActive los pulsantes");
73     lcd_gotoxy(1,2);
74     lcd_putc("    del MODULO");
75 while(true)
76 {
77     valor=portb;
78     if(valor2!=valor){
79         valor2=valor;
80         porte=portb;
81         // printf("\n\rValor %u \n\r",valor2);
82         lcd_gotoxy(1,4);
83         lcd_putc("    ");
84         lcd_gotoxy(1,4);
85         printf(lcd_putc,"Puerto =%u",valor2);
86         envio();
87     }
88 }

```

Fig. 79.- Puerto como entrada y Mensaje de Bienvenida
Fuente: Autores

Una vez establecido el proyecto en PIC C Compiler, ahora programaremos la Pantalla Nextion para ello abriremos un proyecto en el software Nextion y ubicaremos un nombre representativo de la práctica.

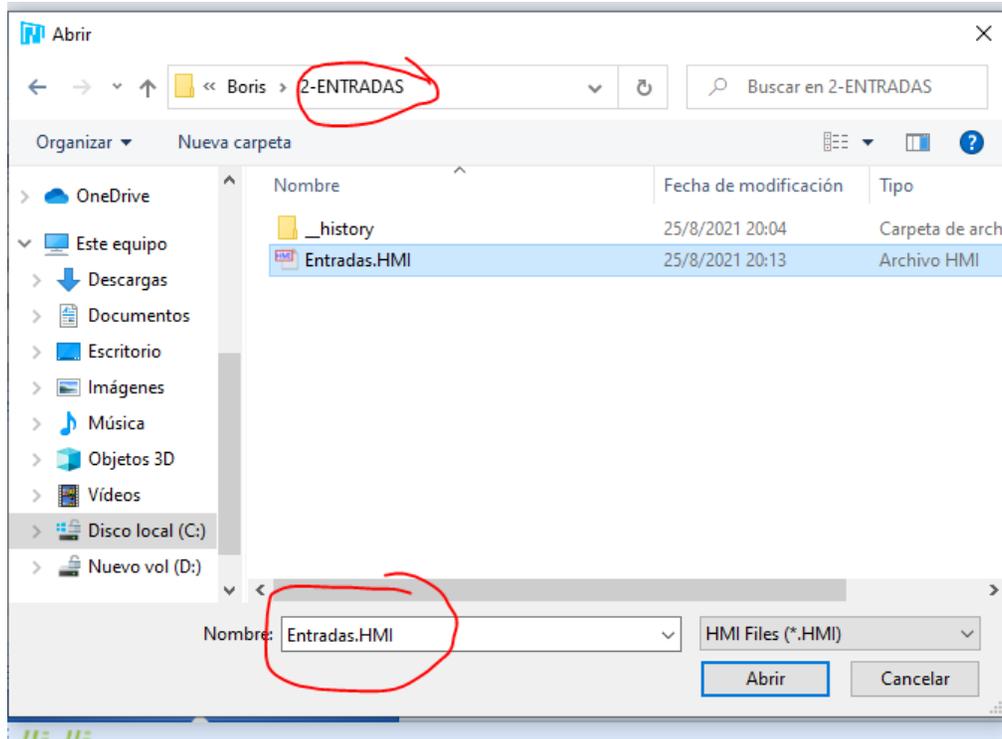


Fig. 80.- Creación de proyecto en Nextion
Fuente: Autores

Estableceremos dos páginas para trabajar en ellas, página 1 la presentación de la práctica y página 2 el desarrollo visual de la misma.

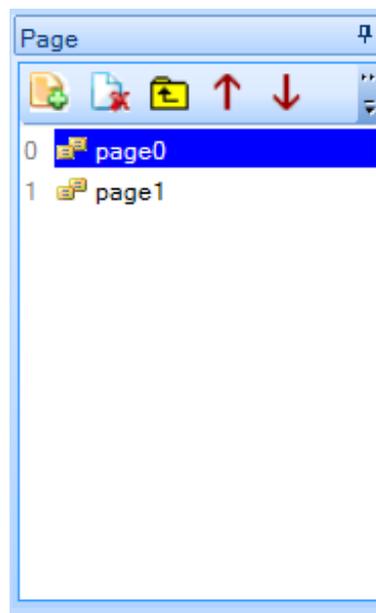


Fig. 81.- Creación de páginas de trabajo.
Fuente: Autores

En la página 1 se establecerá la presentación del proyecto en donde se ubica un botón para ir a la siguiente página.

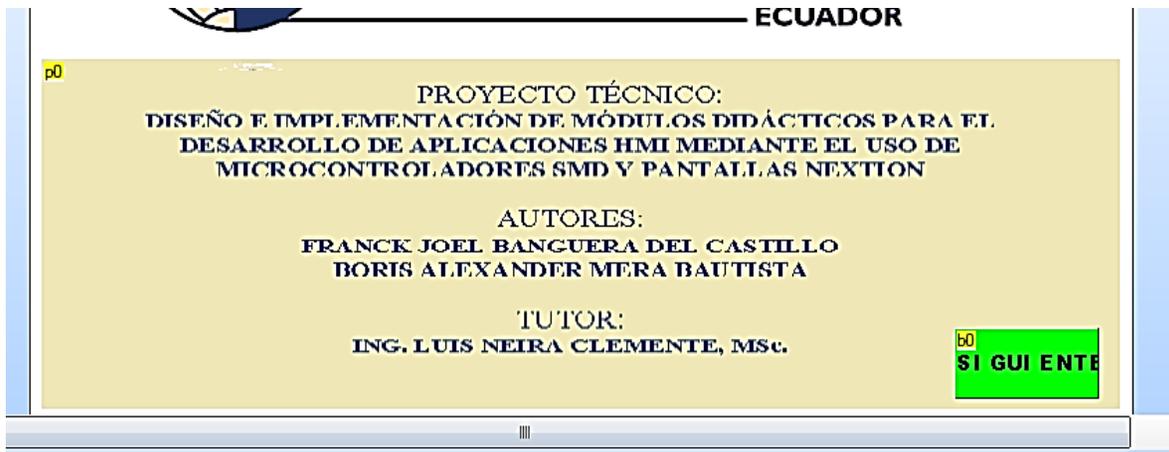


Fig. 82.- Página 1 en Pantalla Nextion
Fuente: Autores

En esta figura, realizamos la programación para que el botón pueda realizar la acción de enviarnos a la página 2.

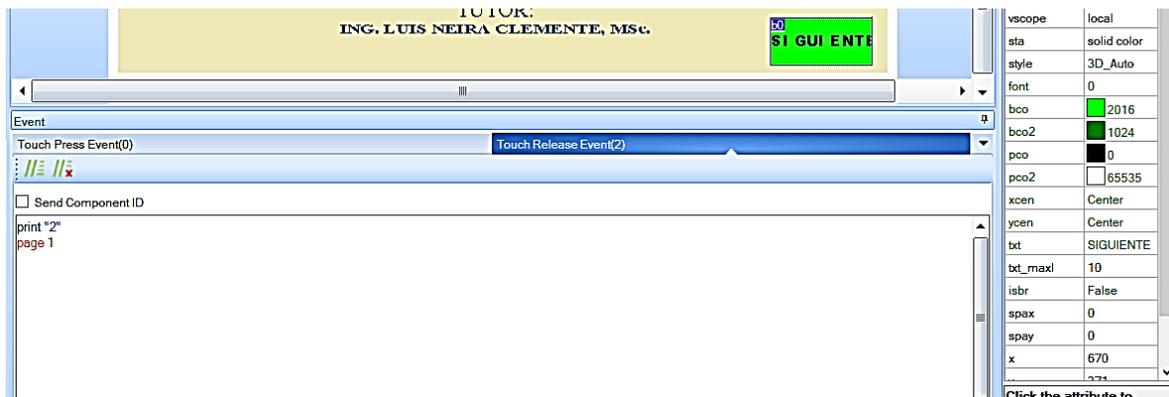


Fig. 83.- Programación de botón Siguiete en Nextion
Fuente: Autores

Luego de haber configurado el botón siguiente en la página 1, nos toca ubicar los elementos en la página 2 que vamos a utilizar en esta pantalla para realizar la práctica como puertos de entrada. Para esta figura se establecerán componentes como texto y botones los cuales se programarán para enviar valores al microcontrolador, los textos no necesitan programación ya que en los atributos se puede modificar sus nombres, mientras que los botones se establecen los valores que enviarán al microcontrolador.

t0 ACTIVE LOS INTERRUPTORES DEL MÓDULO



Fig. 84.- Pantalla de Puertos de Salida
Fuente: Autores

Al ubicarse cada botón se ubicará en el programa que botones deben encenderse, una vez encendido el microcontrolador mantendrá dicho evento hasta que se indique que hay que apagar. Una vez configurada la pantalla y definida la programación en Pic C Compiler todo se debe compilar y subir al módulo didáctico.

ANEXO 3: Solución propuesta de la práctica #3

Creamos un proyecto en Nextion y le asignamos un nombre que describa la práctica.

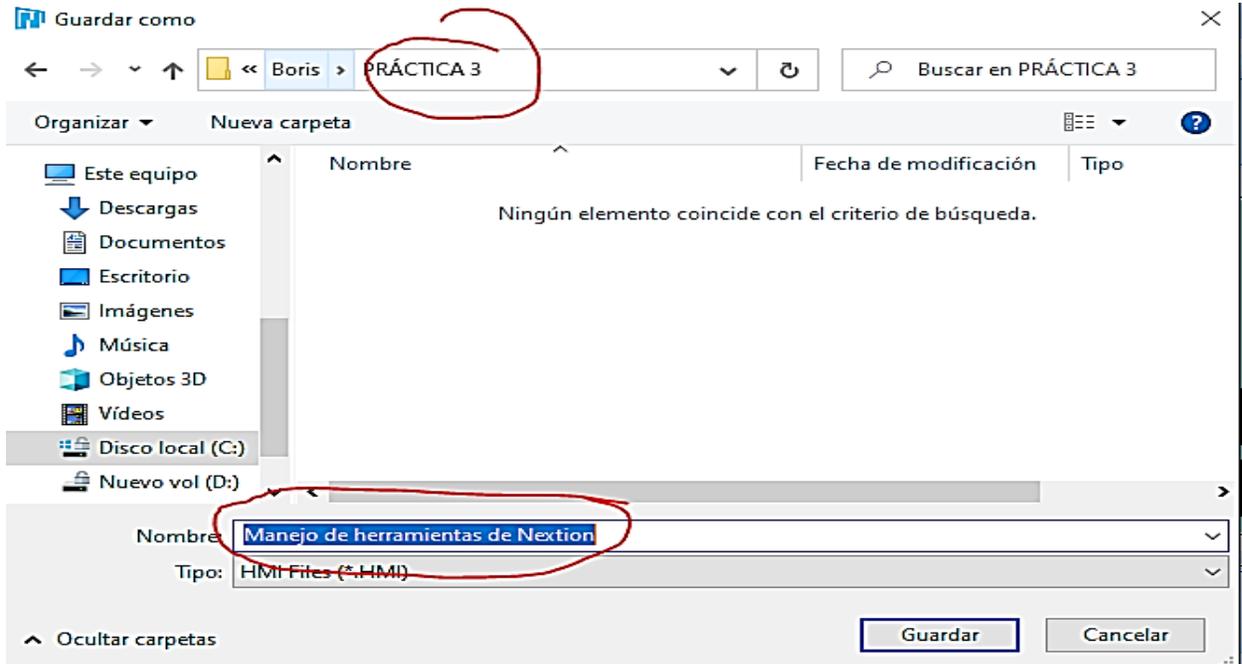


Fig. 85. Creación de proyecto en Nextion
Fuente: Autores

Una vez creado el proyecto estableceremos dos páginas para trabajar en ellas, la pagina 1 es la presentación de la práctica y la página 2 la configuración del proceso industrial que hayamos definido en este caso el ozonizador.

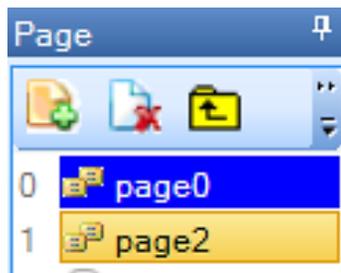


Fig. 86.- Creación de páginas de trabajo
Fuente: Autores

En la página 1 ubicaremos elementos como texto, botones, código QR y desplazamiento de texto; una vez ubicados esos elementos estableceremos los siguientes nombres:

- Elemento de texto → Generador de Ozono (O₃)
- Elemento Botón 1 → Configurar
- Elemento Botón 2 → Activar
- Elemento QR código
- Elemento Desplazamiento de texto → Estado



Fig. 87.- Elementos de página 1 en Práctica 3
Fuente: Autores

Luego de establecer los nombres de cada uno de los elementos, empezaremos a configurar el botón de configurar, este botón deberá enviar a la página 2 en donde se establecen algunos parámetros.



Fig. 88.- Codificación de botón Configurar
Fuente: Autores

Una vez establecida dicha codificación, configuramos el botón de Activar que envía al microcontrolador los valores para que así se muestre en el elemento número dentro del desplazamiento de texto.

```

if(b1.txt=="ACTIVAR")
{
  b1.txt="DESACTIVAR"
  print "AAAAAA"
}else
{
  b1.txt="ACTIVAR"
  print "BBBBBB"
  n0.bco=61309
  n0.val=0
}

```

Los dos botones de la página 1 se definen como punto base del funcionamiento de esta práctica, por ello los valores que ellos envíen al microcontrolador se establecen de forma hexadecimal mientras ocurre alguno de los eventos, ya sea el de configurar o el de activar; el QR código se configura únicamente con los atributos y se escribe la página web donde se dirige la información.

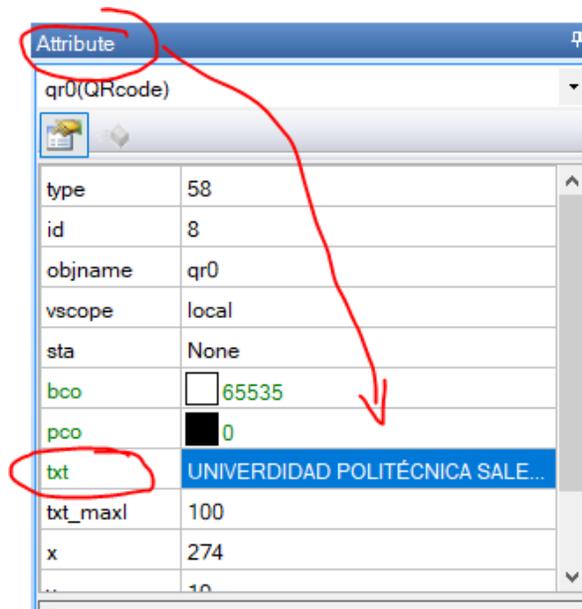


Fig. 89.- Configuración QR código
Fuente: Autores

En el desplazamiento de texto ubicamos un timer para que muestre su desarrollo mientras se envían los datos al microcontrolador y así reciba lo que le el microcontrolador envía.

```

va0.val=va0.val+1
if(va0.val==2)
{
  print "CCCCCC"
  tm0.en=0
}

```

Terminando con la configuración de la página 1 de la práctica, ahora codificaremos la segunda página la cual en el momento que presionamos el botón Configurar de la página 1 nos envía a la página 2 y ahí ubicaremos los siguientes elementos:

- 4 textos
- 2 botones
- 2 slider

En la configuración de los elementos ubicaremos lo siguiente:

- Texto 0 → tiempo ON
- Texto 1 → tiempo OFF
- Texto 2 → min
- Texto 3 → min
- Botón 1 → Aceptar
- Botón 2 → cancelar

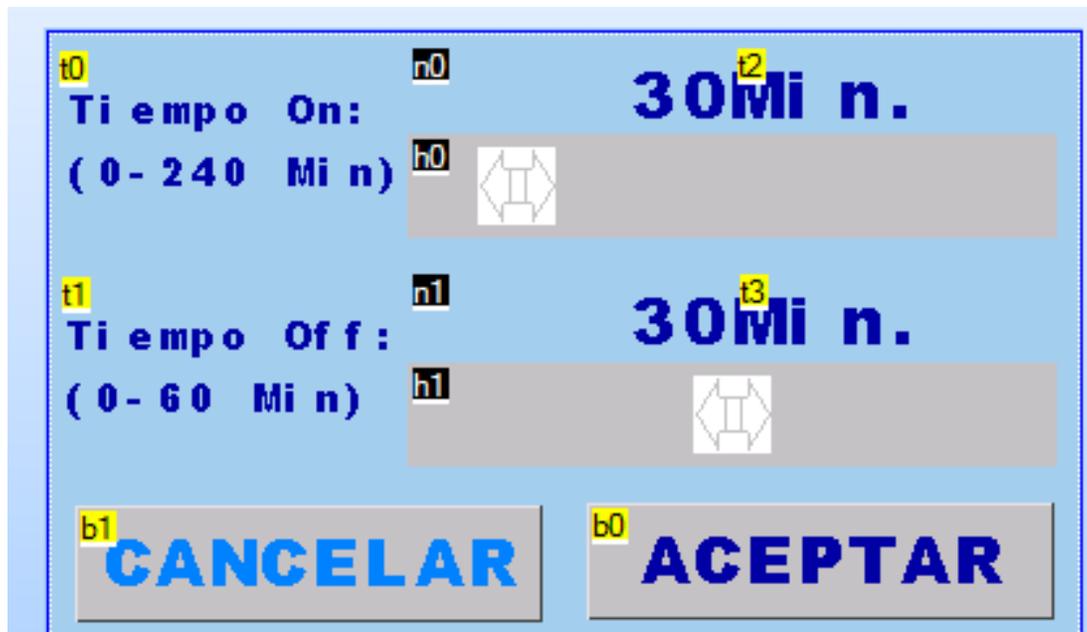


Fig. 90.- Elementos página 2 de práctica 3
Fuente: Autores

La codificación de las sliders para el tiempo de encendido y apagado se realiza en los eventos y así dicho valor se envía al microcontrolador, en cuanto a su movimiento se visualiza en el valor numérico del texto de la parte superior de las sliders:

```
n0.val=h0.val  
n1.val=h1.val
```

Para concluir se presiona el Aceptar que se configura de igual manera que en la página 1 con la diferencia de que al momento de presionar nos envía de la página 2 a la página 1 para iniciar el proceso con la configuración ya programada.

```
print "n0"  
prints n0.val,1  
print "n1"  
prints n1.val,1  
page 0
```

ANEXO 4: Solución propuesta de la práctica #4

Creamos un proyecto en Nextion y le asignamos un nombre que describa la práctica.

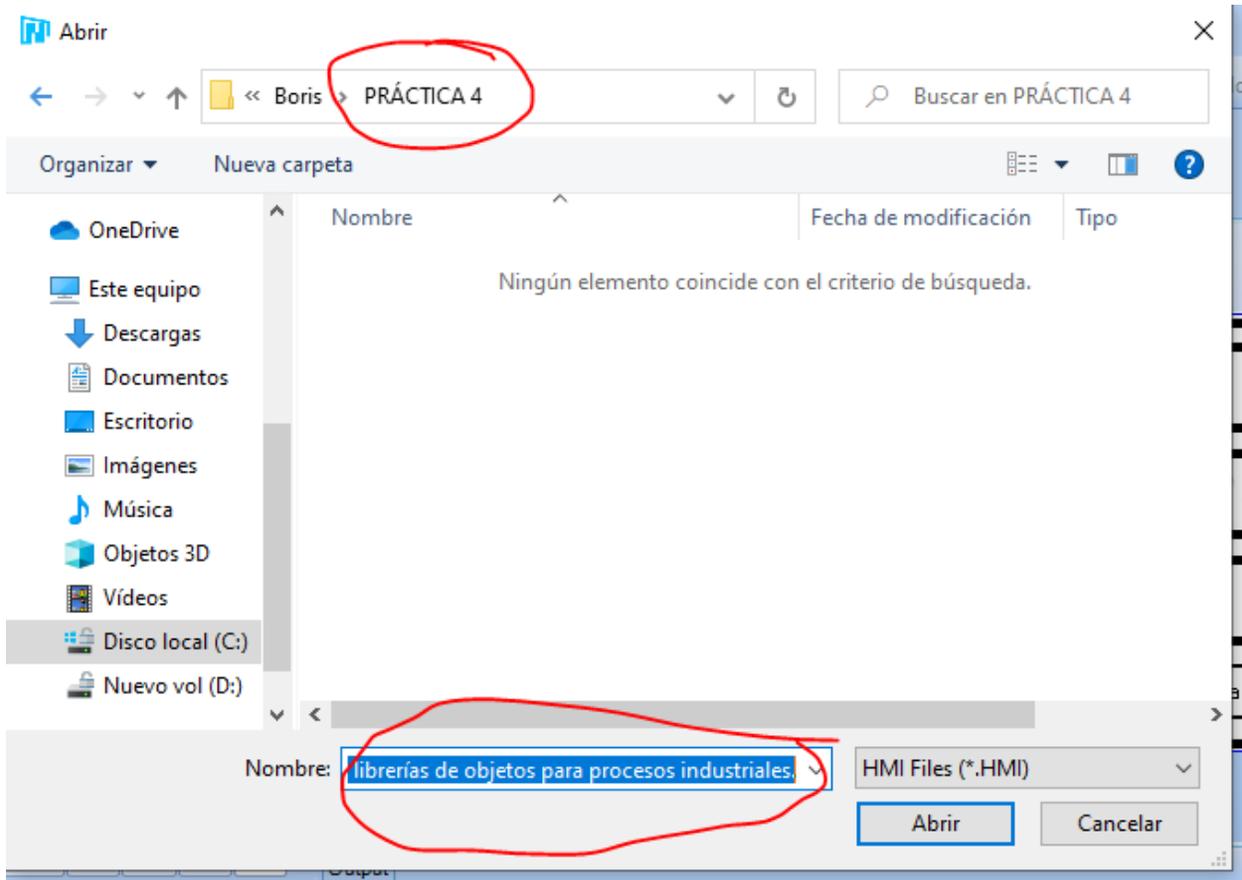


Fig. 91.- Creación de proyecto en Nextion
Fuente: Autores

Una vez creado el proyecto estableceremos una página para trabajar en ella, se realiza la configuración y creación de las librerías para los procesos industriales.

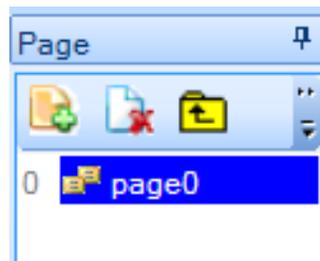
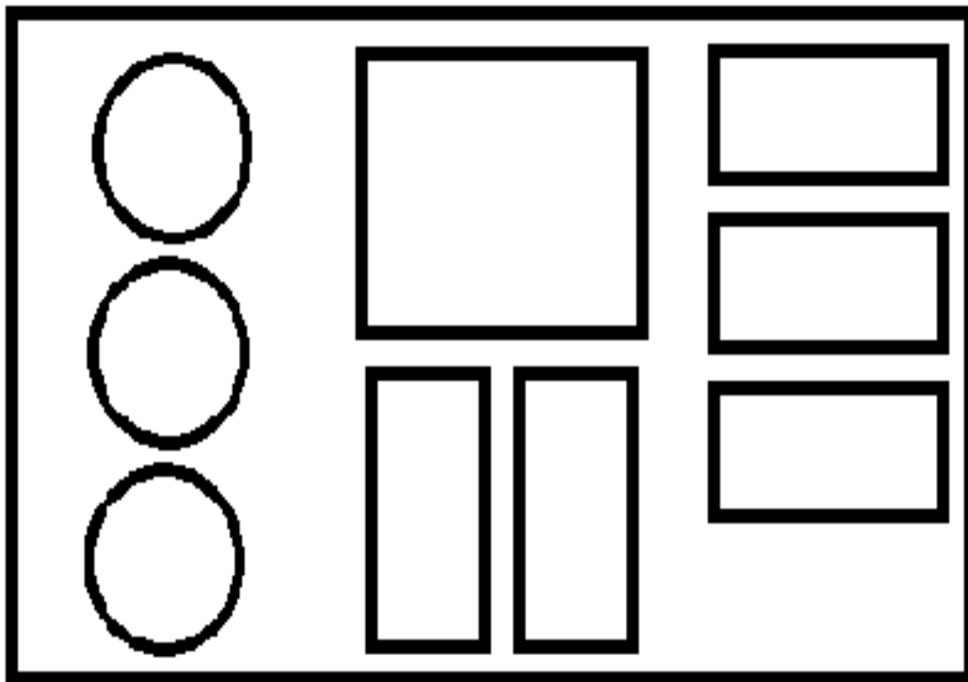


Fig. 92.- Creación de página de trabajo
Fuente: Autores

En la página 1 ubicaremos elementos que sirvan de soporte a las librerías que creamos; una vez ubicados esos elementos estableceremos los siguientes nombres:

- Elemento de checkbox
- Elemento Scrolling text
- Elemento Picture
- Elemento QR código
- Elemento Slider
- Elemento Botón
- Elemento Botón estado dual



*Fig. 93.- Base de librería
Fuente: Autores*

Las imágenes que se adjuntan al espacio establecido en la Nextion deben ser exacta con las medidas de los atributos que se establecen, como:

- Elemento de checkbox medida de 48x172
- Elemento Scrolling text medida de 287x192
- Elemento Picture medida de 18x85

- Elemento QR código medida de 100x161
- Elemento Slider medida de 217x135
- Elemento Botón medida de 298x80
- Elemento Botón estado dual medida de 295x23

Estas medidas están en milímetros y se trabajan en un software en línea sea cual sea, el que se utilizó para la creación de las librerías fue befunky.com, estas imágenes también pueden ser pixeladas si se desea trabajar con ello y todo se visualiza en los atributos.

Attribute	
h0(Slider)	
hig	25
bco	 1024
pco	 63488
val	50
maxval	100
minval	0
x	217
y	135
w	36
h	83

Fig. 94.- Dimensiones en milímetros o pixeles
Fuente: Autores

Luego de tener las imágenes del tamaño que se deseen o se establezcan en la pantalla, se procede a subirlas al submenú de toolbox en donde se utilizan las imágenes como parte de la programación principal.

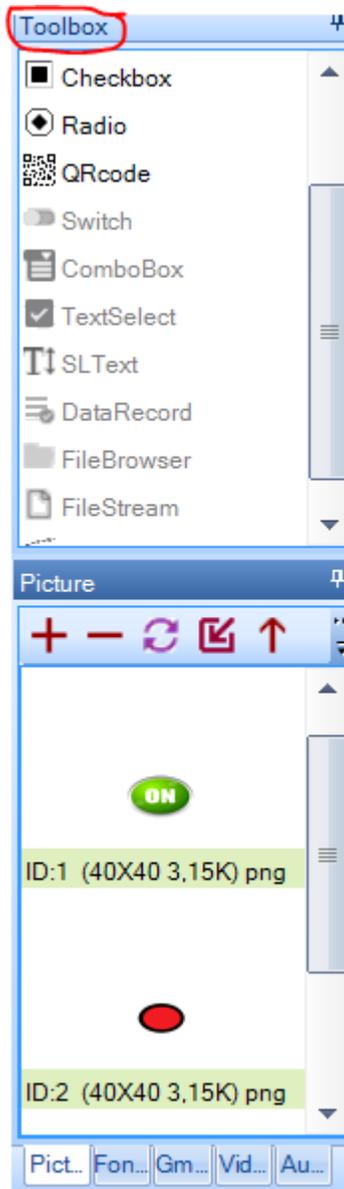


Fig. 95.- Submenú Toolbox
Fuente: Autores

Acá elemento es arrastrado a la ventana principal y se programa de acuerdo con lo que necesita realizar, se establecen la codificación de los botones:

Se visualiza en el botón 1 el cambio que debe realizar cada que se presione y un led realizará su funcionamiento en el encendido o apagado del mismo; ubicando desde los atributos que acción deben generar y que imagen se debe mostrar como el cambio de color verde a rojo o viceversa.

```
if(b1.pic==4)
{
```

```
b1.pic=5
b1.pic2=4
t2.pic=2
}else
{
b1.pic=4
b1.pic2=5
t2.pic=3
}
```

En el botón de estado dual se establece la codificación de acuerdo con su estado de funcionamiento el cual es como un switch o interruptor, este ubica la imagen de acuerdo al estado del botón.

```
if(t3.pic==3)
{
t3.pic=2
}else
{
t3.pic=3
}
```

Para los demás elementos se establece la codificación por medio de los atributos ya sea si debe enviarnos a una página (QR código) o el trabajo del scrolling text para mostrar algún mensaje, las otras librerías se codifican de acuerdo con su funcionamiento en una práctica como envío o recepción en el microcontrolador.

ANEXO 5: Solución propuesta de la práctica #5

En esta práctica se realizan los pasos anteriores de las prácticas 2, 3 y 4 en donde se agregan páginas de trabajo y elementos, en este caso realizaremos la creación de 6 páginas para la elaboración de la práctica.

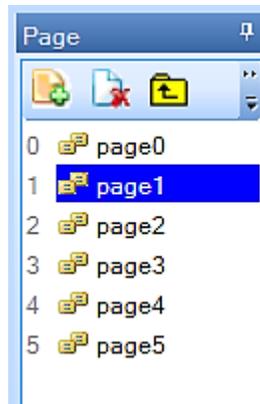


Fig. 96.- Creación de páginas de trabajo
Fuente: Autores

En la creación de las páginas se ubican imágenes principales como parte del fondo de trabajo, es por ello por lo que en cada una de las páginas se establecen elementos bases como botones y textos para su funcionamiento de cada una de ellas.

Cada imagen principal del ambiente para esta aplicación en el HMI debe establecerse por las dimensiones de la pantalla, como se visualiza en la siguiente figura.

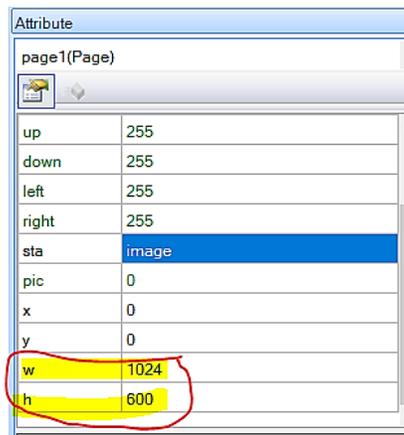


Fig. 97.- Dimensiones de imágenes para pantalla HMI
Fuente: Autores

En la página 1 a la página 5 se establece las mismas dimensiones para ubicar las imágenes que se encuentran en ToolBox, como se visualiza a continuación.



*Fig. 98.- Imágenes con dimensiones de pantalla HMI
Fuente: Autores*

Las dimensiones pueden variar de acuerdo con la imagen que desea ubicar, pero en este caso no varía; ubicamos botones en ciertas partes de la imagen para indicar una función, por ejemplo:

- División de una casa
- Secciones de un cuarto en específico de la casa

Al especificar cada sección, se programa el movimiento de cada uno de los botones, en este caso algunos mueven entre las páginas creadas y otros se programan con sonido de acuerdo con la pantalla HMI en su serie.



Fig. 99.- Ambientes de casa
Fuente: Autores

La figura 99 muestra los ambientes que se programaron y a que página deben enviarse para mostrar los elementos o sectores que se deben pronunciar en el idioma que hemos establecido, la codificación es la misma para los 4 ambientes principales con la diferencia de envío de página.

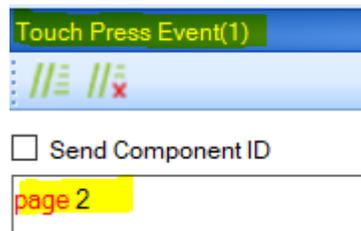


Fig. 100.- Codificación de botones en ambientes principales
Fuente: Autores

Cada ambiente tiene en su codificación a la página que debe enviar y este se da una vez que el botón sea presionado como un pulsador (touch):

- Living room envía a la página 2.
- Bedroom envía a la página 5.
- Yard envía a la página 4.
- Garage envía a la página 3.

Dentro de cada página creado en donde se muestra una imagen del ambiente más específico, se establece botones que sirven como soporte para la configuración del audio y así una vez presionado el botón en la pantalla HMI se escuche la pronunciación correcta de la palabra o el espacio, cada botón se programa en el atributo en donde se agrega el audio de la misma manera que la imagen en formato wav en donde no es necesario codificar a menos de que dicho sonido o audio se debe enviar al microcontrolador para el Buzzer.



Fig. 101.- Audios en librería principal
Fuente: Autores

El formato establecido es wav que se lo puede convertir en cualquier software libre o en línea y se agrega al botón en su funcionamiento, teniendo en cuenta que en la pestaña de atributos para visualizar un mejor manejo podemos cambiar el color del botón si se genera una acción diferente

a la establecida por la fábrica, en la figura 102 se puede visualizar en el toolbox el audio 15 se ubica en el atributo del botón 4 en la sección de vid para su funcionamiento.



Fig. 102.- Inserción de audio en botón a través de atributos
Fuente: Autores

Para un mejor trabajo y evitar que el programa se vuelva cíclico, se programa un botón de regreso (return) que permite detener la página en la que se está y así poder cambiar de ambientes según sea el agrado del programador, este botón se codifica en cada página (importante establecer si es por touch o switch) y se envía a la página principal en donde se encuentran los 4 ambientes de la casa.

De ahí se compila y se sube a la pantalla HMI Nextion presionando los iconos de Compile y Upload en la barra de herramientas, teniendo en cuenta que el módulo debe estar en programación de pantalla.



Fig. 103.- Iconos de compilación y Carga de programación de pantalla HMI
Fuente: Autores

ANEXO 6: Solución propuesta de la práctica #6

Para la practica 6 se visualiza el funcionamiento del simulador que incorpora el Nextion Tech software libre de la empresa que diseña las pantallas HMI, este simulador permite seleccionar los diferentes tipos de versiones de la pantalla y sus pulgadas para que sea más interactiva, lo que tenemos que hacer es ubicar una práctica cualquiera, en este caso es la práctica 5 sobre el manejo de pantallas y en donde se visualiza el funcionamiento correcto de la misma con sonido.



Fig. 104.- Barra de herramientas
Fuente: Autores

En la figura anterior, se puede visualizar en la parte derecha el icono de DEVICE el cual nos despliega un menú en donde se configura la dirección del Display, el tipo del dispositivo (modelo de la pantalla HMI) y la protección de contraseña del proyecto si se desea tal como se visualiza en la figura siguiente.

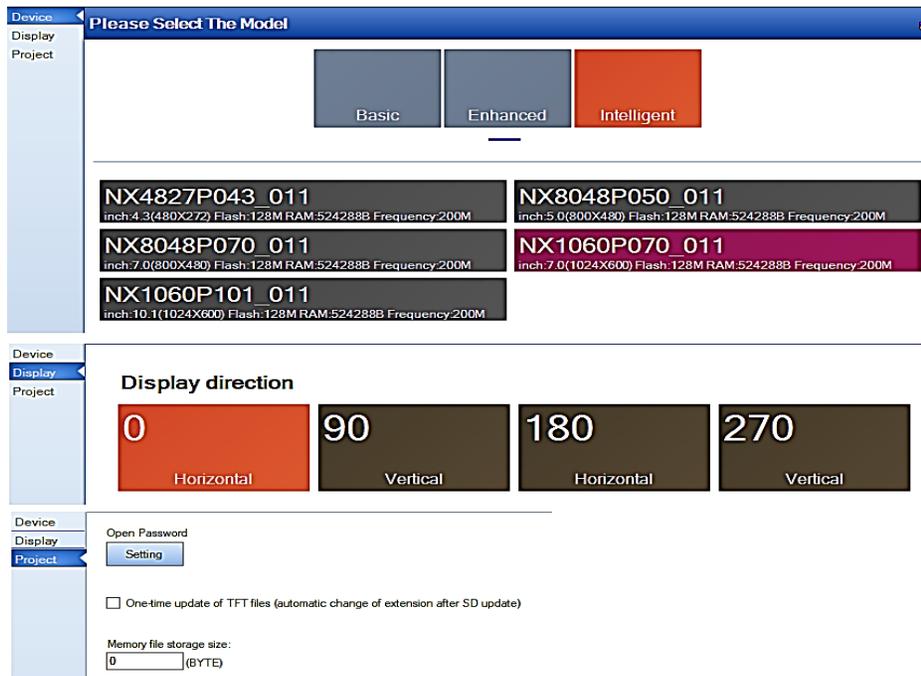


Fig. 105.- Menú de DEVICE en simulador
Fuente: Autores

En esta figura 105 se visualiza el tipo de modelo de HMI que se desea probar o simulador, como se realiza la simulación de la practica 5 nos conviene una inteligente de cualquier serie ya que

incluye sonido aparte del touch capacitivo o resistivo. Luego de establecer el tipo de modelo, la dirección y la clave del proyecto buscamos en la misma barra de herramientas el icono DEBUG que se encuentra entre compilación y cargar.



Fig. 106.- Icono Debug inicia la simulación
Fuente: Autores

Este icono DEBUG permite compilar el programa realizado y nos abre una ventana de simulación para operarla de acuerdo con los botones programados o sino por comando que se puedan enviar desde el microcontrolador a la pantalla a través de la comunicación Rx y Tx.

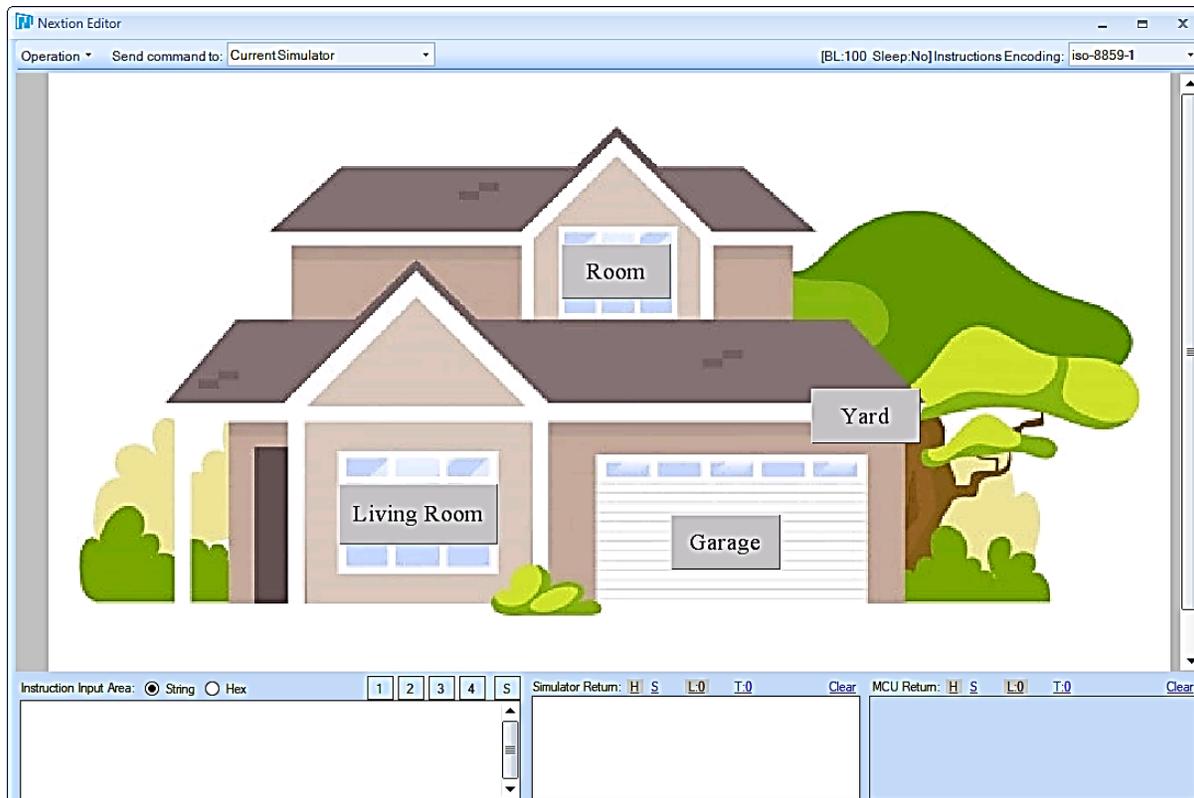
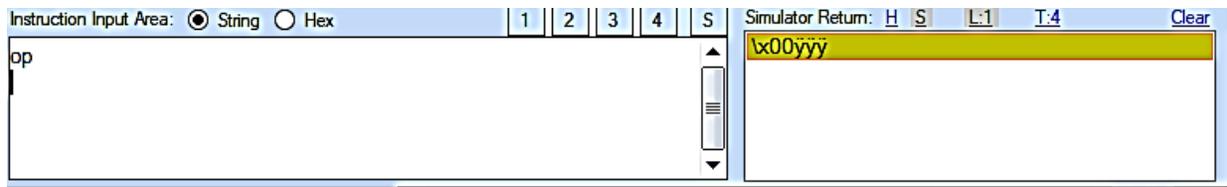


Fig. 107.- Ventana de Simulación
Fuente: Autores

En la sección inferior, se puede establecer códigos ya sean como datos string o hexadecimal (similar al envío de datos de un microcontrolador) y visualizarlos en su salida según datos string o hexadecimal (para envío desde la pantalla HMI al microcontrolador), en la figura 108 se puede

visualizar que valores se ingresan (parte izquierda de la figura) y cuáles son los valores que envía el HMI al microcontrolador (parte derecha de la figura).



*Fig. 108.- Datos como entrada y salida en la simulación
Fuente: Autores*

De ahí el manejo de dicho simulador es similar a Proteus, el cual sirve para visualizar el funcionamiento de lo que estamos diseñando o plasmando para evitar cometer errores.

ANEXO 7: Solución propuesta de la práctica #7

Creamos un proyecto en Nextion y le asignamos un nombre que describa la práctica.

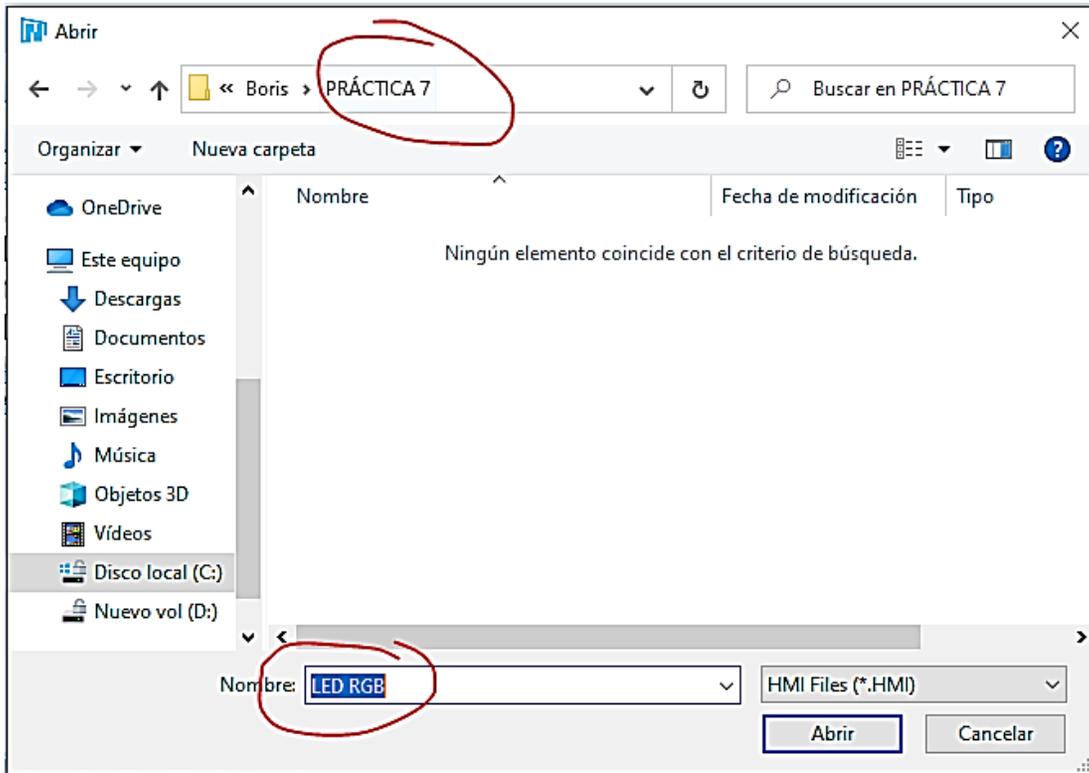


Fig. 109.- Creación de proyecto en Nextion
Fuente: Autores

Una vez creado el proyecto estableceremos dos páginas para trabajar en ellas, la página 1 es la presentación de la práctica y la página 2 la configuración del proceso industrial que hayamos definido en este caso el ozonizador.

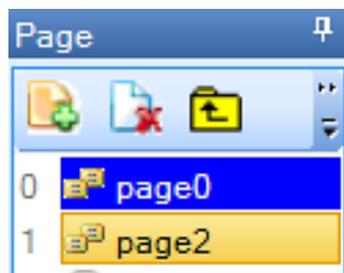


Fig. 110.- Creación de páginas de trabajo
Fuente: Autores

Se ubica el tipo de comunicación, en este caso I2C que trabajan e diferentes pines y la pantalla HMI como maestro.

```
#use i2c(Master, Fast=100000, sda=PIN_HI, scl=PIN_H2, force_sw)
#use rs232(baud=9600, xmit=PIN_G1, rcv=PIN_G2, TIMEOUT=500) //CG -- C7
```

Declaramos el PWM del microcontrolador en donde se respalda el funcionamiento de la programación de la pantalla HMI para el encendido de los leds RGB.

```
#USE PWM(OUTPUT=PIN_G0, FREQUENCY=10kHz, DUTY=25, stream=red) //PWM en G0, CCP3
#USE PWM(OUTPUT=PIN_G3, FREQUENCY=10kHz, DUTY=25, stream=green) //PWM en G3, CCP4
#USE PWM(OUTPUT=PIN_G4, FREQUENCY=10kHz, DUTY=25, stream=blue) //PWM en G4, CCP5
```

Se incluye librería creada de LCD para su visualización en los datos recibidos por la pantalla HMI y las variables a utilizar en el microcontrolador para establecer la comunicación continua con los datos que envía la HMI.

```
#include <i2c_Flex_LCD.c>
#byte porte=0xf84
int s1,s2,s3;
int inicial=50;
int16 z;
int16 valor,valorl;
int16 rojo,verde,azul;
```

Declaramos la interrupción interna RDA para que cuando se envíe un valor lo muestre enseguida, en donde se establece parámetros una vez que en la pantalla HMI haya algún cambio, dándole así al PWM el porcentaje para mostrar dicho cambio.

```
#INT_RDA2
void RDA_isr2()
{
  disable_interrupts(INT_RDA2);
  s1=getc(); //h
  s2=getc(); //0,1,2
  s3=getc(); // valor 0-255
  z=t*s3;
  valorl= z*10;
  if(z<= 1)
    valor=995;
  else if(z>95)
```

```

    valor=2;
else
    valor=(1000-valor);

if(s1==104 && s2==48)
{ rojo=valor; lcd_gotoxy(1,2); printf(lcd_putc"ROJO= %3d",s3); printf(lcd_putc" PWM=%4ld",valor); }

if(s1==104 && s2==49)
{ verde= valor; lcd_gotoxy(1,3); printf(lcd_putc"VERDE= %3d",s3); printf(lcd_putc" PWM=%4ld",valor);}

if(s1==104 && s2==50)
{ azul=valor; lcd_gotoxy(1,4); printf(lcd_putc"AZUL= %3d",s3); printf(lcd_putc" PWM=%4ld",valor);}

    enable_interrupts(INT_RDA2);
}

```

En el Main se establece el funcionamiento y corrido del programa principal, en donde se deben activar todas las interrupciones y lo que se busca visualizar en el LCD.

```

void main()
{
int16 vi=500;
set_tris_b(0xFF);
set_tris_e(0x00);
lcd_init(0x4E,20,4);
porte=0;
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_RDA2);

lcd_putc(" UNIVERSIDAD");
lcd_gotoxy(1,2);
lcd_putc(" POLITECNICA");
lcd_gotoxy(1,3);
lcd_putc(" SALESIANA");
lcd_gotoxy(1,4);
lcd_putc(" GUAYAQUIL");
delay_ms(1000);
lcd_putc("\fActive los SLIDER");

    lcd_gotoxy(1,2); printf(lcd_putc"ROJO= %u",inicial); printf(lcd_putc" PWM= %ld",vi);
    lcd_gotoxy(1,3); printf(lcd_putc"VERDE= %u",inicial); printf(lcd_putc" PWM= %ld",vi);
    lcd_gotoxy(1,4); printf(lcd_putc"AZUL= %u",inicial); printf(lcd_putc" PWM= %ld",vi);
while(true)
{

```

```

if( rojo>= 985)
    pwm_off(red);
else {
    pwm_on(red);
    pwm_set_duty_percent(red,rojo);}

if( verde>= 985)
    pwm_off(green);
else{
    pwm_on(green);
    pwm_set_duty_percent(green,verde);}

if( azul>= 985)
    pwm_off(blue);
else{
    pwm_on(blue);
    pwm_set_duty_percent(blue,azul);}
}
}

```

Cada que se presione la slider de cualquier color, este se visualiza en el encendido del circuito que se diseñó con el Led RGB y su dato se visualiza en el LCD para su mayor entendimiento.

ANEXO 8: Solución propuesta de la práctica #8

Creamos un proyecto en Nextion y le asignamos un nombre que describa la práctica.

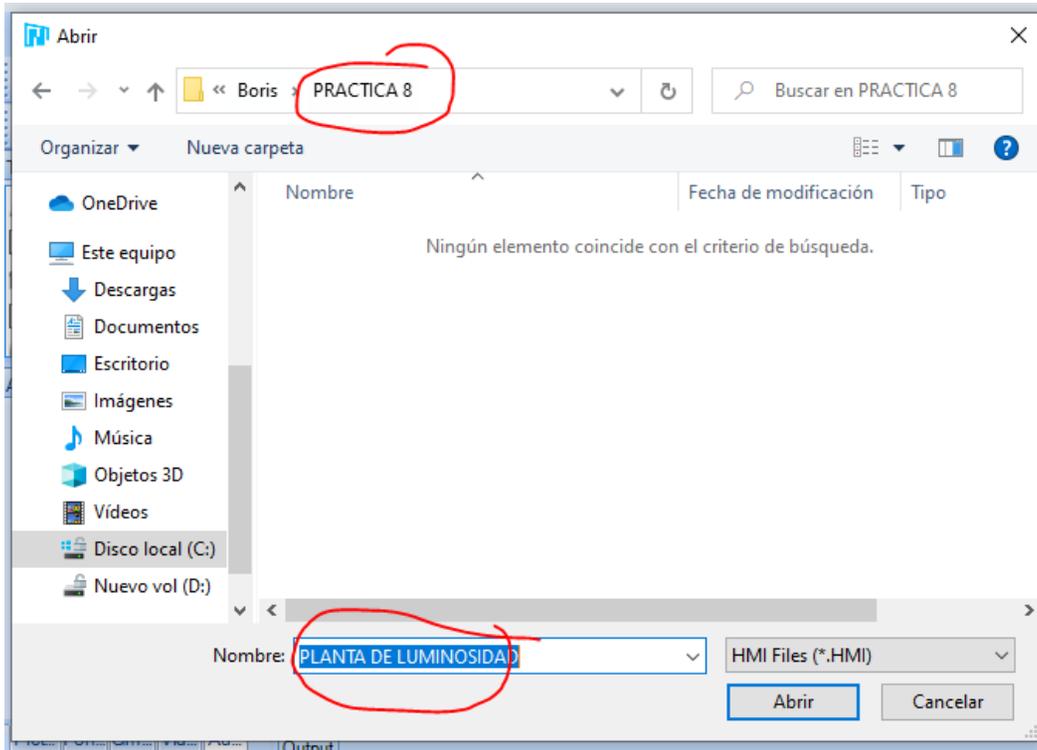


Fig. 112.- Creación de proyecto en Nextion
Fuente: Autores

Una vez creado el proyecto estableceremos dos páginas para trabajar en ellas, la página 1 es la presentación de la práctica y la página 2 la configuración del proceso industrial que hayamos definido en este caso el uso de la planta de control PID de luminosidad.

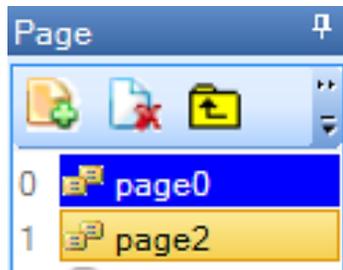


Fig. 113.- Creación de páginas de trabajo
Fuente: Autores

Al crear la carpeta y el archivo de la práctica 8 se crean 2 páginas para su funcionamiento, la página 1 sirve como presentación y la página 2 como estructura para establecer parámetros de labor de la práctica.

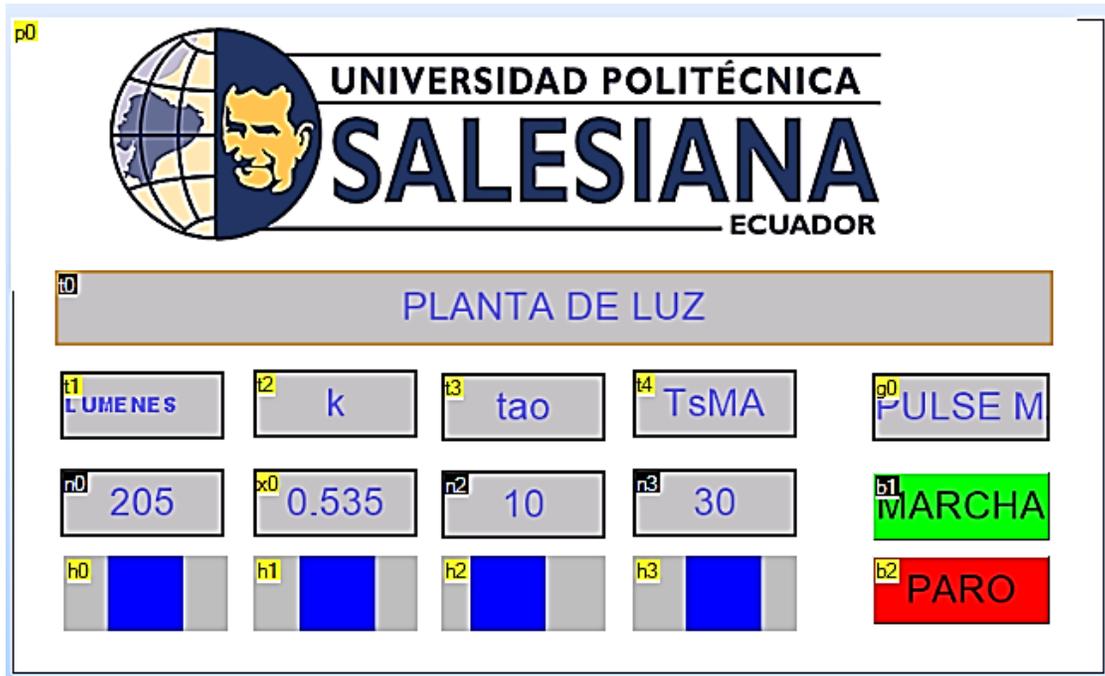


Fig. 114.- Estructura de página 2
Fuente: Autores

En la figura anterior se visualiza la estructura de trabajo de esta práctica, en donde ubicamos los siguientes elementos: 5 textos, 4 sliders, 4 números, 3 botones y 1 imagen, los cuales se establecen para el envío de datos al microcontrolador mediante la combinación de ellos y de sus valores ya configurados. Estos elementos permiten establecer la configuración base que se envía al microcontrolador y a su vez se puede modificar mientras está encendida la práctica, todos los cambios se pueden visualizar en el LCD el cual indica el crecimiento de dicho valor hasta llegar al seteo establecido que es la primera columna de la figura 114, mediante la codificación:

```
n0.val=h0.val
print "h0"
print h0.val
```

Para los valores que están en la columna 2 a la 4 es configurar el PID discreto y su respuesta para llegar al seteo establecido

```
x0.val=h1.val  
print "h1"  
print h1.val
```

```
n2.val=h2.val  
print "h2"  
print h2.val
```

```
n3.val=h3.val  
print "h3"  
print h3.val
```

Estas variables que se detallan a través de los atributos de cada uno de los elementos tienen dos funciones:

- La primera de mostrar en los elementos de número los datos enviados por las sliders
- La segunda de enviar dichos valores o variables (dependiendo del caso) al microcontrolador.

En el caso de los botones de marcha y paro, una vez presionados envían la información al scrolling text mediante cadena de caracteres sea este para marcha o paro.

```
print "b01"  
g0.txt=" PROCESO EN MARCHA"
```

```
print "b00"  
g0.txt=" PROCESO DETENIDO"
```

Una vez establecida la configuración, la creación y el diseño de nuestro HMI pasamos a codificar nuestra práctica con el microcontrolador para que su comunicación sea continua y fluida con los datos que se tienen desde la pantalla HMI.

Establecemos líneas de código básica de funcionamiento del microcontrolador:

```
#include <18f8720.h>  
#DEVICE ADC=10  
#fuses xt  
#use delay(clock=4000000)
```

Se ubica el tipo de comunicación, en este caso I2C que trabajan en diferentes pines y la pantalla HMI como maestro.

```
#use i2c(Master,Fast=100000, sda=PIN_H1, scl=PIN_H2,force_sw)  
#use rs232(baud=9600, xmit=PIN_G1,rcv=PIN_G2,TIMEOUT=500) //G6 -- G7
```

Se incluye librería creada de LCD para su visualización en los datos recibidos por la pantalla HMI y las variables a utilizar en el microcontrolador para establecer la comunicación continua con los datos que envía la HMI.

```
#include <i2c_Flex_LCD.c>
```

```

#byte porte=0xf84
int s1,s2,s3,s4,marcha;
float R=205.0,e=0.0,e_1=0.0,e_2=0.0,u=0.0,u_1=0.0;
float kp,ti,td,q0,q1,q2,T=0.1;
float k=0.535,tao=10.0,theta=1.0,kw;
float Wn,P1,P2;
float TsMA=30;

```

```

int16 control,yM;
int16 cont;
int Timer2,Pascalcr;

```

```

void dis(void);

```

Declaramos la interrupción interna RDA para que cuando se envíe un valor lo muestre enseguida, en donde se establece parámetros una vez que en la pantalla HMI haya algún cambio, dándole así el porcentaje para mostrar dicho cambio.

```

#INT_RDA2

```

```

void RDA_isr2()
{
    s4=0;
    kw=0;
    s1=getc(); //h
    s2=getc(); //0,1,2
    s3=getc(); // valor 0-255
    s4=getc();

    if(s1==98 && s2==48 && s3==49) //codigo ascii de "b"=98 y de "1" es 49
        { marcha=1; }
    if(s1==98 && s2==48 && s3==48) //codigo ascii de "b"=98 y de "0" es 48
        { marcha=0; }
    if(s1==104 && s2==48) //codigo ascii de "h"=104 y de "0" es 48 // seteo la velocidad R
        { R=s3;
          kw=s4;
          if(kw!=0)
              {R=R+(256*kw); }
        }
    if(s1==104 && s2==50) //codigo ascii de "h"=110 y de "3" es 51
        { tao=s3; }
    if(s1==104 && s2==49) //codigo ascii de "h"=110 y de "4" es 52
        { k=s3;
          kw=s4;
          if(kw!=0)

```

```

        {k=k+(256*kw); }
        k=k/1000;
    }
    if(s1==104 && s2==51) //codigo ascii de "h"=104 y de "0" es 48
        { TsMA=s3; }
}

```

La configuración del PID discreto se hace mediante el diseño de asignación de dos polos reales y teniendo en cuenta la variable que busca llegar al seteo oficial, que en este caso es la luminosidad del led RGB donde se establece un color para visualizar su cambio de intensidad.

```

void dis(void)
{
    //***** //
    //***** DISEÑO POR ASIGNACIÓN DE 2 POLOS REALES ***** //
    //***** //

        // TsMA Tiempo deseado en Lazo Cerrado 30
        Wn=5.8335/(TsMA); //Frecuencia natural del sistema

    //Ubicación de 2 Polos reales
    P1=Wn+Wn;
    P2=Wn*Wn;

    kp=(P1*tao-l)/k; //Calculo de Kc
    ti=(k*kp)/(P2*tao); //Calculo de ti

    // Calculo de control PID digital
    q0=kp*(1+T/(2*ti)+td/T);
    q1=-kp*(1-T/(2*ti)+(2*td)/T);
    q2=(kp*td)/T;
}

```

El error se establece mediante las variables definidas P1 y P2 para configurarse con los valores impuestos por el HMI en su configuración principal antes de su funcionamiento.

```

e=R-yM;
// Control PID
u = u_1 + q0*e + q1*e_1 + q2*e_2; //Ley del controlador PID discreto
if (u >= 5000.0) //Saturo la accion de control 'uT' en un tope maximo y minimo
    u = 5000.0;
if (u <= 0.0)
    u = 0.0;

```

```

control=u/5;
//Retorno a los valores reales
e_2=e_l;
e_l=e;
u_l=u;
//La accion calculada la transformo en PWM
set_pwm4_duty(control);
}

```

La variable control es la que se envía al PWM como apoyo y la que se adapta para mostrar la intensidad del led según el seteo si llega a pasarse o no.

```

void main()
{
lcd_init(0x4E,20,4);
set_tris_a(0x0F);
set_tris_g(0x00);
enable_interrupts(GLOBAL);
enable_interrupts(INT_RDA2);
// Generemos una Señal cuadrada de 1 Khz
Timer2=249; //Se carga timer 2 con 249
Prescaler=1; //Preescaler solo puede tomar valores de: 1

setup_timer_2(t2_div_by_4,Timer2,Prescaler); //Configuracion de Timer 2 para establecer frec. PWM a 1kHz
setup_ccp4(ccp_pwm); //Configurar modulo CCP1 en modo PWM

setup_adc(ADC_clock_internal);
setup_adc_ports(AND);
set_adc_channel(0);
lcd_putc(" UNIVERSIDAD");
lcd_gotoxy(1,2);
lcd_putc(" POLITECNICA");
lcd_gotoxy(1,3);
lcd_putc(" SALESIANA");
lcd_gotoxy(1,4);
lcd_putc(" GUAYAQUIL");
delay_ms(1000);
lcd_putc("\f PLANTA DE LUMINOSIDAD.");

while(true)
{
while(marcha==1)
{
lcd_gotoxy(1,1);

```

```

printf(lcd_putc,"ENCENDIDO ");

cont=read_adc();
yM=(cont*3)/10;    // 0 --> 150 GRADOS

dis();
lcd_gotoxy(1,2);
printf(lcd_putc,"L-SET=%4.0f LUM=%ld ",R,yM);
lcd_gotoxy(1,3);
printf(lcd_putc,"PWM=%4ld TsMA=%2.0f",control,TsMA);
lcd_gotoxy(1,4);
printf(lcd_putc,"k=%1.3f tao=%2.0f",k,tao);
}
lcd_gotoxy(1,2);
printf(lcd_putc,"APAGADO ");
lcd_gotoxy(1,3);
printf(lcd_putc,"");
lcd_gotoxy(1,4);
printf(lcd_putc,"");
set_pwm4_duty(0);
}
}

```

ANEXO 9: Solución propuesta de la práctica #9

Creamos un proyecto en Nextion y le asignamos un nombre que describa la práctica.

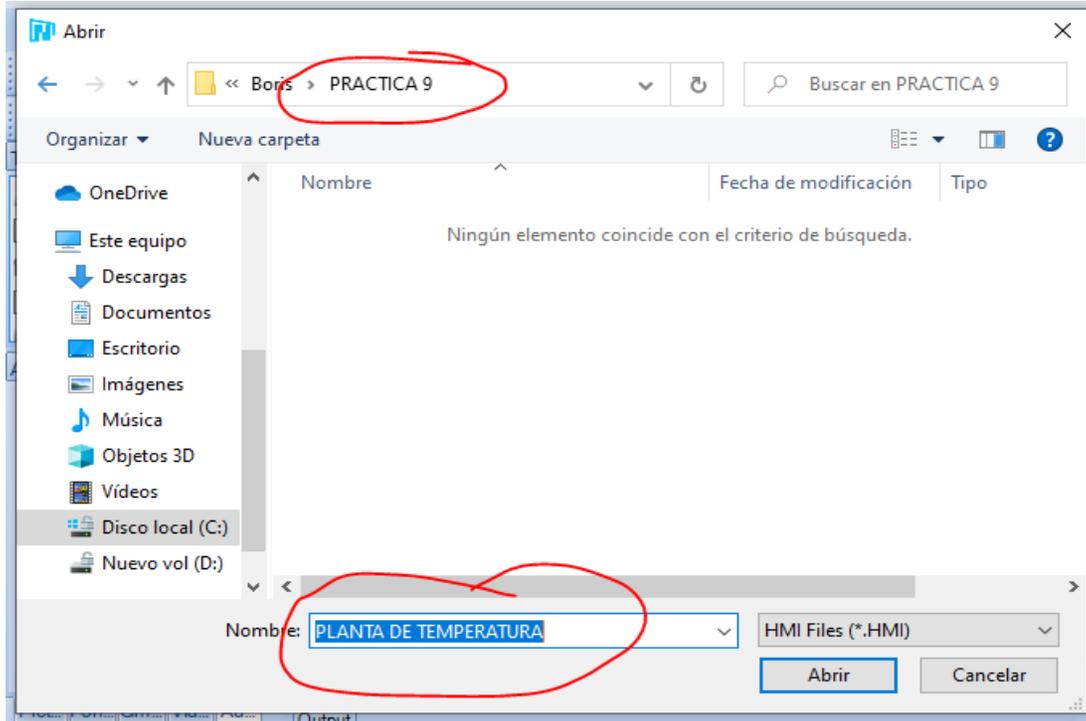


Fig. 115.- Creación de proyecto en Nextion
Fuente: Autores

Una vez creado el proyecto estableceremos cuatro páginas para trabajar en ellas, la página 1 es la presentación de la práctica y de la página 2 a la página 4 la configuración del proceso industrial que hayamos definido en este caso el uso de la planta de control PID de velocidad.

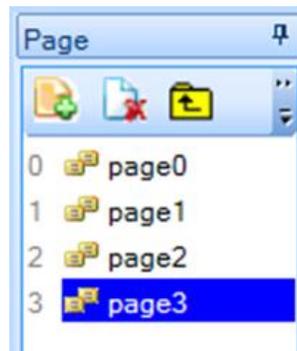


Fig. 116.- Creación de páginas de trabajo
Fuente: Autores

Al crear la carpeta y el archivo de la práctica 9 se crean 4 páginas para su funcionamiento, la página 1 sirve como presentación y las páginas 2, 3 y 4 como estructura para establecer parámetros de labor de la práctica.



Fig. 117.- Estructura de página 2
Fuente: Autores

En la figura anterior se visualiza la estructura de trabajo de esta práctica, en donde ubicamos los siguientes elementos: 2 botones, 1 texto y 1 imagen, los cuales permiten configurar la temperatura o establecerla por default (desde la programación), para el envío de datos al microcontrolador.

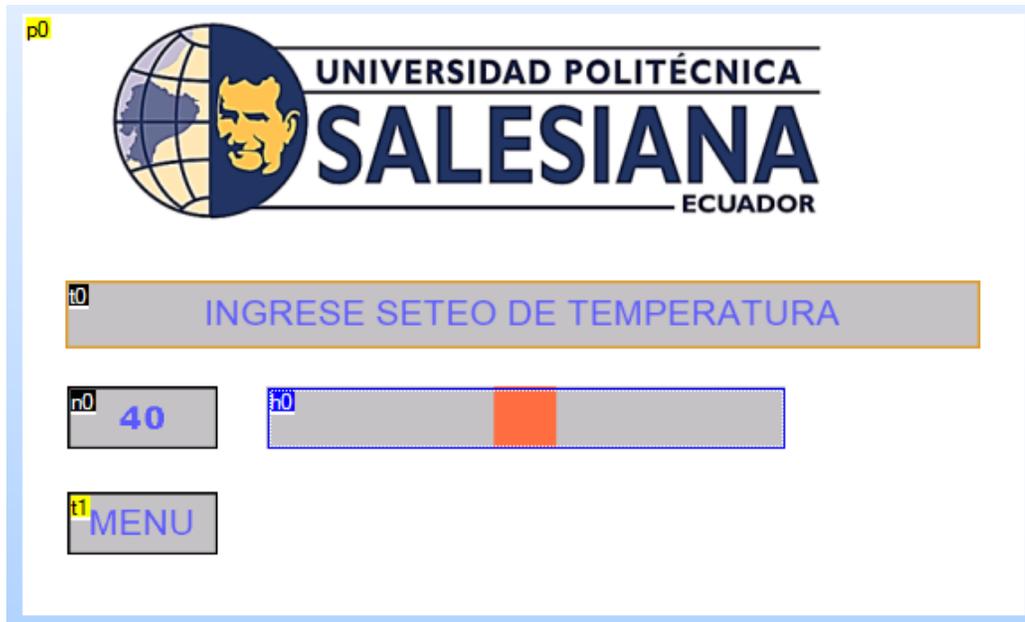


Ilustración 118.- Estructura de página 3
Fuente: Autores

En la figura 118 se puede visualizar los elementos de texto, slider y número para mostrar su diseño y este se realiza una vez que se ha decidido cual de las opciones se tomó en la página 2, si la de configurar los parámetros o dejarlo por default. En la figura siguiente se puede visualizar los elementos: 3 textos, tres botones, 1 scrolling text, 2 progresos, 1 numero y 1 imagen; que permiten visualizar en tiempo real lo que sucede en la planta de control y en cambio de valores y datos que existe entre le microcontrolador y la pantalla HMI.

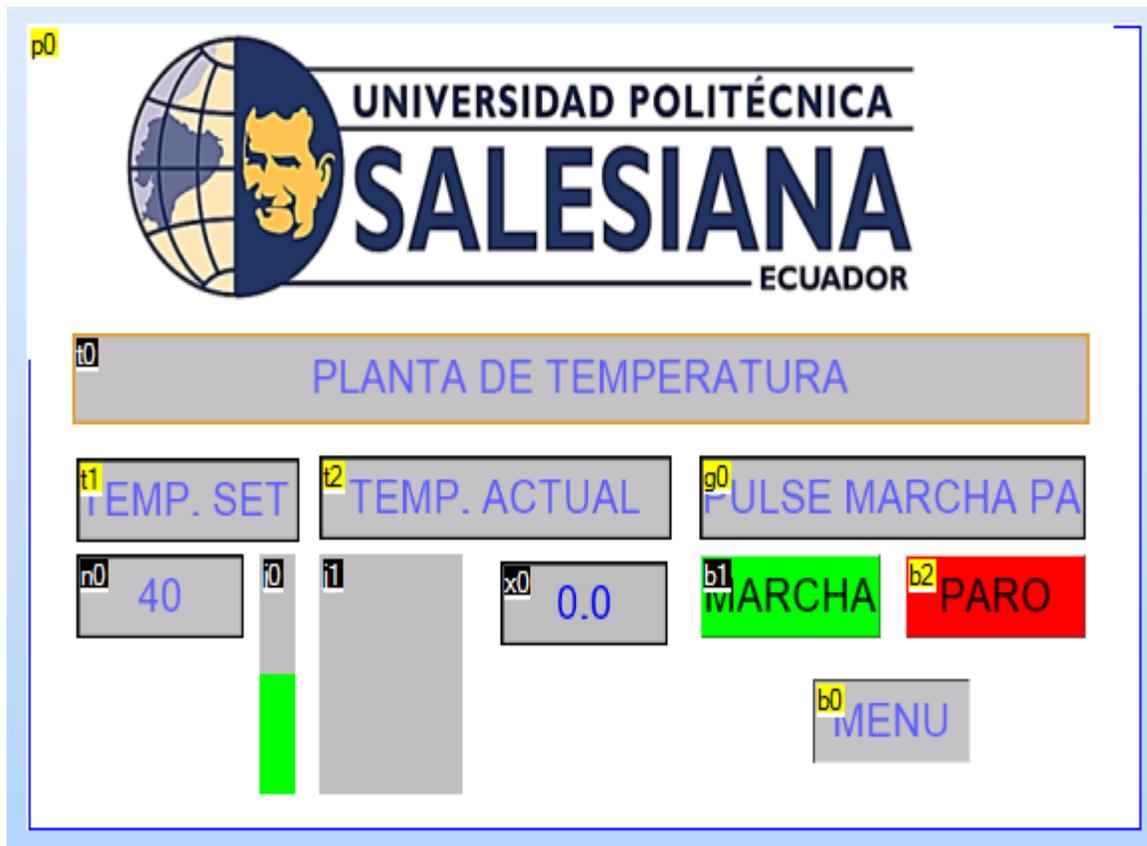


Ilustración 119.- Estructura de página 4
Fuente: Autores

Estos valores se modifican en el elemento de progreso denominado j1 y el número x0 que es un elemento de valor decimal no entero, y se realiza mediante la siguiente codificación en el microcontrolador.

Establecemos líneas de código básica de funcionamiento del microcontrolador:

```
#include <18f8720.h>
#DEVICE ADC=8
```

```
#fuses xt,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOCPD,NOPT,NOVRT
#use delay(clock=4000000)
```

Se ubica el tipo de comunicación, en este caso I2C que trabajan de diferentes pines y la pantalla HMI como maestro.

```
#use i2c(Master,Fast=100000,sda=PIN_H1,scl=PIN_H2,force_sw)
#use rs232(baud=9600,xmit=PIN_G1,rcv=PIN_G2,TIMEOUT=500) //G6 -- G7
```

Declaramos el PWM del microcontrolador en donde se respalda el funcionamiento de la programación de la pantalla HMI para el encendido de los leds RGB.

```
#USE PWM(OUTPUT=PIN_G3,FREQUENCY=10kHz,DUTY=25,stream=vent) //PWM en G3, CCP4
#USE PWM(OUTPUT=PIN_G4,FREQUENCY=10kHz,DUTY=25,stream=cal) //PWM en G4, CCP5
```

Se incluye librería creada de LCD para su visualización en los datos recibidos por la pantalla HMI y las variables a utilizar en el microcontrolador para establecer la comunicación continua con los datos que envía la HMI.

```
#include <i2c_Flex_LCD.c>
#byte porte=0xf84
void hmi(void);
```

```
int s1,s2,s3,porcc,porcv;
int16 tactual;
int16 dato[32];
int tset=100;
int marcha=0;
```

Declaramos la interrupción interna RDA para que cuando se envíe un valor lo muestre enseguida, en donde se establece parámetros una vez que en la pantalla HMI haya algún cambio, dándole así el porcentaje para mostrar dicho cambio.

```
#INT_RDA2
void RDA_isr2()
{
    s1=getc(); //h
    s2=getc(); //0,1,2
    s3=getc(); // valor 0-255

    if(s1==104 && s2==48) //codigo ascii de "h"=104 y de "0" es 48
```

```

    { tset=s3; }
    if(s1==98 && s2==48 && s3==49) //codigo ascii de "b"=98 y de "l" es 49
    { marcha=l; }
    if(s1==98 && s2==48 && s3==48) //codigo ascii de "b"=98 y de "0" es 48
    { marcha=0; }

}

void hmi(void)
{
int16 t1,t2;
float temp;
printf("page3.n1.val=%ld",tactual); printf("ÿÿ");
t1=(tactual*10)/15;
printf("page3.j1.val=%ld",t1); printf("ÿÿ");
}

```

Se establece la conversión analógica digital de un reloj interno para que cuando se active la interrupción sea eso lo que se muestre en el LCD y en la pantalla HMI, buscando llegar a la referencia establecida.

```

void main()
{
int16 x,vato;
set_tris_a(0x0F);
set_tris_b(0xFF);
set_tris_e(0x00);
lcd_init(0x4E,20,4);
porte=0;
setup_adc(ADC_CLOCK_INTERNAL);
setup_adc_ports(AN0|VSS_VREF);
set_adc_channel(0);
enable_interrupts(GLOBAL);
enable_interrupts(INT_RDA2);
lcd_putc(" UNIVERSIDAD");
lcd_gotoxy(1,2);
lcd_putc(" POLITECNICA");
lcd_gotoxy(1,3);
lcd_putc(" SALESIANA");
lcd_gotoxy(1,4);
lcd_putc(" GUAYAQUIL");
delay_ms(1000);
lcd_putc("\f PLANTA DE TEMP.");
}

```

Los valores que envía el PWM como apoyo y que se sirve como adaptación del PID es para generar el proceso de un PID de primer grado y mientras se mantenga en el rango o seteo establecido encenderá y apagará la práctica.

```

while(true)
{
  while(marcha==1)
  {
    set_adc_channel(0);
    for(x=0; x<=10; x++) {
      dato[x]=read_adc();
      delay_ms(100);
    }
    vato=0;
    for(x=0; x<=10; x++) {
      if(dato[x]>vato)
      {
        vato=dato[x];
        delay_ms(100);
      }
    }
    tactual=(10*vato)/17;    // 0 --> 150 GRADOS
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Temperatura= %3ld",tactual);
    if((tactual <= (tset*8)/10))
    {
      pwm_set_duty_percent(vent,5);
      pwm_set_duty_percent(cal,980);
    }
    if((tactual > (tset*8)/10) && (tactual < tset))
    {
      pwm_set_duty_percent(vent,5);
      porcc= (tset-tactual)*50;
      pwm_set_duty_percent(cal,porcc);
    }
    if((tactual > tset) && (tactual <= (tset*1.1) ))
    {
      porcv= (tactual-tset)*100;
      pwm_set_duty_percent(vent,porcv);
      pwm_set_duty_percent(cal,5);
    }
    if((tactual > (tset*1.1)))
    {
      pwm_set_duty_percent(vent,980);
    }
  }
}

```

```

    pwm_set_duty_percent(cal,5);
}
lcd_gotoxy(1,3);
printf(lcd_putc,"dato=%3ld vato=%3ld "data[4],vato);
lcd_gotoxy(1,4);
lcd_putc("SISTEMA ENCENDIDO");
hmi();
}

set_adc_channel(0);
for(x=0; x<=31; x++) {
    dato[x]=read_adc();
    delay_ms(100);
}
vato=0;
for(x=0; x<=31; x++) {
    if(dato[x]>vato)
    {
        vato=dato[x];
        delay_ms(100);
    }
}
tactual=(10*vato)/17;    // 0 --> 150 GRADOS
lcd_gotoxy(1,2);
printf(lcd_putc,"Temperatura= %3ld",tactual);
pwm_set_duty_percent(vent,0);
pwm_set_duty_percent(cal,1000);
lcd_gotoxy(1,3);
lcd_putc("      ");
lcd_gotoxy(1,4);
lcd_putc("SISTEMA APAGADO ");
}
}

```

ANEXO 10: Solución propuesta de la práctica #10

Creamos un proyecto en Nextion y le asignamos un nombre que describa la práctica.

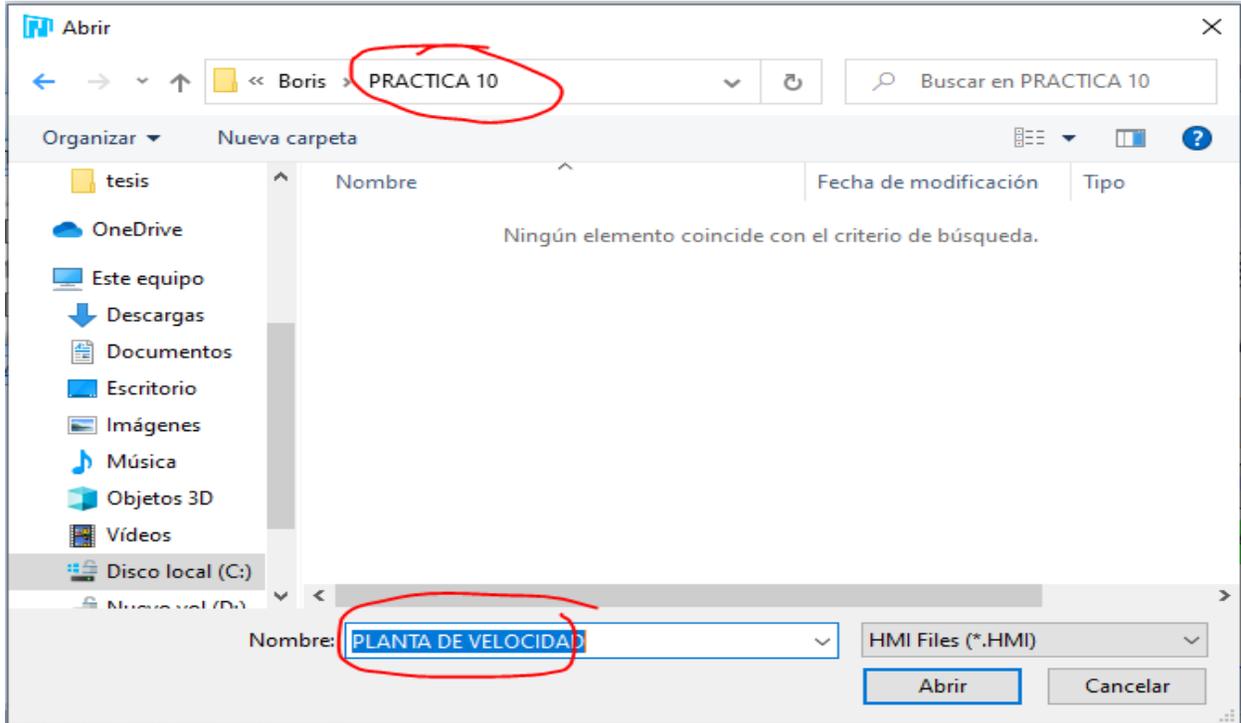


Fig. 120.- Creación de proyecto en Nextion
Fuente: Autores

Una vez creado el proyecto estableceremos cuatro páginas para trabajar en ellas, la página 1 es la presentación de la práctica y las páginas 2, 3 y 4 son para la configuración del proceso industrial que hayamos definido en este caso el uso de la planta de control PID de velocidad.

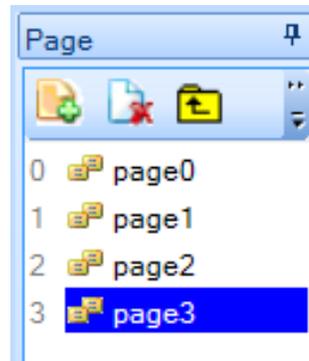


Fig. 121.- Creación de páginas de trabajo
Fuente: Autores

Al crear la carpeta y el archivo de la práctica 10 se crean 4 páginas para su funcionamiento, la página 1 sirve como presentación y de la página 2 a la página 4 como estructura para establecer parámetros de labor de la práctica. En la figura 122 se visualiza la estructura de trabajo de esta práctica, en donde ubicamos los siguientes elementos: 2 botones, 1 texto y 1 imagen, los cuales permiten configurar la temperatura o establecerla por default (desde la programación), para el envío de datos al microcontrolador.



Fig. 122.- Estructura de página 2
Fuente: Autores

Los elementos para diseñar la página 3 se utilizan son: 6 textos, 4 números, 4 sliders y 1 imagen; estos elementos permiten la configuración de la planta de control de velocidad, desde su tiempo de estabilidad hasta su ganancia. Una vez establecidos los valores de funcionamiento en la página 3, presionamos menú para regresar a la página 2 y así iniciar el proceso.



Fig. 123.- Estructura de página 3
Fuente: Autores

En la figura 124 se visualiza el funcionamiento de la práctica en tiempo real es por ello por lo que su diseño tiene 2 elementos de texto, 3 botones, 1 scrolling text, 1 progreso y 1 imagen; en donde se muestra todo lo configurado y los cambios que existen en la planta de control siendo así un PID discreto.

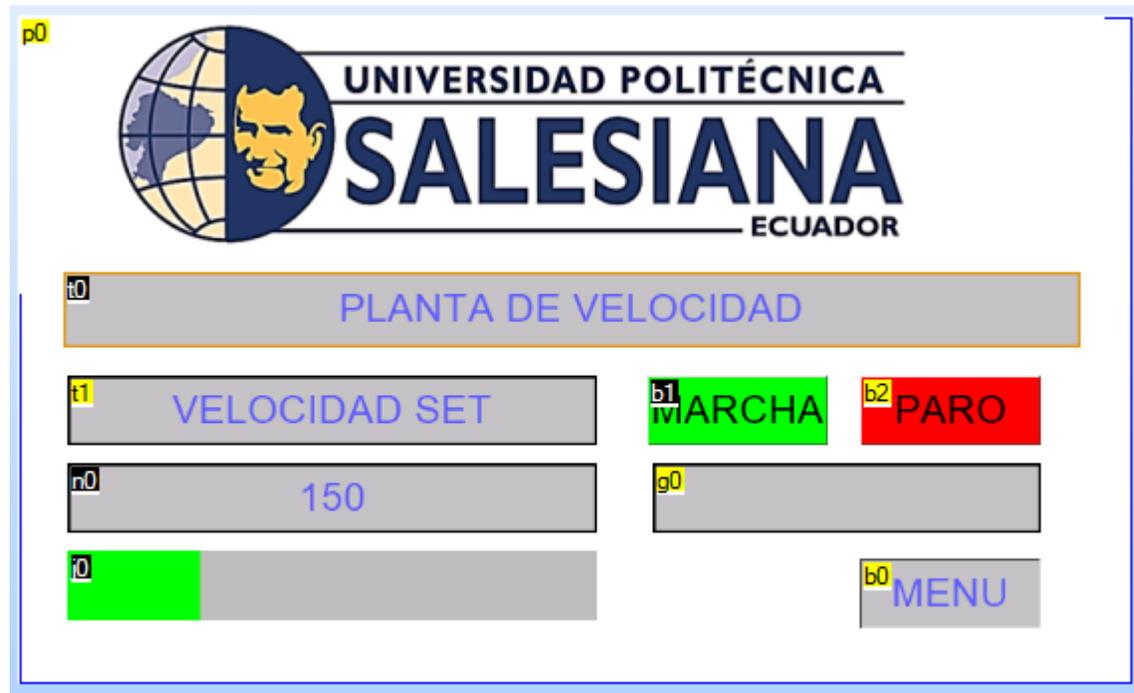


Fig. 124.- Estructura de página 4
Fuente: Autores

Estos valores se modifican en el elemento de progreso denominado j0 y se realiza mediante la siguiente codificación en el microcontrolador.

Establecemos líneas de código básica de funcionamiento del microcontrolador:

```
#include <18f8720.h>
#fuses xt
#use delay(clock=4000000)
```

Se ubica el tipo de comunicación, en este caso I2C que trabajan de diferentes pines y la pantalla HMI como maestro.

```
#use i2c(Master,Fast=100000, sda=PIN_HI, scl=PIN_H2,force_sw)
#use rs232(baud=9600, xmit=PIN_G1,rcv=PIN_G2,TIMEOUT=500) //C6 -- C7
```

Se incluye librería creada de LCD para su visualización en los datos recibidos por la pantalla HMI y las variables a utilizar en el microcontrolador para establecer la comunicación continua con los datos que envía la HMI.

```
#include <i2c_Flex_LCD.c>
#byte porte=0xf84
int s1,s2,s3,s4,marcha;
float R=150.0,e=0.0,e_1=0.0,e_2=0.0,u=0.0,u_1=0.0;
float kp,ti,td,q0,q1,q2,T=0.1;
float k=0.535,tao=10.0,theta=1.0,kw;
float Wn,P1,P2;
float TsMA=30;

int16 control,control2,yM;
int16 cont;
int Timer2,Poscaler;
INT BA=0;
void dis(void);
```

Declaramos la interrupción interna RDA para que cuando se envíe un valor lo muestre enseguida, en donde se establece parámetros una vez que en la pantalla HMI haya algún cambio, dándole así el porcentaje para mostrar dicho cambio.

```
#INT_RDA2
void RDA_isr2()
{
    s4=0;
    kw=0;
    s1=getc(); //h
    s2=getc(); //0,1,2
    s3=getc(); // valor 0-255
    s4=getc();

    if(s1==98 && s2==48 && s3==49) //codigo ascii de "b"=98 y de "1" es 49
        { marcha=1; }
    if(s1==98 && s2==48 && s3==48) //codigo ascii de "b"=98 y de "0" es 48
        { marcha=0; }
    if(s1==104 && s2==48) //codigo ascii de "h"=104 y de "0" es 48
        { R=s3;
          kw=s4;
          if(kw!=0)
              {R=R+(256*kw); }
        }
```

```

    }
    if(s1==104 && s2==50) //codigo ascii de "h"=110 y de "3" es 51
    { tao=s3; }
    if(s1==104 && s2==49) //codigo ascii de "h"=110 y de "4" es 52
    { k=s3;
      kw=s4;
      if(kw!=0)
        {k=k+(256*kw); }
      k=k/1000;
    }
    if(s1==104 && s2==51) //codigo ascii de "h"=104 y de "0" es 48
    { TsMA=s3; }

}

#int_ext //Activa la interrupcion, solo en B0
void ext_isr()
{
  cont++;
}

```

La configuración del PID discreto se hace mediante el diseño de asignación de dos polos reales y teniendo en cuenta la variable que busca llegar al seteo oficial, que en este caso es la velocidad del motor donde se establece una mayor potencia del giro de su eje.

```

void dis(void)
{
  //***** DISEÑO POR ASIGNACIÓN DE 2 POLOS REALES *****//

  //Tiempo deseado en Lazo Cerrado 30
  Wn=5.8335/(TsMA); //Frecuencia natural del sistema

  //Ubicación de 2 Polos reales
  P1=Wn+Wn;
  P2=Wn*Wn;

  kp=(P1*tao-l)/k; //Calculo de Kc
  ti=(k*kp)/(P2*tao); //Calculo de ti

  // Calculo de control PID digital
  q0=kp*(1+T/(2*ti)+td/T);
  q1=-kp*(1-T/(2*ti)+(2*td)/T);
  q2=(kp*td)/T;
}

```

El error se establece mediante las variables definidas P1 y P2 para configurarse con los valores impuestos por el HMI en su configuración principal antes de su funcionamiento.

```
e=R-yM;
// Control PID
u = u_1 + q0*e + q1*e_1 + q2*e_2; //Ley del controlador PID discreto
if (u >= 5000.0) //Saturo la accion de control 'uT' en un tope maximo y minimo
u = 5000.0;
if (u <= 0.0)
u = 0.0;
control=u/5;
//Retorno a los valores reales
e_2=e_1;
e_1=e;
u_1=u;
//La accion calculada la transformo en PWM
if(control<250)
{
if (BA==0)
{
set_pwm5_duty(900);
delay_ms(100);
BA=1;
}
}
set_pwm5_duty(control);
}
```

La variable control es la que se envía al PWM como apoyo y la que se adapta para mostrar la intensidad de velocidad del motor según el seteo si llega a pasarse o no en sus revoluciones.

```
void main()
{
lcd_init(0x4E,20,4);
set_tris_g(0x04);
enable_interrupts(GLOBAL);
enable_interrupts(int_ext); //habilitar la interrupcion externa
ext_int_edge(L_to_H);
enable_interrupts(INT_RDA2);
// Generemos una Señal cuadrada de 1 Khz
Timer2=249; //Se carga timer 2 con 249
Poscaler=1; //Preescaler solo puede tomar valores de: 1
```

```

setup_timer_2(t2_div_by_4,Timer2,Poscaler); //Configuracion de Timer 2 para establecer frec. PWM a 1kHz
setup_ccp5(ccp_pwm); //Configurar modulo CCP1 en modo PWM
lcd_putc(" UNIVERSIDAD");
lcd_gotoxy(1,2);
lcd_putc(" POLITECNICA");
lcd_gotoxy(1,3);
lcd_putc(" SALESIANA");
lcd_gotoxy(1,4);
lcd_putc(" GUAYAQUIL");
delay_ms(1000);
lcd_putc("\fPLANTA DE VELOCIDAD.");
delay_ms(1000);

while(true)
{
while(marcha==1)
{
lcd_gotoxy(1,1);
printf(lcd_putc,"ENCENDIDO ");
enable_interrupts(int_ext);
cont=0;
delay_ms(333);
yM=cont*3;
disable_interrupts(int_ext);
cont=0;
dis();
lcd_gotoxy(1,2);
printf(lcd_putc,"SET=%4.0f Vel=%ld ",R,yM);
lcd_gotoxy(1,3);
printf(lcd_putc,"PWM=%4ld TsMA=%2.0f",control,TsMA);
lcd_gotoxy(1,4);
printf(lcd_putc,"k=%1.3f tao=%2.0f",k,tao);

}
BA=0;
lcd_gotoxy(1,2);
printf(lcd_putc,"APAGADO ");
set_pwm5_duty(0);
}
}

```