



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA



**DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE (PLC)
MODULAR BASADO EN UN MICROCONTROLADOR PIC PARA
APLICACIONES INDUSTRIALES Y DIDÁCTICAS CON CAPACIDAD DE
COMUNICACIÓN 1-WIRE®**

HERNÁNDEZ O. JOSÉ F.
LUGO P. EFRAÍN G.

VALENCIA, JUNIO DE 2013



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA



**DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE (PLC)
MODULAR BASADO EN UN MICROCONTROLADOR PIC PARA
APLICACIONES INDUSTRIALES Y DIDÁCTICAS CON CAPACIDAD DE
COMUNICACIÓN 1-WIRE®**

Trabajo especial de grado presentado ante la ilustre Universidad de Carabobo para
optar al título de Ingeniero Electricista

HERNÁNDEZ O. JOSÉ F.
LUGO P. EFRAÍN G.

VALENCIA, JUNIO DE 2013



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA



CERTIFICADO DE APROBACIÓN

Los abajo firmantes miembros del jurado asignado para evaluar el trabajo especial de grado titulado **“Desarrollo de un Controlador Lógico Programable (PLC) Modular Basado en un Microcontrolador PIC para Aplicaciones Industriales y Didácticas con Capacidad de Comunicación 1-Wire®”**, realizado por los bachilleres: **Hernández O. José F. C.I. 18.020.680** y **Lugo P. Efraín G. C.I. 15.662.808**, hacemos constar que hemos revisado y aprobado dicho trabajo.

Prof. Ángel Villegas

TUTOR

Prof. Andrés Simone

JURADO

Prof. Aída Pérez

JURADO

VALENCIA, JUNIO DE 2013



AGRADECIMIENTOS

A la Universidad de Carabobo por la inestimable formación académica que nos brindó, por ser nuestra casa de estudios y por brindarnos las herramientas para nuestro desarrollo personal y profesional.

Al Profesor Ángel Villegas, por aceptar ser nuestro tutor, y por su orientación y dedicación en la elaboración de este proyecto de grado.

Agradecemos también a nuestros compañeros y amigos, que hicieron de nuestra pasada por la universidad un período muy agradable.

Y a todas aquellas personas que siempre creyeron en nosotros y que de una u otra forma, aún en los momentos difíciles, nos ayudaron a culminar nuestro proyecto de grado, gracias, muchas gracias.

José F. Hernández O.

Efraín G. Lugo P.



DEDICATORIA

Dedico este trabajo de grado a mi madre que ha sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante y por estar en todo momento conmigo.

A mi padre, a pesar de nuestra distancia física, siento que estás conmigo siempre y aunque nos faltaron muchas cosas por vivir juntos, sé que este momento es tan especial para ti como para mí.

A mi esposa, por amarme, ayudarme, guiarme y apoyarme en toda mi carrera, y por estar siempre a mi lado incondicionalmente.

A toda mi familia por ser tan imprescindibles, por todo el apoyo y los consejos que siempre me han brindado, por todo esto y más les dedico mi trabajo, los amo a todos.

Efraín G. Lugo P.



DEDICATORIA

Este proyecto de grado va dedicado con todo mi amor, cariño y esfuerzo a toda mi familia y amigos...

A mi espectacular madre por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de vida, por estar siempre a mi lado, amándome, ayudándome, guiándome y apoyándome en toda mi carrera y muy especialmente por creer siempre en mí.

A mis tíos por ser mis padres ejemplares, siempre atentos a corregirme, ayudarme en todo momento y estar siempre cuando los he necesitado.

A mis amados hermanos, primos y amigos, que al verlos graduarse e ir alcanzando sus propias metas, me motivaron a desear alcanzar las mías y cumplirlas.

A mi novia por ayudarme, apoyarme, motivarme y ser ejemplo de que lo que se quiere se puede.

A toda mi familia por estar siempre conmigo en todo momento apoyándome y fomentado en mí el deseo de superación y el anhelo de triunfo en la vida.

José F. Hernández O.



ÍNDICE GENERAL

ÍNDICE GENERAL.....	VII
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS	XVI
RESUMEN	XVII
ABSTRACT	XVIII
INTRODUCCIÓN	1
CAPÍTULO I.....	3
EL PROBLEMA	3
1.1 PLANTEAMIENTO DEL PROBLEMA.....	3
1.2 JUSTIFICACIÓN DE LA INVESTIGACIÓN	5
1.3 OBJETIVOS.....	7
1.3.1 OBJETIVO GENERAL.....	7
1.3.2 OBJETIVOS ESPECÍFICOS.....	7
1.4 ALCANCE Y LIMITACIONES	8
CAPÍTULO II	10
MARCO TEÓRICO.....	10
2.1 ANTECEDENTES.....	10
2.2 CONTROLADOR LÓGICO PROGRAMABLE (PLC)	12
2.2.1 Introducción	12
2.2.2 Definición.....	13
2.2.3 Clasificación.....	14



2.2.4	Estructura.....	15
2.2.4.1	Estructura Externa.....	15
2.2.4.2	Estructura Interna.....	16
2.2.5	Buses.....	16
2.2.6	Unidad Central de Proceso.....	17
2.2.7	Sistema de Memorias.....	18
2.2.8	Tipos de Memoria y Aplicaciones en los PLC.....	19
2.2.9	Memoria del PLC según los datos almacenados.....	22
2.2.9.1	Memoria de Programa de Usuario.....	22
2.2.9.2	Memoria Interna.....	24
2.2.10	Interfaces de Entrada/Salida.....	25
2.2.11	Clasificación de los Módulos de Entrada/Salida.....	26
2.2.12	Fuente de Alimentación.....	27
2.2.13	Funcionamiento General de un PLC.....	28
2.2.14	Tipos de Lenguajes de Programación para PLC.....	29
2.2.14.1	Diagrama de Contactos (<i>Ladder - LD</i>).....	30
2.2.14.2	Plano de Funciones Lógicas (<i>Function Block Diagram - FBD</i>)....	30
2.2.14.3	Lista de Instrucciones (<i>Instruction List - IL</i>).....	31
2.2.14.4	Lenguaje Estructurado (<i>Structured Text - ST</i>).....	32
2.2.14.5	Gráfico de Funciones Secuenciales (<i>Sequential Function Chart</i>)	32
2.2.14.6	Aplicación en la Industria.....	33
2.2.15	Ventajas e Inconvenientes de los PLC.....	35



2.2.15.1	Ventajas	35
2.2.15.2	Inconvenientes	36
2.3	MICROCONTROLADOR.....	37
2.3.1	Definición.....	37
2.3.2	Tipos de Encapsulados.....	38
2.3.3	Microcontrolador PIC16F887	39
2.3.3.1	Características.....	39
2.3.3.2	Estructura Externa.....	40
2.4	Ambiente de Desarrollo <i>Proton IDE</i>	40
2.5	Tecnología 1-Wire®.....	42
2.5.1	Bus 1-Wire®	43
2.5.2	Elementos que componen una Red 1-Wire®.....	44
2.5.3	Protocolo de Comunicaciones 1-Wire®	45
2.6	LDmicro	48
2.6.1	Instrucciones	49
2.6.2	La Simulación	57
2.6.3	Compilando a Código Nativo.....	58
CAPÍTULO III.....		62
MARCO METODOLÓGICO.....		62
3.1	TIPO DE INVESTIGACIÓN.....	62
3.2	PROCEDIMIENTO METODOLÓGICO	62
CAPÍTULO IV.....		66
RESULTADOS.....		66



4.1	Definición de la estructura y funcionalidad del prototipo a desarrollar atendiendo a las características existentes en diferentes PLC de Gama Baja	66
4.2	Diseño de los diagramas esquemáticos y de circuito impreso requeridos por el PLC para las interfaces de Entrada/Salida Digitales y Analógicas, Protocolo de Comunicación y Fuente de Alimentación	69
4.2.1	Módulos Industriales de Entrada y Salida	70
4.2.1.1	Entrada/ Salida Digital	71
4.2.1.2	Entrada Analógica	72
4.2.2	Salida Analógica	74
4.2.3	Fuente de Alimentación	75
4.2.4	Interfaz de Hardware para Comunicación 1-Wire®	77
4.3	Construcción de los diseños preliminares	87
4.4	Construcción de circuitos prototipos	95
4.5	Adaptación del compilador de lenguaje escalera distribuido bajo licencia GNU LDmicro	104
4.6	MonitorPLC887	106
4.7	Elaboración de las prácticas de laboratorio	114
4.8	Cargar archivos hexadecimales (.hex) al PLC887	115
CAPÍTULO V		116
CONCLUSIONES		116
RECOMENDACIONES		119
BIBLIOGRAFÍA		120
ANEXOS		123

A Estudio de Características de PLC de Gama Baja.



B Manuales de Usuario.

C Prácticas y Soluciones.

- **Práctica 1:** Introducción al PLC887 y a su programación en lenguaje escalera utilizando el LDmicro.
- **Práctica 2:** Arranque, parada y cambio de giro de un motor trifásico.
- **Práctica 3:** Introducción al Proton IDE como herramienta alternativa de programación del PLC887.

D Hojas de Especificaciones Técnicas.

- Microcontrolador PIC 16F887 – Microchip.
- Módulo de Entrada Digital MIAC/MIDC – Dataforth.
- Módulo Relé de Salida Digital MORO/MORC – Dataforth.
- Módulo de Entrada en Corriente 8B32 – Dataforth.
- Transmisor de Lazo en Corriente de 4-20mA XTR115 – Texas Instruments.
- Optoacoplador Fototransistor 4N35 – Fairchild Semiconductor.
- Transistor Bipolar de Potencia Media 2N4922 – On Semiconductor.
- Conversor DC/DC no Regulado DCP022405U – Texas Instruments.
- Acoplador MicroLAN DS2409 – Dallas Semiconductor.
- Microcontrolador PIC 18F2550 – Microchip.
- Referencia de Tensión LM285 – Texas Instruments.
- Resonador Cerámico 20MHz – Abracon Corporation.



ÍNDICE DE FIGURAS

Figura 2.1	Diagrama Conceptual del uso del PLC.	13
Figura 2.2	Diagrama de bloques simplificado en el que se detalla la estructura de los buses de un PLC estándar.	17
Figura 2.3	Representación gráfica de una exploración.....	29
Figura 2.4	Ejemplo de programa con diagrama de contactos.....	30
Figura 2.5	Ejemplo de Símbolos con Lenguaje de Plano de Funciones Lógicas (FBD).	31
Figura 2.6	Ejemplo de Programa con Lenguaje de Instrucciones (IL).	31
Figura 2.7	Ejemplo de programa con Lenguaje Estructurado (ST).....	32
Figura 2.8	Ejemplo de programa realizado en SFC.....	33
Figura 2.9	Tipos de Encapsulados de Microcontroladores PIC.....	38
Figura 2.10	Diagrama de Terminales del Microcontrolador PIC16F887.	40
Figura 2.11	Presentación de la ayuda sobre el Compilador Proton IDE.	41
Figura 2.12	La configuración 1-Wire maestro/esclavo utiliza una única línea de datos más su referencia a tierra.....	42
Figura 2.13	Representación gráfica de un pulso de Reset y Presencia.....	46
Figura 2.14	Transmisión y recepción de bits empleando el protocolo 1-Wire®.....	47
Figura 2.15	Configuración de la Instrucción Conversor A/D del LDmicro.	55
Figura 2.16	Menú para Simulaciones con el LDmicro.	57
Figura 2.17	Modo de Simulación del LDmicro.....	58
Figura 2.18	Configuración de los Terminales Entrada/Salida del LDmicro.	59
Figura 2.19	Menú de Compilación del LDmicro.....	60



Figura 2.20	Ventana Resultado de la Compilación del LDmicro.	61
Figura 4.1	Diagramas de Conexiones entre el PLC887 y Módulos de Expansión. 68	
Figura 4.2	Diagrama de Terminales del Microcontrolador PIC16F887 asociados al PLC887.	70
Figura 4.3	Esquema Electrónico del Módulo de Entrada Digital – DataForth®. ...	71
Figura 4.4	Esquema Electrónico del Módulo de Salida Digital – DataForth®.	72
Figura 4.5	Esquema Electrónico del Módulo de Entrada Analógica – DataForth®...	73
Figura 4.6	Esquema Electrónico del XTR115UA – Texas Instruments®.	74
Figura 4.7	Esquema Electrónico del 4N35SR2M – Fairchild Semiconductor®. ...	75
Figura 4.8	Esquema Electrónico del DCP022405U – Texas Instruments®.	76
Figura 4.9	Esquema Electrónico de la Conexión en Paralelo del DCP022405U – Texas Instruments®.	76
Figura 4.10	Esquema Electrónico del DS2409 – Dallas Semiconductor®.	77
Figura 4.11	Diagrama de Terminales del Microcontrolador PIC18F2550.	77
Figura 4.12	Esquemático Conexiones al PIC16F887.	79
Figura 4.13	Esquemático de Conexiones a los Módulos de Entrada Analógica.	80
Figura 4.14	Esquemático de Conexiones a los Módulos de E/S Digital.	81
Figura 4.15	Esquemático Salidas Analógicas.	82
Figura 4.16	Esquemático Comunicación 1-Wire®.	83
Figura 4.17	Esquemático Módulo de Expansión Digital.	84
Figura 4.18	Esquemático Módulo de Expansión Analógica.	85
Figura 4.19	Esquemático del Módulo de Teclado.	86
Figura 4.20	Montaje de Pruebas del PLC.	87



Figura 4.21 Montaje de Pruebas PLC – Interfaz de Salida de Corriente.....	88
Figura 4.22 Diagrama de los algoritmos usados para las pruebas de las entradas y salidas Digitales del PLC887.....	89
Figura 4.23 Diagrama de los algoritmos usados para las pruebas de las entradas y salidas Analógicas del PLC887.	89
Figura 4.24 Montaje de Pruebas PLC – Interfaz de Salida de Corriente.....	90
Figura 4.25 Gráfica de Valores Ideales para el Manejo de la PWM.	92
Figura 4.26 Gráfica de Valores Reales para el Manejo de la PWM.	93
Figura 4.27 Circuito Impreso del Módulo Principal del PLC (Capa Superior).....	95
Figura 4.28 Circuito Impreso del Módulo Principal del PLC (Capa Inferior).	96
Figura 4.29 Acabado final de la Tarjeta de Circuito Impreso del Módulo Principal del PLC (Capa Superior).	96
Figura 4.30 Acabado final de la Tarjeta de Circuito Impreso del Módulo Principal del PLC (Capa Inferior).....	97
Figura 4.31 Circuito Impreso del Módulo Principal del PLC Totalmente Ensamblado (Vista Superior).....	97
Figura 4.32 Circuito Impreso del Módulo Principal del PLC Totalmente Ensamblado (Vista Inferior).	98
Figura 4.33 Puertos de Conexión del PLC887.	98
Figura 4.34 Circuito Impreso de los Módulos de Expansión.	99
Figura 4.35 Circuito Impreso del Módulo de Teclado.....	99
Figura 4.36 Circuito Impreso del Módulo de Expansión Digital Totalmente Ensamblado (Vista Superior e inferior).....	100
Figura 4.37 Circuito Impreso del Módulo de Expansión Analógica Totalmente Ensamblado (Vista Superior e inferior).....	100



Figura 4.38 Circuito Impreso del Módulo de Teclado Totalmente Ensamblado (Vista Superior e inferior).	100
Figura 4.39 Programa de Prueba del LDmicro después de los cambios.	105
Figura 4.40 Ventana Principal del MonitorPLC887.	106
Figura 4.41 Diagrama de Bloques de la Topología de la Comunicación del PLC887 y el Computador.	107
Figura 4.42 Simulador del PLC887 realizado con el software de simulación PROTEUS Design Suite.	108
Figura 4.43 Descripción de la Ventana Principal del MonitorPLC887.	109
Figura 4.44 Descripción de la Ventana de Gráficas del MonitorPLC887.	110
Figura 4.45 Prueba de monitoreo de entradas y salidas digitales con el MonitorPLC887.	112
Figura 4.46 Prueba de monitoreo de entrada analógica con el MonitorPLC887. ...	113



ÍNDICE DE TABLAS

Tabla 2.1	Tabla de clasificación de las memorias en función del tipo que se utilice en un PLC y de la clase de datos almacenados.....	22
Tabla 2.2	Características del Microcontrolador PIC16F887.	39
Tabla 2.3	Funciones de ROM del bus 1-Wire®.....	46
Tabla 4.1	Especificaciones de los Módulos de Entrada Digital – DataForth®.....	71
Tabla 4.2	Especificaciones de los Módulos de Salida Digital – DataForth®.	72
Tabla 4.3	Especificaciones de los Módulos de Entrada Analógica – DataForth®.	73
Tabla 4.4	Especificaciones del XTR115UA – Texas Instruments®.	74
Tabla 4.5	Especificaciones del 4N35SR2M – Fairchild Semiconductor®.	75
Tabla 4.6	Especificaciones del 4N35SR2M – Texas Instruments®.	76
Tabla 4.7	Especificaciones del Microcontrolador Microchip® PIC18F2550.	78
Tabla 4.8	Valores Ideales para el Manejo de la PWM.	91
Tabla 4.9	Valores Reales para el Manejo de la PWM.....	92
Tabla 4.10	Cálculo de errores de la corriente generada por la PWM.....	94
Tabla 4.11	Elementos que componen el Modulo Principal PLC887.	101
Tabla 4.12	Elementos que componen el Modulo de Expansión Digital.	103
Tabla 4.13	Elementos que componen el Modulo de Expansión Analógica.	103
Tabla 4.14	Elementos que componen el Módulo de Teclado.	104



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA



**DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE (PLC)
MODULAR BASADO EN UN MICROCONTROLADOR PIC PARA
APLICACIONES INDUSTRIALES Y DIDÁCTICAS CON CAPACIDAD DE
COMUNICACIÓN 1-WIRE®**

Autores: Efraín G. Lugo P. | José F. Hernández O.

Tutor: Ángel Villegas

Fecha: Diciembre de 2012

RESUMEN

La presente investigación permitió el desarrollo de un prototipo de Controlador Lógico Programable (PLC) modular, el cual posee un microcontrolador PIC como unidad de control y capacidad de comunicación 1-Wire®. Este dispositivo es capaz de: medir y generar señales analógicas y digitales, comunicarse con el computador empleando el protocolo 1-Wire® para monitorear el estado de las variables físicas conectadas a él, programarse en lenguaje escalera (*Ladder*) y en cualquier lenguaje de programación para microcontroladores PIC y expandir las entradas y salidas analógicas y digitales a través de puertos especiales dedicados a tal fin.

Para la construcción del prototipo de PLC se realizó el diseño esquemático y físico (tarjetas de circuito impreso o PCB) de los circuitos necesarios, para finalmente realizar las pruebas operativas y verificar su correcto funcionamiento.

Con la realización de este proyecto se obtuvieron como producto final los siguientes dispositivos: Módulo principal de control llamado PLC887, el cual cuenta con 8 entradas/salidas digitales, 2 entradas analógicas, 2 salidas analógicas en corriente de 4 a 20 mA y comunicación 1-Wire®. Módulo de expansión digital de 6 entradas/salidas. Módulo de expansión analógica de 4 entradas. Módulo de teclado de 5 teclas y 1 led. Los dispositivos construidos son de bajo costo, alta funcionalidad y fueron diseñados con prestaciones de hardware especiales para su aplicación en laboratorios didácticos y en ambientes industriales. En los ensayos de generación de corriente (usando la PWM del microcontrolador), el PLC887 presentó un comportamiento lineal y una exactitud del 0,3%, demostrando ser un instrumento exacto y confiable en el rango de (4 - 20) mA.

PALABRAS CLAVE: PLC, microcontroladores, Tecnología 1-Wire®.



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA



**DEVELOPMENT OF A MODULAR PROGRAMMABLE LOGIC
CONTROLLER (PLC) BASED ON A PIC MICROCONTROLLER FOR
INDUSTRIAL AND DIDACTIC APPLICATIONS WITH 1-WIRE®
COMMUNICATION CAPABILITY**

Autores: Efraín G. Lugo P. | José F. Hernández O.

Tutor: Ángel Villegas

Fecha: Diciembre de 2012

ABSTRACT

This research allowed the development of a prototype of a modular Programmable Logic Controller (PLC), which has a PIC microcontroller as a control unit and 1-Wire® communication capability. This device is able to: measure and generate analog and digital signals, communicate with the computer using the 1-Wire® protocol for monitoring the state of the physical variables connected to it, programmed in ladder language and any PIC microcontroller programming language and expand the analog inputs and digital inputs and outputs through special ports dedicated to this purpose.

For the develop of this prototype was performed an schematic and physical design (printed circuit board or PCB) for the necessary circuits to finally make the operational testing and checking proper operation.

With the completion of this project were obtained as final product the following devices: Control Main Module named PLC887, which has 8 digital inputs / outputs, 2 analog inputs, 2 analog outputs in current 4 to 20 mA and 1-Wire® communication. Digital Expansion Module has 6 inputs / outputs. Expansion module has 4 analog inputs. Keyboard module for 5 keys and 1 led. The devices are constructed of low cost, high functionality and were designed with special hardware features for use in teaching laboratories and industrial environments. In the current generation assays (using the microcontroller's PWM), the PLC887 showed a linear behavior and an accuracy of 0.3%, proving to be an accurate and reliable instrument in the range of (4-20) mA.

KEYWORDS: PLC, microcontroller, 1-Wire® Technology.



INTRODUCCIÓN

En este proyecto se desarrolló un prototipo de Controlador Lógico Programable (PLC) modular, el mismo tiene como unidad de control un microcontrolador PIC y capacidad de comunicación 1-Wire®. Su uso comprende aplicaciones de ámbito industrial y didáctico. Éste es capaz de medir variables tanto analógicas como digitales, generar salidas analógicas y digitales, comunicarse con un computador PC a través del protocolo 1-Wire® para monitorear el estado de las variables físicas conectadas a él, programarse en lenguaje escalera (*Ladder*) y en cualquier lenguaje de programación para microcontroladores PIC y expandir las entradas y salidas analógicas y digitales a través de puertos especiales dedicados a tal fin.

Dicho trabajo contempló el diseño esquemático y físico (tarjetas de circuito impreso o PCB) de los circuitos necesarios para finalmente construir el prototipo de PLC y realizar las pruebas operativas para verificar su correcto funcionamiento.

Con la realización de este proyecto se obtuvo un dispositivo capaz de realizar cualquier acción de control o automatismo, ya sea a nivel educativo contribuyendo con el entrenamiento de estudiantes de Ingeniería Eléctrica, reforzando sus conocimientos prácticos en el área de automatización y control de procesos industriales, en el manejo del PLC como dispositivo de control; como a nivel industrial colaborando con las pequeñas empresas de nuestro país a cubrir las necesidades básicas (en cuanto a automatización y control de procesos), promoviendo así la producción nacional. Este trabajo se ha estructurado en cinco capítulos con la finalidad de integrar y organizar todos los elementos desarrollados en el mismo.

En el Capítulo I se desarrolla el planteamiento del problema, justificación, objetivo general y específicos, además de establecer el alcance de la investigación.



El Capítulo II, comprende una recopilación de las bases teóricas necesarias para sustentar y respaldar el presente proyecto.

El Capítulo III está conformado por el procedimiento metodológico que describe los pasos seguidos en el desarrollo de cada uno de los objetivos trazados, describiendo por etapas las actividades realizadas para obtener el equipo propuesto.

En el Capítulo IV se presentan los resultados obtenidos: Características eléctricas, de operación, físicas, de funcionamiento, de configuración.

En el Capítulo V se presentan las conclusiones y recomendaciones derivadas del desarrollo del presente trabajo especial de grado.

La sección de anexos contiene información adicional referente a las hojas técnicas de los componentes electrónicos utilizados en la construcción del PLC, manual de configuración del PLC887, manual del Programa MonitorPLC887 y los manuales de funciones básicas del Proton IDE y LDMicro, y las prácticas de laboratorio.



CAPÍTULO I

EL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

La utilización de Controladores Lógicos Programables (PLC), en los sistemas de control industrial se ha extendido tan rápido en los últimos años que, en la actualidad, es difícil encontrar un ambiente industrial en el que no se empleen estos dispositivos electrónicos. Se puede afirmar que casi todo tipo de control de procesos, se puede realizar con un PLC, ésta es justamente, la razón por la cual son tan populares [1].

El PLC representa un factor clave en la automatización industrial en lo que refiere a control de procesos. Este sistema de control cuenta con una programación simple, la cual puede variar sin realizar cambios en el hardware. Dentro de las características de los sistemas basados en PLC se puede destacar: son pequeños, económicos y tienen mayor confiabilidad que los sistemas de control basados en relés, bajo costo de mantenimiento y su uso permite una flexible adaptación a diferentes procesos [2].

Los PLC son computadores “pre-empaquetados” capaces de resistir los riesgos de un entorno industrial. En concreto, sus entradas y salidas están aisladas ópticamente, además las salidas están íntegramente protegidas, y sus componentes internos son eléctricamente inmunes al ruido presente en el entorno industrial [3].

De esta manera, los fabricantes de PLC han desarrollado familias de productos que comprenden equipos desde unas pocas entradas/salidas hasta grandes controladores capaces de gobernar unas miles, con memorias de hasta 5MB o incluso



más. El campo de aplicación cubre desde el mínimo nivel de automatización de una secuencia de enclavamientos hasta el control completo de un proceso de producción continua [4].

En esta investigación se hace referencia a los PLC de gama baja, los cuales tienen las siguientes características: pueden contener hasta 128 entradas/salidas y memoria de 1KB a 4KB, así como también, funciones básicas de control de variables binarias, enteras y reales, temporizadores, contadores y registros, operaciones aritméticas, trigonométricas, logarítmicas, entre otras, y manipulación de gran cantidad de datos [5].

Actualmente, existe un elemento electrónico capaz de ejecutar órdenes o secuencias que estén grabadas en su memoria, este es conocido como microcontrolador, el cual, es un circuito integrado programable que contiene todos los componentes de un computador. Estos incluyen dentro del mismo encapsulado un procesador, memoria RAM, memoria ROM, puertos de entrada y salida; y pueden ser fácilmente programados para llevar a cabo diversas tareas [6].

Normalmente se emplean para controlar el funcionamiento de una tarea determinada y debido a su reducido tamaño, suelen ir incorporados en el propio dispositivo al que gobiernan. Esta última característica es la que les confiere la denominación de Controlador Incrustado (*Embedded Controller*) [6].

El número de productos que funcionan en base a uno o varios microcontroladores aumenta de forma exponencial. No es aventurado pronosticar que en el siglo XXI habrá pocos elementos que carezcan de un microcontrolador. En esta línea de prospección del futuro, la empresa Dataquest calcula que en cada hogar existirán varios centenares de microcontroladores en los comienzos del tercer milenio [6].



El uso de microcontroladores está tomando parte importante en el ámbito industrial [6], por lo que la idea de realizar un PLC con un microcontrolador no escapa de la realidad. En efecto, algunos fabricantes de microcontroladores consideran que el uso de estos dispositivos para la implementación de un PLC representa la siguiente etapa en la evolución del mercado de los PLC de gama baja [3].

Considerando que las características de los microcontroladores modernos, en cuanto a cantidad de entradas y salidas, velocidad de procesamiento y recursos periféricos disponibles, cumplen con los requerimientos básicos para realizar un PLC (de gama baja) se propuso el desarrollo de un Controlador Lógico Programable (PLC) basado en un microcontrolador PIC, el cual podrá ser empleado para aplicaciones industriales y didácticas dentro de los laboratorios de la Universidad de Carabobo.

1.2 JUSTIFICACIÓN DE LA INVESTIGACIÓN

Actualmente en Venezuela existen leyes que dan importancia y promueven el desarrollo tecnológico, como por ejemplo, la Ley Orgánica de Ciencia, Tecnología e Innovación, la cual en su artículo 24 establece que, “el Ministerio de Ciencia y Tecnología creará mecanismos de apoyo, promoción y difusión de invenciones e innovaciones populares, propiciando su transformación en procesos, sistemas o productos que generen beneficios a la población o logren un impacto económico o social” [7]. Otra ley que se puede resaltar es la Ley de Contrataciones Públicas Decreto N° 5.929 publicada en la gaceta oficial N° 38.895 de fecha del 25 de marzo de 2008, que fortalece la importancia de la producción nacional, otorgándole prioridad a las nuevas tecnologías desarrolladas en el país [8].

Tomando en cuenta el marco legal existente en el país en cuanto al desarrollo tecnológico, se propuso la creación de un prototipo de PLC basado en un



microcontrolador PIC16F887 con características similares a los PLC (de gama baja) existentes en el mercado, capaz de ser utilizado tanto en la industria como para fines educativos.

Con el prototipo de PLC desarrollado se pueden cubrir las necesidades básicas (en cuanto a automatización y control de procesos) de pequeñas empresas de nuestro país, promoviendo así la producción nacional, ofreciendo un producto de calidad y bajo costo; así como también, contribuir con el entrenamiento de los estudiantes de Ingeniería Eléctrica de la Universidad de Carabobo, reforzando sus conocimientos prácticos en el área de automatización y control de procesos industriales, en el manejo del PLC como dispositivo de control.

Las características de los PLC varían desde su número de entradas y salidas hasta el tamaño de la memoria. El PLC desarrollado al estar basado en un microcontrolador PIC, está condicionado a las características que este posee, otorgando ventajas como: incorporar salidas analógicas, que no todos los PLC cuentan con ellas, como por ejemplo los módulos lógicos AC010 de ABB [9] o el LOGO-230RC de SIEMENS [10], los cuales son usados para tareas de control de pequeña y mediana escala.

En el prototipo a construir se incorpora la capacidad de ser utilizado en red mediante el protocolo de comunicación 1-Wire®, considerando que este protocolo no ha sido utilizado en ningún PLC comercial de gama baja de los estudiados por los autores (ver anexo A), se considera que su integración al prototipo es un aporte valioso al uso de esta tecnología en ambientes industriales. Este tipo de comunicación es de bajo costo por solo usar un conductor de datos y su referencia [11].

El PLC construido cuenta con la posibilidad de ser programado en lenguaje escalera (*Ladder*), según el estándar IEC 61131-3 de Programación de Sistemas de



Automatización Industrial [12] y en cualquier lenguaje de programación utilizado para microcontroladores PIC. El proyecto incluyó el desarrollo de ejemplos de programación tanto en lenguaje escalera (*Ladder*) como en *Proton IDE*.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar un prototipo de Controlador Lógico Programable (PLC) modular de gama baja basado en un microcontrolador PIC para aplicaciones industriales y didácticas con capacidad de comunicación en red mediante el protocolo 1-Wire®.

1.3.2 OBJETIVOS ESPECÍFICOS

- 1 Definir la estructura y funcionalidad del equipo atendiendo a las características existentes en diferentes PLC de gama baja obtenidas mediante la revisión bibliográfica de sus manuales técnicos.
- 2 Desarrollar los diagramas esquemáticos y de circuito impreso requeridos por las interfaces de Entrada/Salida digitales y analógicas.
- 3 Incorporar el sistema de comunicaciones para la interconexión en red y monitoreo de señales provenientes del PLC utilizando la red 1-Wire®.
- 4 Construir el prototipo del PLC para efectuar la verificación de su correcto funcionamiento mediante la realización de diferentes pruebas operativas.
- 5 Adaptar un compilador de lenguaje escalera distribuido bajo licencia GNU acorde con el prototipo de PLC construido para la programación del mismo.



- 6 Elaborar un conjunto de tres prácticas de laboratorio que puedan ser utilizadas para el entrenamiento de estudiantes de Ingeniería Eléctrica de la Universidad de Carabobo.

1.4 ALCANCE Y LIMITACIONES

El prototipo desarrollado contempló la construcción de un Controlador Lógico Programable (PLC) con fines didácticos e industriales. Dicho PLC es modular y está compuesto por un módulo principal de 2 entradas y 2 salidas analógicas y 8 entradas/salidas digitales, un módulo de expansión analógico de 4 entradas, un módulo de expansión digital de 6 entradas/salidas y un módulo de teclado. Se programa en lenguaje escalera (*Ladder*) [12] con una versión especial del programa *LDmicro* modificada y adaptada por los autores y en cualquier lenguaje de programación utilizado para microcontroladores PIC.

Se utilizó el microcontrolador PIC16F887, el cual cumple con los requerimientos básicos para realizar un PLC de gama baja, por poseer las siguientes características [13]:

- Módulo convertidor Analógico/Digital de 10 bits.
- Canales de entradas analógicas.
- 24 Canales de entrada/salida Digital.
- Frecuencia de operación de 20Mhz (una instrucción se ejecuta en 200ns).
- 8KB de memoria de programa.
- 2 Módulos para PWM (*Pulse Width Modulation* - Modulación del Ancho de Pulso).
- Programación en circuito "*In-Circuit Serial Programming* (ICSP)".



- Comunicación serial USART (*Universal Synchronous Asynchronous Receiver Transmitter* - Transmisor Receptor Universal Síncrono Asíncrono).

El prototipo de PLC construido puede ser monitoreado desde un PC convencional a través del protocolo de comunicación 1-Wire®. Para esto, se hace uso de un servidor de datos OPC (OmniServer Pro) y se provee de una aplicación monitor llamada “MonitorPLC887”. Este proyecto contempló el diseño de la plataforma de hardware necesaria para permitir la comunicación 1-Wire® y el desarrollo de la aplicación de monitoreo, no incluyendo la implementación del servidor de datos 1-Wire® a OPC.



CAPÍTULO II

MARCO TEÓRICO

2.1 ANTECEDENTES

- MikroElektronika es una empresa especializada en electrónica desde 1997 ubicada en Belgrado - Serbia, esta produce una amplia gama de herramientas de desarrollo y compiladores para diferentes familias de microcontroladores. Entre las herramientas de desarrollo se encuentran el **PICPLC8A** [14] y **PICPLC16B** [15], estos son sistemas diseñados para el control de dispositivos industriales y automáticos mediante el uso de salidas basadas a relés, entradas aisladas por optoacopladores y como unidad de control utilizan microcontroladores PIC Microchip® de encapsulados DIP40. Tomando como referencia los criterios de diseño de los modelos consultados, se extrajo la idea de utilizar como unidad de control un microcontrolador PIC Microchip®, se propuso disminuir las dimensiones del dispositivo y manejar las entradas y las salidas como un mismo elemento.
- Otro dispositivo desarrollado por la empresa MikroElektronika para el desarrollo de control industrial y el hogar o la oficina, es el **PICPLC16 BOX** [16]. Este además de tener como unidad de control un microcontrolador PIC Microchip®, manejar sus salidas mediante relés y tener entradas aisladas con optoacopladores, tiene un teclado para la configuración y control del dispositivo y una pantalla LCD. De este sistema se extrajo la idea de colocar un pequeño teclado con la misma finalidad y una pantalla LCD.



-
- En Waterloo, Canadá durante el año 2005, Jonathan Westhues, publicó un proyecto de software llamado LDmicro, con el cual se puede editar, compilar y simular la lógica de escalera (usada para la programación de PLC) para microcontroladores de 8 bits. Este programa es capaz de generar códigos entendibles por microcontroladores Atmel AVR y Microchip PIC16, a partir del diagrama de escalera [17]. La contribución de este proyecto para la presente investigación lo constituye la idea de programar el microcontrolador PIC Microchip® con el software LDmicro. Este programa se distribuye en forma de software libre y su código fuente fue adaptado para ajustarlo a los requerimientos de este trabajo.



2.2 CONTROLADOR LÓGICO PROGRAMABLE (PLC)

2.2.1 Introducción

La mayoría de los procesos industriales requieren de algún tipo de coordinación, supervisión o control. La necesaria automatización de estas funciones puede ser llevada a cabo de diversas maneras, pero hasta hace algunos años, la práctica común consistía en el control de secuencias de operación en base a sistemas de relés y la utilización de módulos especiales para el control de variables continuas (como la temperatura) y de tableros indicadores para proveer la interfaz con un operador supervisor. El problema de estos sistemas basados en relés era que cuando los requerimientos de producción cambiaban, también lo hacía el sistema de control, esto comenzó a elevar los costos de mantenimiento cuando los cambios eran frecuentes. Por otra parte, en ocasiones se debían realizar conexiones entre cientos o miles de relés, lo que implicaba un enorme esfuerzo de diseño y mantenimiento [1,5].

Es entonces cuando el primer Controlador Lógico Programable hace su aparición a finales de 1960, gracias a un grupo de ingenieros de General Motors, como solución al problema que presentaban los sistemas de control basados en relés, ofreciendo muchas ventajas como lo son: un costo efectivo para el control de sistemas complejos, puede ser aplicable nuevamente a otros sistemas de control de forma rápida y sencilla, posee capacidades computacionales que permiten un control más sofisticado, fácil programación que reduce el tiempo de inactividad y solución de problemas, los confiables componentes que conforman al Controlador Lógico Programable hacen que pueda funcionar durante años antes de presentar una falla [2,18,19].

Se puede decir que un Controlador Lógico Programable es un dispositivo electrónico que sustituye los circuitos auxiliares o de mando de los sistemas

automáticos. A él se conectan los sensores (finales de carrera, pulsadores, etc.) por una parte, y los actuadores (bobinas de contactores, lámparas, etc.) por otra [5].

2.2.2 Definición

Se conoce como Controlador Lógico Programable (PLC), o Autómata Programable a todo circuito electrónico diseñado para controlar en tiempo real máquinas, sistemas o procesos secuenciales dentro de un entorno industrial. Para controlar los procesos dispone de puertos de conexión de entradas y salidas, en los que se conectan directamente los elementos primarios y finales de control. Además, es capaz de memorizar el programa de usuario construido por instrucciones lógicas, secuenciales y cíclicas, según las cuales y de acuerdo al estado de las entradas, permite la manipulación del estado de las salidas. A continuación se ilustra un diagrama conceptual del uso del PLC (ver figura 2.1).

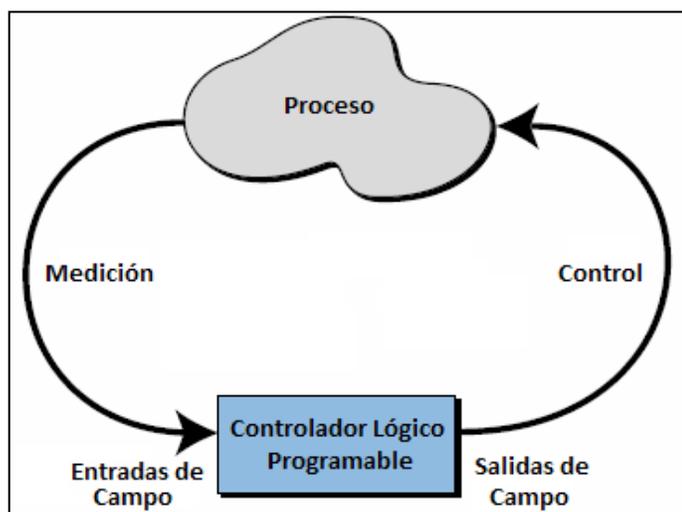


Figura 2.1 Diagrama Conceptual del uso del PLC.

Fuente: Programmable Controllers Theory and Implementation. [20]

Los Controladores Lógicos Programables pueden llegar a ser, en términos simples, computadoras industriales con arquitecturas basadas en unidades centrales



(el PLC como tal), y en circuitos de interconexión con los dispositivos del proceso (conexiones de entrada - salida) [20].

2.2.3 Clasificación

Hoy día, los Controladores Lógicos Programables disponibles comercialmente suelen clasificarse de diferente manera, muchas veces dependiendo del propio fabricante. Esta clasificación suele ser más bien flexible y poco rigurosa, pues el hecho de clasificar un autómatas dentro de una cierta categoría depende de multitud de factores tales como: la potencia de su CPU, disponibilidad de módulos de ampliación de entrada/salida, tamaño de la memoria e incluso de factores comerciales. Sin embargo, atendiendo al número de entradas y salidas que dispone un Controlador Lógico Programable, la clasificación genérica se suele hacer en tres tipos o categorías [5]:

- **Controladores Lógicos Programables de gama baja:**
 - Hasta un máximo de 128 entradas/salidas.
 - La memoria de usuario disponible es de hasta 4K instrucciones.

- **Controladores Lógicos Programables de gama media:**
 - Entre 128 y 512 entradas/salidas.
 - La memoria de usuario disponible alcanza hasta 16K instrucciones.

- **Controladores Lógicos Programables de gama alta:**
 - Más de 512 entradas/salidas.

La memoria de usuario disponible es superior a las 16K instrucciones, e incluso en algunos casos puede llegar a superar las 100K instrucciones.



2.2.4 Estructura

2.2.4.1 Estructura Externa

El término estructura externa o configuración externa de un Controlador Lógico Programable se refiere al aspecto físico exterior del mismo, bloques o elementos en que está dividido [21].

Actualmente son tres las estructuras más significativas que existen en el mercado [21]:

- **Estructura Compacta:** Este tipo de autómatas se distingue por presentar en un sólo bloque todos sus elementos, como lo son la fuente de alimentación, el CPU, las memorias, las interfaces de entradas/salidas, etc. Son los autómatas de gama baja los que suelen tener una estructura compacta. Su potencia de proceso suele ser muy limitada dedicándose a controlar máquinas muy pequeñas o cuadros de mando.
- **Estructura Modular Americana (semi - modular):** Se caracteriza por separar las interfaces de entradas/salidas del resto del autómata, de tal forma que en un bloque compacto están reunidas las CPU, memoria de usuario o de programa y fuente de alimentación y separadamente las interfaces de entradas/salidas. Son los autómatas de gama media los que suelen tener una estructura modular americana (semi - modular).
- **Estructura Modular Europea:** Su característica principal es la de que existe un módulo para cada uno de los diferentes elementos que componen al autómata como pueden ser una fuente de alimentación, CPU, interfaces de



entradas/salidas, etc. La sujeción de los mismos se hace por medio de un riel DIN, una placa perforada o sobre un RACK, en donde va alojado el BUS externo de unión de los distintos módulos que lo componen. Son los autómatas de gama alta los que suelen tener este tipo de estructura modular, que permiten una gran flexibilidad en su constitución.

2.2.4.2 Estructura Interna

A continuación se presenta la estructura interna de los actuales PLC mediante el estudio de sus bloques constructivos primordiales: la unidad central de proceso (CPU), el sistema de memorias, las interfaces de entrada/salida y la fuente de alimentación [5].

2.2.5 Buses

Todos estos bloques (ver figura 2.2) se encuentran unidos mediante un conjunto interno de líneas eléctricas que está compuesto, a su vez, por tres subgrupos de líneas denominados buses. Estos tres buses son los siguientes [5]:

- **Bus de direcciones.** Las líneas por donde viajan las direcciones procedentes de la CPU a los diferentes bloques de memoria o interfaces de entrada/salida.
- **Bus de datos.** Las líneas por donde viajan los datos o códigos de instrucciones desde la CPU a los diferentes bloques de memoria o interfaces de entrada/salida o viceversa.
- **Bus de control.** Las líneas necesarias para que la CPU controle los bloques de memoria o bloques de entrada/salida.

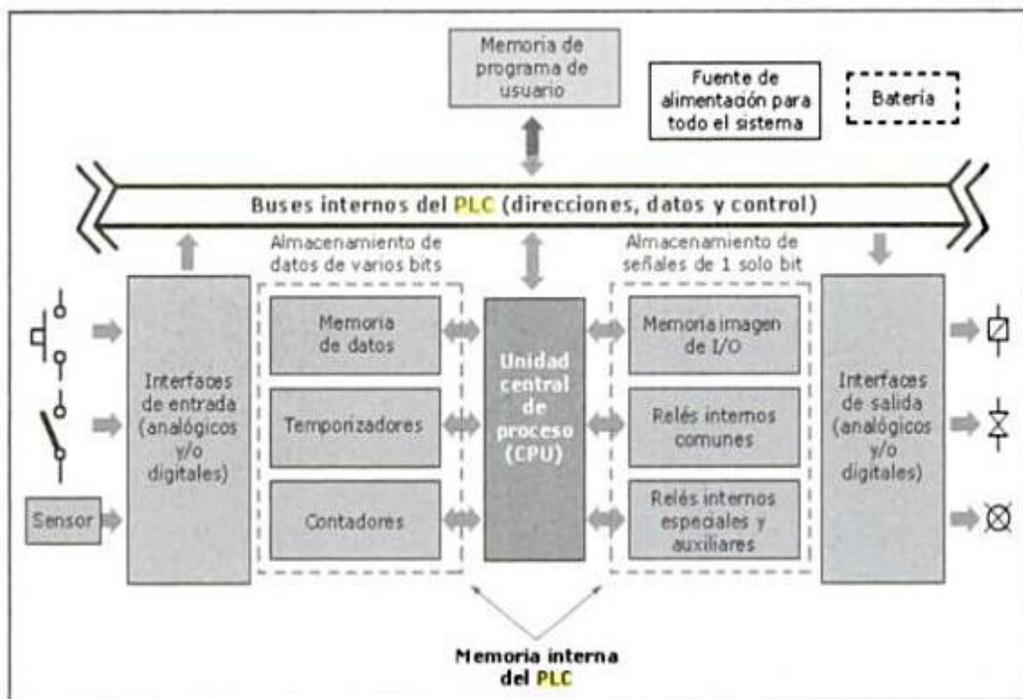


Figura 2.2 Diagrama de bloques simplificado en el que se detalla la estructura de los buses de un PLC estándar.

Fuente: Introducción a los Automatas Programables. [5]

2.2.6 Unidad Central de Proceso

La unidad central de proceso o CPU (*Control Process Unit*) se puede considerar el cerebro de todo el sistema. Está formada por dos bloques o unidades fundamentales [5]:

- **Unidad de Control:** Se encarga de generar todas las líneas de control y direcciones para el control y acceso correctos a la gran mayoría de los dispositivos conectados a la CPU.
- **Unidad Operativa:** Realiza aquellas operaciones lógicas, aritméticas y de movimientos de datos necesarias para la correcta ejecución del programa que tiene encomendado (es decir, almacenado previamente en la memoria de



usuario). Dicha unidad operativa está formada, entre otros elementos, por una serie de registros internos y por un módulo principal denominado unidad aritmético-lógica o ALU (*Arithmetic Logic Unit*). La ALU es el “centro de cálculo” lógico y aritmético del PLC. La CPU en su conjunto tiene como misiones fundamentales [5]:

- Consultar el estado de las diferentes entradas que existen en el PLC.
- Recoger la secuencia de instrucciones que hay que ejecutar de la memoria de programa secuencialmente.
- Elaborar, a partir del programa y las entradas obtenidas, las señales u órdenes que se enviarán al proceso por medio de las interfaces de salida del PLC.
- Actualizar continuamente los temporizadores y contadores internos del programa de usuario.

2.2.7 Sistema de Memorias

La memoria del PLC es la parte donde se almacena toda la información (datos, constantes, instrucciones, etc.) necesaria para el correcto funcionamiento del sistema de control. Los datos que contienen las memorias internas del PLC son básicamente de dos tipos [5,21]:

- **Datos de proceso:**
 - Señales provenientes del proceso o hacia éste, es decir, las entradas y salidas del PLC respectivamente.
 - Variables internas, de bit y de palabra.
 - Datos alfanuméricos y constantes.



- **Datos de control:**

- Instrucciones de usuario (programa de usuario).
- Datos de configuración del propio PLC (modo de funcionamiento, número de entradas/salidas conectadas, etc.).

En general, toda esta información, incluyendo el programa de usuario, está contenida en memorias de semiconductor. Una memoria de semiconductor es un dispositivo electrónico capaz de almacenar datos binarios (señales de niveles alto y bajo, “unos” y “ceros”, determinadas bits), que pueden ser leídos posición a posición (bit a bit), o por bloques de ocho (byte) o dieciséis (word) posiciones. La memoria ideal para el PLC debería ser simultáneamente rápida, pequeña, barata y de bajo consumo de energía. Como ninguna de las memorias del mercado reúne todas estas condiciones, los PLC combinan distintos tipos de ellas [21].

2.2.8 Tipos de Memoria y Aplicaciones en los PLC

Dentro de las actuales tecnologías de fabricación de circuitos integrados de memoria existentes en el mercado y atendiendo a sus características de lectura y escritura, se puede realizar una clasificación de memorias como la mostrada a continuación [5,21]:

- **Memoria de acceso aleatorio o RAM (*Random-Access Memory*):**

- Memoria volátil. Si la alimentación cesa, los datos almacenados en ella desaparecen.
- Si son memorias estáticas (sin necesidad de refresco de la información por parte de la CPU), la densidad de integración es relativamente baja, es decir, pocos bits de almacenamiento por unidad de área de silicio empleada.



-
- Si son unidades dinámicas (en las que existe la necesidad de que la CPU refresque la información periódicamente), la densidad de integración es bastante elevada. Este tipo de memorias son más modernas y avanzadas que las estáticas.
 - Es utilizada para datos internos cuya desaparición en caso de cortarse la alimentación no sea crítica.
 - Añadiendo una batería puede memorizar el programa de usuario y todos aquellos datos que haya que mantener de forma permanente.
 - En caso de llevar acoplada esta batería, esta última tendrá que ser cambiada cada cierto tiempo con el fin de que los datos de la memoria RAM no desaparezcan.
- **Memoria de sólo lectura o ROM (*Read-Only Memory*):**
 - Sólo puede ser leída, nunca escrita por el usuario. En general es el propio fabricante quien la suministra convenientemente programada.
 - Cuando se incluye en un PLC es una memoria no accesible desde el exterior (ni por el usuario ni por el programador).
 - Suele contener instrucciones de rutinas necesarias por el PLC, tales como sentencias de inicialización, test y auto diagnóstico, programa monitor de inicio y programa intérprete, etc.
 - **Memoria de sólo lectura programable o PROM (*Programmable ROM*):**
 - El usuario puede escribirla una única vez (mediante el programador adecuado). Es, por tanto, un dispositivo denominado *OTP Device (One Time Programmable)*.
 - Una vez programada solo puede ser leída.
 - También es una memoria no accesible desde el exterior, ni por el usuario ni por el programador del PLC.



- Contiene normalmente rutinas de inicialización, test y auto diagnóstico, programa monitor e intérprete, etc.
- **Memoria de sólo lectura programable y borrable o EPROM (*Erasable PROM*):**
 - Borrada por rayos ultravioleta (UV) tras someterla a dicha fuente luminosa entre 5 y 20 minutos a una ventana superior en el cuerpo del circuito integrado.
 - Posibilidad de programación una vez borrada mediante el programador adecuado.
 - Poco utilizada en los PLC, aunque puede almacenar programas de usuario (tras haber sido convenientemente depurados).
- **Memoria de sólo lectura programable y borrable eléctricamente o EEPROM (*Electrically Erasable PROM*) y EAROM (*Electrically Alterable ROM*):**
 - Memoria que puede ser borrada por señales eléctricas sin la necesidad de fuentes de rayos UV.
 - En la actualidad sigue siendo una memoria relativamente lenta (sobre todo en escritura).
 - Los modelos actuales tienen limitado el número de ciclos de borrado/escritura (a decenas o cientos de miles).
 - Su precio sigue siendo elevado, aunque las expectativas son que vaya descendiendo progresivamente.
 - Pueden almacenar el programa de usuario.
 - Gran versatilidad junto con una RAM. Durante el funcionamiento del PLC, los datos se guardarán siempre sobre la RAM, pero en caso de que

la alimentación sea menor de un cierto umbral, los datos serán automáticamente traspasados a una EEPROM.

Tabla 2.1 *Tabla de clasificación de las memorias en función del tipo que se utilice en un PLC y de la clase de datos almacenados.*

Tipo de memoria	Volátil	No Volátil			
Lectura/Escritura	RAM		EPROM	RAM + Batería RAM + EEPROM	EEPROM
Sólo Lectura		ROM/PROM			
Aplicaciones	Datos internos. Imagen de entrada/salida	Programa monitor e intérprete. Rutinas de test y auto diagnóstico.	Programa de usuario.	Programa de usuario. Datos internos y parámetros varios mantenidos	Programa de usuario. Parámetros varios.

Fuente: Introducción a los Autómatas Programables. [5]

2.2.9 Memoria del PLC según los datos almacenados

Según lo que se puede observar en el diagrama de bloques de la estructura interna de un PLC (ver figura 2.2), la memoria del mismo puede dividirse en dos grandes grupos atendiendo a los datos que almacena [5].

2.2.9.1 Memoria de Programa de Usuario

En la memoria de programa se almacena el programa de usuario que se va a ejecutar. En el argot común de los PLC, la capacidad de dicha memoria normalmente no se mide en Kbits (1.024 bits) ni en Kbytes (8.192 bits), sino en K instrucciones que puede contener. Cada una de estas instrucciones ocupa por norma general como mínimo 2 bytes de información. En la mayoría de los PLC, el espacio de memoria va



desde en torno a 1K instrucciones en los de gama baja hasta varios cientos de K instrucciones en los de gama alta.

Desde el punto de vista físico, esta memoria de programa puede ser tanto interna (en PLC compactos de gama baja) como disponible en módulos externos (en PLC modulares).

En los PLC comerciales, esta memoria es necesariamente permanente para, de este modo, no perder el programa en caso de desconexión o fallo de la alimentación del PLC. De acuerdo a lo indicado por [5], puede estar formada por:

- Una RAM + Batería. Requiere un mantenimiento de la batería para que un deterioro de la misma no provoque pérdidas irreparables de información en la memoria.
- Sólo módulos de memoria EPROM/EEPROM. Es la solución menos común, debido a que en la actualidad el número de ciclos de reescritura sobre el circuito integrado es limitado y requieren un tiempo de acceso para su lectura relativamente elevado.
- Conjunto de memoria RAM sin batería y memoria EEPROM. Esta asociación permite siempre trabajar en la memoria de usuario con módulos RAM (de elevada velocidad) para, en el momento en que la CPU detecta que la alimentación pasa por debajo de cierto umbral (debido, por ejemplo, a un corte del suministro eléctrico), volcar la información sobre la EEPROM con el objetivo de que nunca se pierda la información contenida en la memoria de usuario.
- Conjunto de memoria RAM + Batería y memoria EPROM o EEPROM. Disponible en los PLC más modernos. Esta última solución es la más versátil, pues permite programar el PLC sobre la memoria RAM en la fase de



desarrollo y depurado del programa para, una vez finalizada, volcar la información sobre la EPROM o EEPROM.

La ejecución del programa es siempre prioritaria, de forma que si se da tensión al PLC con un módulo conectado, la CPU ejecuta su programa y no el contenido en memoria RAM interna [21].

2.2.9.2 Memoria Interna

La memoria interna queda dividida en diferentes áreas o zonas donde se almacenan datos de diferentes tipos. Como se mencionó anteriormente las áreas o zonas más importantes son [5]:

- **Zona de almacenamiento de variables internas o señales de un solo bit.**

En esta zona se almacenan señales binarias que contienen información de un solo bit. Estas señales son principalmente:

- **Relés internos comunes.** Disponibles para marcas de programa, almacenamiento de variables internas de un solo bit, etc.
- **Relés internos especiales y auxiliares.** Disponibles para, entre otras funciones, marcas internas que conservan su valor aun con falta de alimentación del PLC, enclavamientos, básculas, etc.
- **Indicadores (denominados en inglés *flags*).** Señalizan algún posible error o alarma en el funcionamiento del sistema del PLC.

- **Zonas de almacenamiento de datos de varios bits (8 o más).**

En esta zona se almacenan datos codificados en varios bits. Esta información es principalmente:

- Temporizadores.



- Contadores.
- Registros de uso general y para almacenamiento de datos; disponibles, entre otros usos, para almacenar operandos y resultados en operaciones aritméticas y lógicas de varios bits, así como para almacenar señales de entradas analógicas que han sido digitalizadas por convertidores Analógico/Digital (A/D) o de salidas analógicas que serán digitalizadas por convertidores Digital/Analógico (D/A).

2.2.10 Interfaces de Entrada/Salida

Las interfaces de entrada y salida son los módulos o dispositivos del PLC encargados de realizar la unión o interfaz entre el proceso o planta y la CPU. En ellos se realizan una serie de acondicionamientos de la señal eléctrica que entra o sale del PLC [5].

En concreto, los módulos de entrada se encargan de [5]:

- Amplificar la señal de entrada proveniente del dispositivo sensor.
- Adaptar los niveles de tensión de dicha señal.
- Filtrar la señal.
- Codificar la señal para que sea procesada de forma adecuada por la CPU.

Por otra parte, los módulos de salida se encargan de:

- Decodificar la información proveniente de la memoria imagen de salida.
- Amplificar la señal de salida decodificada.
- Adaptar los niveles de tensión de esa señal.



En la actualidad, para la mayoría de los PLC modulares existentes (que van montados sobre un soporte o *rack*) hay una gran variedad de módulos de entrada y salida, con lo cual, estas interfaces pueden ser clasificadas en forma muy diversa, es decir, atendiendo a diferentes criterios de selección. Por su parte, en PLC de gama baja, que acostumbran a ser compactos, los tipos de interfaces suelen ser mucho más limitados [5].

2.2.11 Clasificación de los Módulos de Entrada/Salida

- **Por el tipo de señales con las que trabaja:**
 - Digitales de 1 bit solamente.
 - Digitales de varios bits (por ejemplo, 1 byte o 2 bytes)
 - Módulos analógicos mediante convertidores A/D y D/A (típicamente de 10, 12 o 16 bits).

- **Por el aislamiento:**
 - Con separación galvánica (mediante un relé electromagnético o transformador) u óptica (por medio de opto acopladores) entre el PLC y la planta (sensores y actuadores).
 - Con acoplamiento directo.

- **Por su ubicación en el sistema PLC:**
 - Locales. Para PLC compactos, o cuando el módulo de entrada o de salida está montado en el mismo bastidor donde se halla la CPU en PLC modulares.
 - Remotos. Cuando el módulo de entrada o de salida está ubicado en un bastidor diferente y a cierta distancia del bastidor donde se encuentra la CPU.



- **Por la forma de comunicación con la CPU:**
 - Módulos con comunicación serie.
 - Módulos con comunicación en paralelo.

- **Módulos de entrada:**
 - DC a baja tensión (típicamente 12V, 24V, 48V).
 - AC a 60V, 110V o 220V.
 - Módulos analógicos de 0 - 10V ó 4 - 20mA.

- **Módulos de salida:**
 - Por relé electromagnético o salidas libres de tensión.
 - Estáticos con TRIAC o SCR para 110V o 220V.
 - Estáticos por transistor en colector abierto (*open collector*) NPN o PNP para 24V o 48V.
 - Módulos analógicos de 0 - 10V ó 4 - 20mA [5].

2.2.12 Fuente de Alimentación

La fuente de alimentación de PLC proporciona las tensiones necesarias para el funcionamiento de los distintos circuitos del sistema de control, tanto del propio PLC como de los elementos que lo rodean como pueden ser sensores y actuadores.

En los primeros PLC y en algunos de gama baja actuales la alimentación existente es única, aunque hoy en día en la mayoría de los modelos comerciales se independiza la alimentación interna del PLC de la fuente auxiliar que suministra alimentación a otros elementos del proceso. De esta forma, además de generar tensiones diferentes adecuadas a cada módulo o unidad, se minimiza el efecto de interferencias electromagnéticas acopladas por medio de la alimentación, provocadas



por elementos de potencia (como actuadores electromecánicos, por ejemplo) sobre circuitería de baja potencia (como lo es la CPU del PLC).

Así pues, en general los sistemas PLC poseen tres sistemas independientes de alimentación:

- Alimentación del propio PLC (CPU, interfaces de Entrada/Salida, etc.),
- Alimentación de las entradas (sensores),
- Alimentación de las salidas (cargas) del tipo electromagnético (actuadores).

En los PLC compactos, esta fuente auxiliar que proporciona el PLC para alimentar sensores o actuadores suele tener niveles de tensión de 24V o 48V, pero proporciona corrientes muy limitadas (inferiores a un amperio en la gran mayoría de los modelos comerciales) [5].

2.2.13 Funcionamiento General de un PLC

El PLC está diseñado para realizar un conjunto de actividades de forma repetitiva durante la ejecución del programa de usuario. A este conjunto de tareas que se ejecutan cíclicamente es lo que se conoce como ciclo de *scan* o exploración.

En este ciclo de exploración, la unidad de control del PLC debe cíclicamente leer los datos de entrada de los dispositivos del proceso por medio de la interfaz de entrada y almacenar su estado para ser utilizado durante todo el ciclo, así como ejecutar el programa de usuario en el orden en que fue estructurado desde la primera hasta la última línea de instrucciones. Además, deberá procesar cualquier señal recibida por el puerto de comunicación y por último actualizar el estado de las salidas por medio de la interfaz de salida (ver figura 2.3) [20,21].

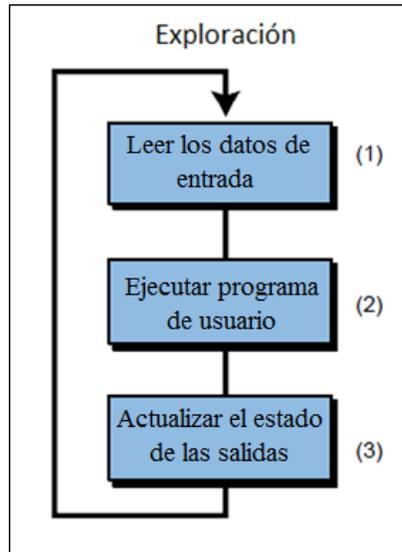


Figura 2.3 Representación gráfica de una exploración.

Fuente: Programmable Controllers Theory and Implementation. [20]

2.2.14 Tipos de Lenguajes de Programación para PLC

Cuando surgieron los autómatas programables, lo hicieron con la necesidad de sustituir a los sistemas de control contruidos con contactores y relés. Por lo tanto, la comunicación hombre-máquina debería ser similar a la utilizada hasta ese momento. El lenguaje usado, debería ser interpretado con facilidad, por los mismos técnicos electricistas que anteriormente estaban en contacto con la instalación. Estos lenguajes han evolucionado en los últimos tiempos, de tal forma que algunos de ellos ya no tienen nada que ver con el típico plano eléctrico a relés [2,12]. Los lenguajes más significativos según el Estándar IEC 61131-3 de Programación de Sistemas de Automatización Industrial [12] son:

2.2.14.1 Diagrama de Contactos (*Ladder - LD*)

Este es el método de programación más usual, diseñado para permitir una programación de manera sencilla basada en símbolos y esquemas con los que el personal técnico está familiarizado, pues se basa en símbolos del diagrama de control electromecánico (ver figura 2.4) [1].

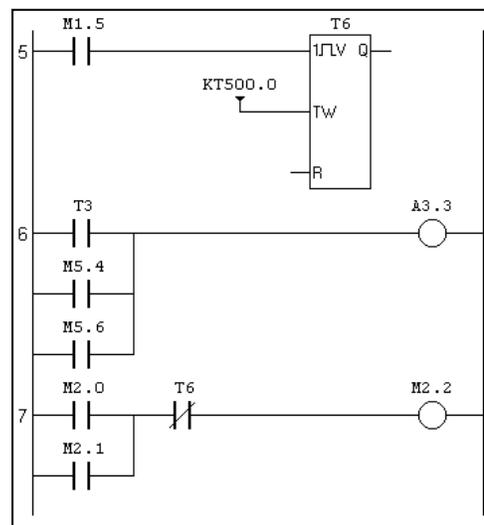


Figura 2.4 Ejemplo de programa con diagrama de contactos.
Fuente: Estándar de Programación de Sistemas de Automatización Industrial IEC-61131-3. [12]

2.2.14.2 Plano de Funciones Lógicas (*Function Block Diagram - FBD*)

El plano de funciones lógicas, resulta especialmente cómodo de utilizar, a técnicos habituados a trabajar con circuitos de puertas lógicas, ya que la simbología usada en ambos es equivalente (ver figura 2.5) [2].

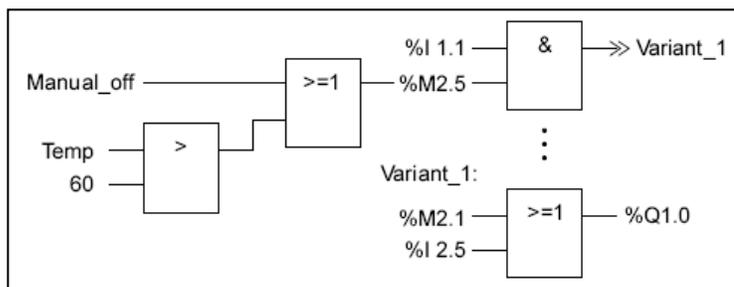


Figura 2.5 Ejemplo de Símbolos con Lenguaje de Plano de Funciones Lógicas (FBD).

Fuente: Estándar de Programación de Sistemas de Automatización Industrial IEC-61131-3. [12]

2.2.14.3 Lista de Instrucciones (*Instruction List - IL*)

Es una representación mnemotécnica de las instrucciones, enlistadas en secuencia y que permiten, sin la necesidad de una interfaz gráfica, el acceso a la programación del dispositivo. Es lo más parecido al “lenguaje natural o nativo” del procesador, que se maneja. Hoy en día, es una alternativa de programación y de depuración del programa, y permite el máximo nivel de explotación de las ventajas del PLC. Aunque representa un nivel mayor de complejidad (ver figura 2.6) [1].

000	LD	%I0.1	Bp. inicio ciclo
	AND	%I0.0	Dp. presencia vehículo
	AND	%M3	Bit autorización reloj calendario
	AND	%I0.5	Fc. alto rodillo
	AND	%I0.4	Fc. detrás pórtico
005	S	%M0	Memo inicio ciclo
	LD	%M2	
	AND	%I0.5	
	OR	%I0.2	Bp. parada ciclo
	R	%M0	
010	LD	%M0	
	ST	%Q0.0	Piloto ciclo

Figura 2.6 Ejemplo de Programa con Lenguaje de Instrucciones (IL).

Fuente: Estándar de Programación de Sistemas de Automatización Industrial IEC-61131-3. [12]

2.2.14.4 Lenguaje Estructurado (*Structured Text - ST*)

Consiste en la posibilidad de utilizar un lenguaje de alto nivel como “C” o “BASIC” en cualquiera de sus versiones, o en algún otro lenguaje exclusivo (variante de estos) adaptado por el fabricante. Las instrucciones pueden ser: selección de instrucciones, como IF ... THEN ... ELSE etc., repetición de instrucciones, tales como FOR, WHILE etc., entre otras (ver figura 2.7) [1,2].

```
IF %M3 THEN
  FOR %MW99:=0 TO 31 DO
    IF %MW100[%MW99]<>0 THEN
      %MW10:=%MW100[%MW99];
      %MW11:=%MW99;
      %M1:=TRUE;
      EXIT;
    ELSE
      %M1:=FALSE;
    END_IF;
  END_FOR;
ELSE
  %M1:=FALSE;
END_IF;
```

Figura 2.7 Ejemplo de programa con Lenguaje Estructurado (ST).

Fuente: Estándar de Programación de Sistemas de Automatización Industrial IEC-61131-3. [12]

2.2.14.5 Gráfico de Funciones Secuenciales (*Sequential Function Chart*)

Es un lenguaje gráfico para describir ciclos automáticos mediante símbolos. Los elementos del SFC, son pasos y transiciones. Cada paso representa un estado de procesamiento de un programa de control, los cuales pueden ser activos o inactivos (ver figura 2.8) [2].

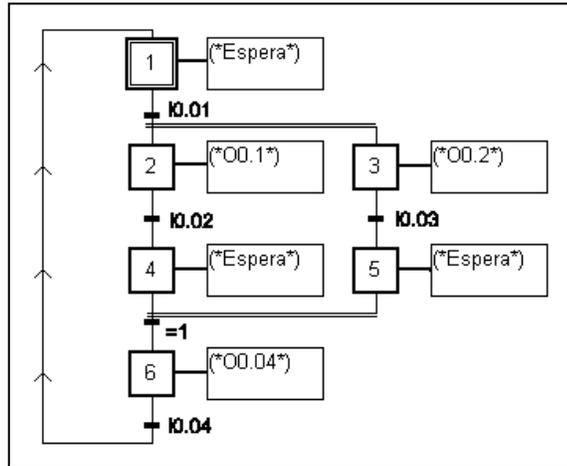


Figura 2.8 Ejemplo de programa realizado en SFC.
Fuente: Estándar de Programación de Sistemas de Automatización Industrial IEC-61131-3. [12]

2.2.14.6 Aplicación en la Industria

El PLC por sus especiales características de diseño tiene un campo de aplicación muy extenso. La continua evolución del hardware y software amplía constantemente este campo para poder satisfacer las necesidades que se detectan en el espectro de sus posibilidades reales.

Su utilización se da fundamentalmente en aquellas instalaciones en donde es necesario un proceso de maniobra, control, señalización, etc., por tanto, su aplicación abarca desde procesos de fabricación industriales de cualquier tipo a transformaciones industriales, control de instalaciones, entre otras.

Actualmente los controladores programables cuentan con posibilidades de expansión de módulos opcionales de entradas y/o salidas (módulos Entrada/Salida) ya sean de tipo digital o analógico, o módulos de salida de relé o de estado sólido. Por todas estas razones se puede afirmar que los PLC pueden resolver cualquier problema de aplicación de control de procesos industriales.



El campo de aplicación de los PLC se ha extendido enormemente, y suelen emplearse en procesos industriales que tengan una o varias de las siguientes necesidades [1,5,20,21]:

- Espacio reducido.
- Utilización en ambientes exigentes o agresivos.
- Procesos de producción periódicamente cambiantes.
- Procesos secuenciales.
- Maquinaria de procesos variables.
- Instalaciones de procesos amplios y complejos.
- Chequeo de programación centralizada de las partes del proceso.

Las aplicaciones típicas en el área de automatización de planta, manejo de materiales y robótica, son las siguientes:

- Maniobra de máquinas.
- Máquinas de inyección y otro tipo de prensas.
- Dispositivos de transferencia automática.
- Maquinaria industrial de plástico.
- Mecanismos de ensamble de los más variados tipos.
- El ambiente en casas y edificios, entre otras aplicaciones domésticas.
- Maniobra de instalaciones:
 - Instalación de aire acondicionado, calefacción, etc.
 - Instalaciones de seguridad.
 - Señalización y control.



2.2.15 Ventajas e Inconvenientes de los PLC

2.2.15.1 Ventajas

Fundamentalmente las ventajas que un PLC ofrece se basan en los siguientes puntos [5]:

- Tiempo reducido en el desarrollo de proyectos debido a que:
 - No es preciso dibujar esquemas de una forma detallada.
 - Al disponer de memoria suficiente para almacenar ecuaciones no simplificadas, no se requiere la simplificación de las ecuaciones lógicas.
 - Se utilizan menos materiales y componentes, con lo que el costo de la instalación es sumamente inferior al anularse proveedores, plazos de entrega, recambios, etc.
- Posibilidad de introducir modificaciones sin cambiar el cableado ni añadir elementos.
- Fácil reutilización. En caso de que el proceso controlado por el PLC quede obsoleto, el PLC es útil para incorporarlo a otro proceso.
- Costo de mano de obra en la instalación reducido.
- Incorpora un mínimo de elementos móviles.
- Posibilidad de gobierno de varias máquinas e instalaciones con un mismo autómatas y, caso de necesitar varios, fácil conexión entre éstos.
- Menor tiempo de puesta en servicio al reducirse el tiempo de la instalación en comparación con los sistemas de lógica cableada.
- Su constitución y funcionalidad son adaptables al número de elementos necesarios para cada caso de automatización, de forma que su tamaño no es fijo y el cambio o ampliación de módulos lo dota de una versatilidad difícilmente superable.



2.2.15.2 Inconvenientes

El PLC también posee ciertos inconvenientes, siendo el fundamental de ellos que precisa un programador mínimamente experimentado. En segunda instancia quedaría el costo inicial de la instalación (algo en cualquier caso cuestionable en función de su complejidad y naturaleza).

En resumen, las principales desventajas que surgen al llevar a cabo una instalación mediante autómatas programables son las mostradas a continuación de forma también sintetizada [5]:

- Se necesita personal especializado en la programación de una familia de autómatas o de un fabricante concreto.
- Falta de un lenguaje estándar y “universal” de programación.
- Alta incompatibilidad entre *software* y *hardware* para autómatas de diferentes fabricantes.
- Alto costo inicial (sobre todo si es necesario el uso de autómatas de gama media y alta).
- Su utilización no es posible en procesos con constantes de tiempo pequeñas o con señales de transiciones entre niveles lógicos producidas a altas frecuencias.



2.3 MICROCONTROLADOR

2.3.1 Definición

Es un circuito integrado programable que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una tarea determinada y, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de controlador incrustado (*embedded controller*) [6].

El microcontrolador es un computador dedicado. En su memoria sólo reside un programa destinado a gobernar una aplicación determinada; sus líneas de Entrada/Salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar, y todos los recursos complementarios disponibles. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada [6].

En general los microcontroladores disponen de los siguientes componentes [6]:

- Procesador o CPU.
- Memoria RAM.
- Memoria EEPROM.
- Líneas de entrada/salida.
- Módulos en hardware para el desempeño de tareas específicas y control de periféricos como: temporizadores, comunicación serial y paralela, convertidor Analógico/Digital, modulador de ancho de pulso (*PWM*), entre otros.

2.3.2 Tipos de Encapsulados

En general, se pueden encontrar encapsulados del tipo PDIP (Plastic Dual In Line Package), SOIC (Small-Outline Integrated Circuit), PLCC (Plastic Leaded Chip Carrier) y QFP (Quad Flat Package), (ver figura 2.9) [13]:

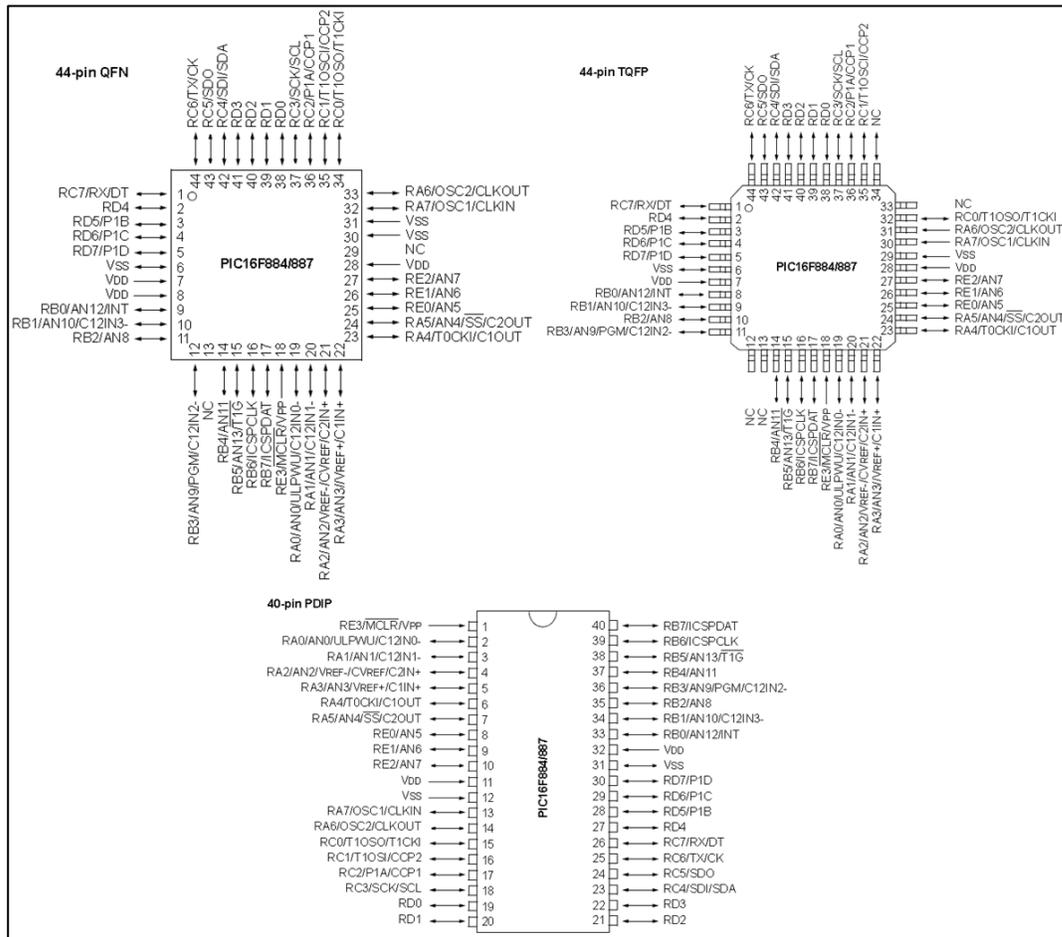


Figura 2.9 Tipos de Encapsulados de Microcontroladores PIC.

Fuente: Hoja de Especificaciones. Microchip Inc. [13]



2.3.3 Microcontrolador PIC16F887

2.3.3.1 Características

El microcontrolador PIC16F887 de Microchip Inc. pertenece a la gama media de microcontroladores; es un dispositivo que posee módulos dedicados a ayudar al microprocesador o CPU del mismo a cumplir con su tarea. En la siguiente tabla de pueden observar las características más relevantes del dispositivo [13]:

Tabla 2.2 Características del Microcontrolador PIC16F887.

CARACTERÍSTICAS	16F887
Frecuencia máxima	20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos de Entrada/Salida	A, B, C, D, E
Número de pines	40
Interrupciones	14
Temporizadores (<i>Timers</i>)	3
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones paralelo	PSP
Líneas de entrada de CAD de 10 bits	8
Repertorio de instrucciones	35 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard
CPU	RISC
Canales PWM	2
Pila Hardware	8 niveles
Soporte para Programación en circuito, ICSP (<i>In-Circuit Serial Programming</i>)	Sí

Fuente: Hoja de Especificaciones. Microchip Inc. [13]

Estas características se conjugan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos y programa, y por lo tanto en la velocidad de ejecución.

2.3.3.2 Estructura Externa

A continuación en la figura 2.10, se muestra el diagrama de terminales, del encapsulado PDIP, del microcontrolador PIC16F887, el cual fue utilizado para el diseño del PLC desarrollado en este proyecto:

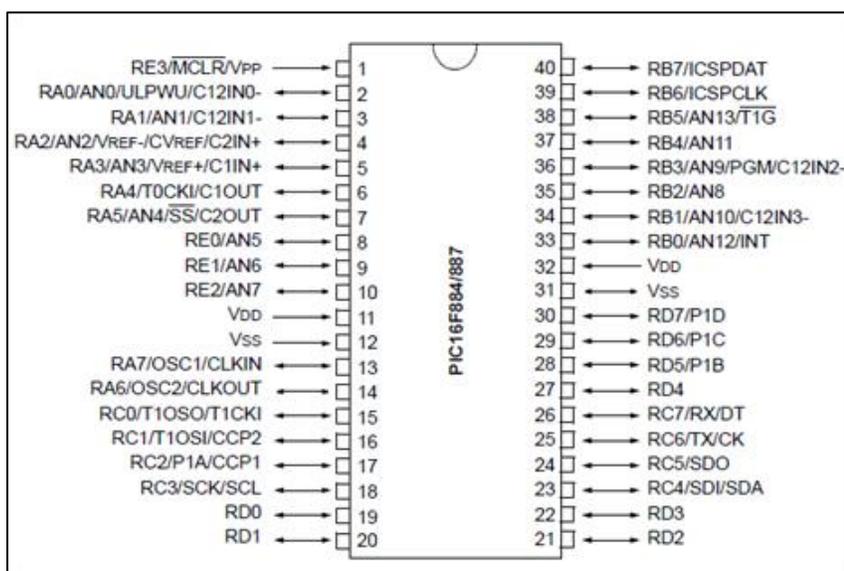


Figura 2.10 Diagrama de Terminales del Microcontrolador PIC16F887.

Fuente: Hoja de Especificaciones. Microchip Inc. [13]

2.4 Ambiente de Desarrollo Proton IDE

El ambiente de desarrollo *Proton IDE* combina el compilador *Proton Plus* con un entorno de programación y de simulación virtual. Es el resultado de la

colaboración de tres empresas británicas, Crownhill Associates, Mecanique y Labcenter, (ver figura 2.11).



Figura 2.11 Presentación de la ayuda sobre el Compilador Proton IDE.

Fuente: Ayuda del Compilador Proton IDE (Help About). [22]

Conveniente para todos los niveles de usuarios, desde principiantes hasta los más experimentados profesionales, el ambiente de desarrollo *Proton IDE* permite desarrollar programas BASIC en una interfaz de programación amigable e intuitiva, y luego compilar los mismos y ver el resultado en lenguaje ensamblador (Assembler) [22].

Está diseñado para funcionar sobre una plataforma PC con Microsoft Windows®. En la actualidad no brindan soporte para sistemas operativos como Linux o Mac OS. Para utilizar el ambiente de desarrollo *Proton IDE* es necesario un PC con al menos las siguientes características:

- **Procesador:** 233 MHz (500 MHz o superior recomendado).
- **Memoria RAM:** 64 MB (128 MB o superior recomendado).
- **Espacio en disco:** 100 MB.
- **Sistema operativo:** Windows 98 SE, ME, 2000, XP (Windows Vista y Windows XP 64-bit no son compatibles).

Este entorno de programación se utilizó fundamentalmente para la elaboración de programas de pruebas y calibración del PLC. Se recomienda consultar el “Manual de funciones básicas del *Proton IDE*” contenido en el anexo B, con el fin de dar conocer al *Proton IDE* como una herramienta de programación adicional.

2.5 Tecnología 1-Wire®

La tecnología 1-Wire® es un protocolo de comunicación serial desarrollado por la empresa Dallas Semiconductor, la cual, utiliza sólo un conductor de datos más su referencia para efectuar la comunicación y transmisión de energía entre un dispositivo maestro y múltiples esclavos [23].

Una red de dispositivos 1-Wire® está conformada por un maestro y uno o muchos esclavos que poseen un único terminal de datos de tipo “open drain” (Drenaje Abierto), al que se conecta una resistencia de “Pull-Up” alimentada por +5V nominales.

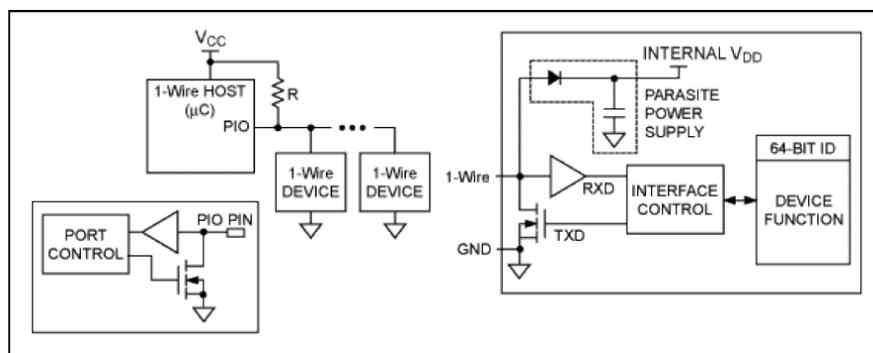


Figura 2.12 La configuración 1-Wire maestro/esclavo utiliza una única línea de datos más su referencia a tierra.

Fuente: 1-Wire Devices – Maxim. [23]

El dispositivo maestro inicia y controla la comunicación con uno o más dispositivos esclavos presentes en la red 1-Wire® (ver figura 2.12). Cada dispositivo



esclavo es único, inalterable y programado de fábrica, con 64-bit de ID (número de identificación) grabada en su memoria ROM, lo cual, facilita las acciones típicas de búsqueda en una red, el direccionamiento y transferencia de información entre múltiples dispositivos.

La mayoría de los dispositivos 1-Wire® no tiene un terminal para el suministro de energía, sino que la toman directamente de la red 1-Wire® (suministro parásito) [11,23].

2.5.1 Bus 1-Wire®

El bus 1-Wire®, permite realizar una comunicación serial asincrónica entre un dispositivo maestro y uno o varios dispositivos esclavos, utilizando un único terminal de Entrada/Salida, la comunicación se realiza a través de microcontroladores y/o interfaz de comunicación con un computador. Algunas características de este bus, según [11,23,24]:

- Utiliza niveles de alimentación CMOS/TTL con un rango de operación que abarca desde 2.8V hasta 5.25V.
- Tanto el maestro como los esclavos transmiten información de forma bidireccional, pero solo en una dirección a la vez. De esta manera la comunicación es “*Half-duplex*”.
- Toda la información es leída o escrita comenzando por el bit menos significativo (LSB, *Less significant Bit*).
- No se requiere el uso de un reloj, ya que cada componente 1-Wire® posee un oscilador interno que se sincroniza con el del maestro cada vez que en la línea de datos aparezca un flanco de bajada.



- La alimentación de los esclavos se puede hacer utilizando el voltaje propio del bus. Para ello, cada circuito esclavo posee un rectificador de media onda y un condensador. Durante los periodos en los cuales no se efectúa comunicación, la línea de datos se encuentra en estado alto debido a la resistencia de “Pull-Up”; en esa condición, el diodo entra en conducción y carga al condensador. Cuando el voltaje de la red cae por debajo de la tensión del condensador, el diodo se polariza en inverso evitando que el condensador se descargue. La carga queda almacenada en el condensador es la que finalmente alimentará al circuito esclavo.
- La red de dispositivos 1-Wire®, en general, tiene capacidad de manejar hasta 100 dispositivos esclavos distribuidos a lo largo de una distancia de 200 metros.
- Todas las tensiones mayores a 2.2V son consideradas un (1) lógico, mientras que un (0) lógico será cualquier voltaje menor a 0.8V.
- La transferencia de información es a 15.4 Kbps en modo “Standard” y hasta 125 Kbps en modo “Overdrive”.

2.5.2 Elementos que componen una Red 1-Wire®

La red 1-Wire® utiliza una topología maestro-esclavo, en la cual, existe un único dispositivo maestro y uno o más esclavos. Esta red se compone de tres elementos principales, un maestro de bus con un software de control, el cableado y los conectores asociados y los dispositivos 1-Wire®. Por otra parte, la red 1-Wire®, brinda un estricto control sobre la comunicación debido a que ningún nodo de interconexión está autorizado a transmitir información a menos que se lo indique el dispositivo maestro, además, no está permitida la comunicación directa entre los esclavos. La implementación del maestro 1-Wire® puede hacerse a través de diferentes medios, siendo los más utilizados un microcontrolador o un computador.



En cada caso, se requiere de determinados recursos de hardware. Para la implementación basada en un computador, se necesita un adaptador de puerto de comunicaciones, denominado “Host Adapter”. Este dispositivo permite establecer la conexión física entre el puerto del computador y la red 1-Wire®. Existen adaptadores de red para los puertos paralelo, serial RS-232C y USB, siendo estos dos últimos los más utilizados [23].

2.5.3 Protocolo de Comunicaciones 1-Wire®

Se puede describir al protocolo de comunicaciones 1-Wire® como una secuencia de transacciones de información, la cual, se desarrolla según los siguientes pasos: Inicialización, Comandos y funciones de ROM, Comandos y funciones de control y memoria, Transferencia de bytes o datos. A continuación se describe cada secuencia [23]:

- **Inicialización:** Las comunicaciones en el bus 1-Wire® comienzan con una secuencia de un pulso de Reset y Presencia. El pulso de Reset provee una forma conveniente de iniciar las comunicaciones, ya que, con él se sincronizan los dispositivos esclavos presentes en el bus. Un Reset es un pulso que genera el dispositivo maestro al colocar la línea de datos en estado lógico bajo por aproximadamente 480 μ s. Una vez liberado, el bus retorna al nivel alto y luego de un tiempo comprendido entre 15 y 60 μ s, los dispositivos esclavos transmitirán un pulso de presencia. Éste consiste en forzar la línea de datos a nivel bajo durante un tiempo entre 60 y 240 μ s. El dispositivo maestro, esperará por los pulsos de Presencia con el bus en estado alto, a través de la resistencia de “Pull-Up”, por un tiempo de al menos 240 μ s. Cuando en el bus se emplea el modo de velocidad “Standard”, los pulsos de Reset y Presencia tendrán características como las mostradas en la figura 2.13 [23].

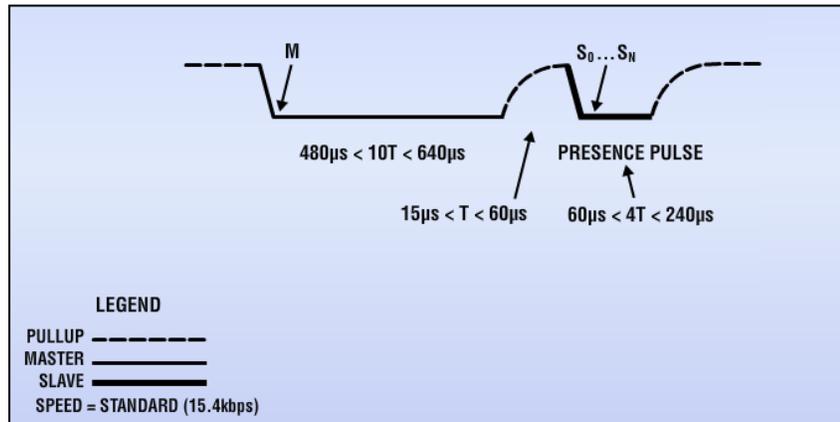


Figura 2.13 Representación gráfica de un pulso de Reset y Presencia.
 Fuente: 1-Wire Tutorial Presentation. [25]

- Comandos y funciones de ROM:** Una vez que el dispositivo maestro recibe el pulso de Presencia de los dispositivos esclavos, puede enviar un comando de ROM. Los comandos de ROM son comunes a los dispositivos 1-Wire® y se relacionan con la búsqueda, lectura y utilización de la dirección de 64 bits que identifica a los dispositivos 1-Wire® esclavos. Seguidamente se muestran los comandos de ROM comúnmente utilizados con los dispositivos 1-Wire®, (ver tabla 2.3) [23].

Tabla 2.3 Funciones de ROM del bus 1-Wire®

Comando	Valor	Descripción
Read ROM	\$33	Lee el número de identificación (ID) de 64 bits del dispositivo. Puede ser utilizado solamente cuando existe un sólo dispositivo esclavo presente en la red 1-Wire®.
Match ROM	\$55	Este comando, seguido de una identificación de 64 bits, permite seleccionar a un dispositivo esclavo en particular.
Skip ROM	\$CC	Direcciona a un dispositivo sin necesidad de conocer la identificación, puede ser utilizado solamente cuando existe un sólo esclavo conectado en la red 1-Wire®.
Search ROM	\$F0	Lee los 64 bits de identificación de los dispositivos esclavos conectados en la red. Se utiliza un proceso de eliminación para distinguir a cada dispositivo conectado.

Fuente: 1-Wire Tutorial Presentation. [25]

- **Comandos y funciones de control y memoria:** Son funciones propias de cada dispositivo 1-Wire®. Incluyen comandos para leer/escribir en localidades de memoria, leer memorias *scratchpad*, controlar el inicio de la conversión de un convertidor analógico digital (ADC), iniciar la medición de temperatura o manipular el estado de un bit de salida, entre otros. Cada dispositivo define un conjunto de comandos propios a su funcionalidad [23].
- **Transferencia de datos:** La lectura y escritura de datos en el bus 1-Wire® se hacen por medio de “Slots”, la generación de éstos es responsabilidad del dispositivo maestro.

Cuando el maestro lee información del bus, debe forzar la línea de datos a un estado bajo durante al menos $4 \mu\text{s}$ y esperar adicionalmente $15 \mu\text{s}$ para leer el estado presente en la línea. El estado lógico de la línea en ese momento, estará determinado por el dispositivo esclavo. A continuación se muestra el proceso de lectura de un bit (Slot de lectura) y de escritura de un bit (Slot de escritura), (ver figura 2.14) [23].

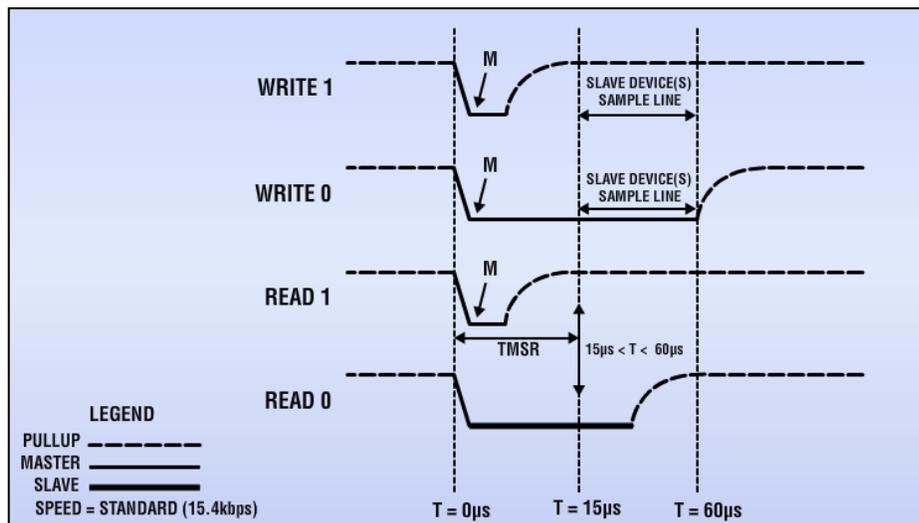


Figura 2.14 Transmisión y recepción de bits empleando el protocolo 1-Wire®.

Fuente: 1-Wire Tutorial Presentation. [25]



2.6 LDmicro

LDmicro es un programa con el cual se puede editar, compilar y simular la lógica de escalera (usada para la programación de PLC) para microcontroladores de 8 bits. Éste es capaz de generar códigos entendibles por microcontroladores Atmel AVR y Microchip PIC16, a partir del diagrama de escalera diseñado.

Este programa puede ser obtenido gratuitamente desde el portal web del autor (<http://www.cq.cx/ladder.pl>) teniendo acceso, también, al código fuente. Está amparado por el acuerdo de licencia GNU, lo que representa una ventaja para la realización de un PLC ya que disminuye costos por adquisición de licencias.

El programa se presenta en un entorno gráfico, no como una lista de instrucciones, lo que permite al usuario, desarrollar el programa de la misma manera como si estuviera utilizando un software especializado para programación de PLC.

Los programas diseñados con LDmicro se verán como diagramas de circuitos, con contactos, bobinas, temporizadores, entre otros. Esto es intuitivo para programadores que conocen la teoría de circuitos eléctricos. El programador sólo tiene que dedicarse a desarrollar la lógica del programa, el cual, cubrirá todas las necesidades de automatización, ya que, LDmicro, se encargará de convertir lo programado a lenguaje que el microcontrolador entienda. Es importante destacar que este programa no es muy amigable al usuario ya que no expresa de manera fácil la manipulación de variables y funciones. Este compila la lógica de escalera a código de PIC16 o AVR. Los microcontroladores soportados por el LDmicro son los siguientes [17]:



- PIC16F877
- PIC16F887
- PIC16F628
- PIC16F876
- PIC16F88
- PIC16F819
- ATmega128
- ATmega64
- ATmega162
- ATmega32
- ATmega16
- ATmega8

2.6.1 Instrucciones

A continuación se darán a conocer algunas de las instrucciones con las que trabaja el LDmicro [17]:

- **Contacto Normalmente Abierto**

Xname	Rname	Yname	
---]	[---	---]	[---

Representa un contacto de relé normalmente abierto, se cerrará cuando exista un estado lógico verdadero en el elemento que representa (terminal de entrada, terminal de salida, o relé interno).

- **Contacto Normalmente Cerrado**

Xname	Rname	Yname
---]/[---	---]/[---	---]/[---

Representa un contacto de relé normalmente cerrado, se abrirá cuando exista un estado lógico verdadero en el elemento que representa (terminal de entrada, terminal de salida, o relé interno).



- **Bobina Normal**

Rname	Yname
---()---	---()---

Representa la bobina de activación de un relé. Si la señal que va a la instrucción es verdadera, entonces el relé interno o el terminal de salida es verdadero. Esta instrucción debe estar en el extremo derecho del escalón que la contiene.

- **Bobina Negado**

Rname	Yname
---(/)---	---(/)---

Representa la bobina de activación de un relé. Si la señal que va a la instrucción es falsa, entonces el relé interno o el terminal de salida es verdadero. Esta instrucción debe estar en el extremo derecho del escalón que la contiene.

- **Retardo a la Conexión (TON: Turn-On Delay)**

Tname
– [TON 1.000 s] –

Esta función implementa un retardo a la conexión, el cual se manifiesta como un tiempo configurable que tarda la instrucción en cambiar su salida de estado. Para que la salida cambie de estado, la entrada debe pasar de falso a verdadero y permanecer en este estado durante el tiempo especificado en la instrucción. El temporizador se reinicia cada vez que la entrada cambia a falso. Es posible manipular desde otra parte la variable del contador, por ejemplo con una instrucción MOV.



- **Retardo a la Desconexión (TOF: Turn-Off Delay)**

Tname
– [TOF 1.000 s] –

Esta función implementa un retardo a la desconexión, el cual se manifiesta como un tiempo configurable que tarda la instrucción en cambiar su salida de estado. Para que su salida cambie de estado, la entrada debe pasar de verdadero a falso y permanecer en este estado durante el tiempo especificado en la instrucción. El temporizador se reinicia cada vez que la entrada cambia a verdadero. Es posible manipular desde otra parte la variable del contador, por ejemplo con una instrucción MOV.

- **Pulso Ascendente (OSR: One-Shot Rising)**

-- [OSR _ / ^] --

La salida de esta instrucción normalmente es falsa. Si la señal de entrada a la instrucción es verdadera durante su ciclo de escaneo y era falsa en el ciclo de escaneo anterior, entonces, la salida será verdadera y durará un ciclo de escaneo.

- **Pulso Descendente (OSF: One-Shot Falling)**

-- [OSF ^ \ _] --

La salida de esta instrucción normalmente es falsa. Si la señal de entrada a la instrucción es falsa durante su ciclo de escaneo y era verdadera en el ciclo de escaneo anterior, entonces, la salida será verdadera y durará un ciclo de escaneo.



- **Mover (MOV: Move)**

```
{destvar := }      {Tret := }  
- { 123     MOV }-  - { srcvar  MOV }-
```

Cuando la entrada a esta instrucción es verdadera, establece el valor de la variable destino igual al de la variable origen o fuente. Se puede asignar cualquier variable a la instrucción mover (MOV); esto incluye las variables de los temporizadores y contadores de estados, que pueden distinguirse llevando las iniciales “T” o “C”. Esta instrucción debe estar en el extremo derecho del escalón que la contiene.

- **Operaciones Aritméticas**

```
{ADD kay :=}      {SUB Ccnt :=}  
- { 'a' + 10 }-   - { Ccnt - 10 }-  
  
{MUL dest :=}    {DIV dv := }  
- { var * -990 }- - { dv / -10000 }-
```

Cuando la entrada a esta instrucción es verdadera, establece la variable destino igual a la expresión seleccionada. Los operandos pueden ser cualquier variable (incluso variables como el temporizador y contador). Estas instrucciones usan 16 bits. El resultado se evalúa en todos los ciclos mientras que la condición de entrada sea verdadera. Si la variable destino es uno de los operandos debe colocarse un OSR o un OSF para evitar cálculos recurrentes. Esta operación de división se implementa con números enteros y sin redondeos, es decir, trunca el resultado, por ejemplo $8 / 3 = 2$. Esta instrucción debe estar en el extremo derecho del escalón que la contiene.



- **Comparador**

$$\begin{array}{ccc}
 -[\text{var} ==]- & -[\text{var} >]- & -[1 \geq]- \\
 -[\text{var2}]- & -[1]- & -[\text{Ton}]- \\
 \\
 -[\text{var} \neq]- & -[-4 <]- & -[1 \leq]- \\
 -[\text{var2}]- & -[\text{vartwo}]- & -[\text{Cup}]-
 \end{array}$$

Si la entrada a esta instrucción es falsa entonces la salida es falsa. Si la entrada es verdadera, entonces la salida es verdadera si y sólo si la condición seleccionada es verdadera. Esta instrucción puede usarse para comparar (igual, mayor que, mayor o igual que, no igual, menor que, menor o igual que) una variable con otra.

- **Contador**

$$\begin{array}{cc}
 \text{Cname} & \text{Cname} \\
 -- [\text{CTU} \geq 5] -- & -- [\text{CTD} \geq 5] --
 \end{array}$$

Existen dos tipos de contadores: El contador ascendente (CTU: Count Up, cuenta ascendente) y el contador descendente (CTD: Count Down, cuenta descendente). En cualquier caso, la operación de conteo asociada se realiza en cada flanco ascendente de la condición de entrada de la instrucción. En el ejemplo mostrado la condición de salida del contador es verdadera si la variable del contador es mayor o igual que 5, de lo contrario es falso. La condición de salida del escalón puede ser verdadera aun cuando la condición de entrada es falsa; sólo depende de la variable del contador. Se pueden tener instrucciones CTU y CTD con el mismo nombre para incrementar y decrementar al mismo contador.

- **Contador Cíclico**

$$\begin{array}{c}
 \text{Cname} \\
 -- \{ \text{CTC } 0:7 \} --
 \end{array}$$

Un contador cíclico trabaja como un contador ascendente normal, sólo que después de alcanzar su límite, reinicia su cuenta a 0. Por ejemplo, el contador mostrado cuenta 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2,.... Puede ser usado



como un secuenciador. El contador cíclico cuenta en el flanco ascendente de la condición de entrada del escalón. Esta instrucción debe estar en el extremo derecho del escalón que la contiene.

- **Lectura del Conversor A/D**

```
Aname  
-- {READ ADC} --
```

LDmicro puede generar el código para usar el conversor A/D incluido en el microcontrolador. Si la condición de entrada a esta instrucción es verdadera, entonces se adquiere una simple muestra del conversor A/D y se almacena en la variable "Aname". Esta variable posteriormente puede manipularse con operaciones generales (menor que, mayor que, aritmética, entre otras).

Se debe asignar un terminal de entrada analógica a la variable "Aname" del mismo modo que asignaría un terminal a una entrada o salida digital, haciendo doble clic en la lista al final de la pantalla (ver figura 2.15). Si la condición de la entrada a este escalón es falsa la variable "Aname" no se altera.

0 voltios de entrada corresponde a 0, y una entrada igual a Vdd (voltaje de suministro) corresponde a 1023. Se pueden usar operaciones aritméticas para adecuar la lectura a unidades más convenientes, pero se debe tener en cuenta que se estarán usando enteros matemáticos. En general no todos los terminales estarán disponibles para ser usados con el conversor A/D. El software no le permitirá asignar los terminales que no sean A/D a una entrada analógica. Esta instrucción debe estar en el extremo derecho del escalón que la contiene.

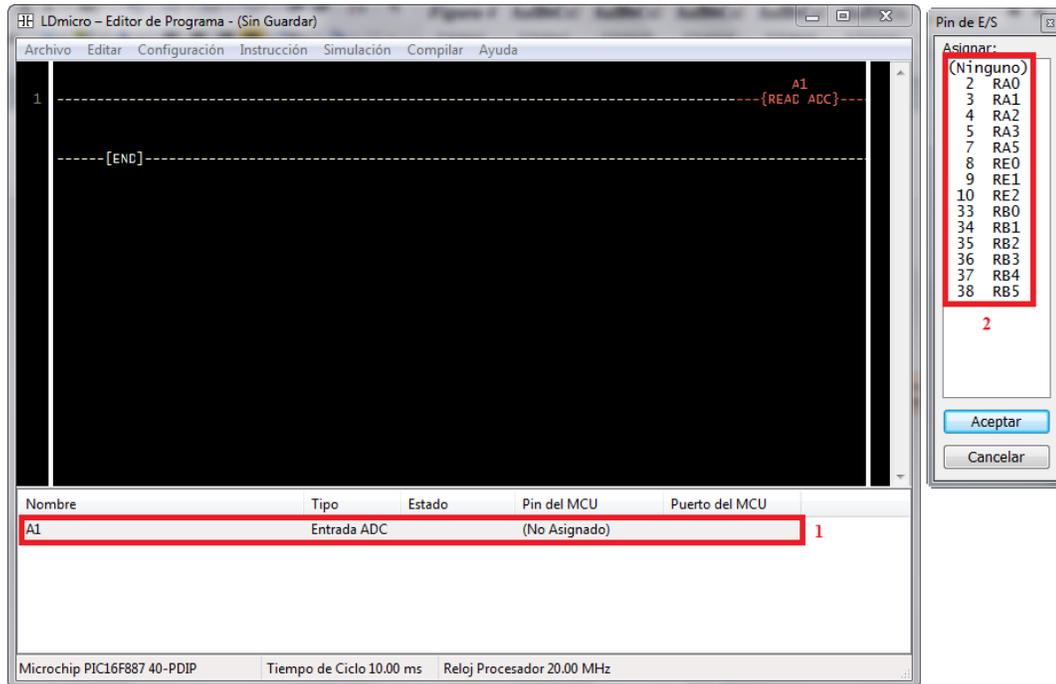


Figura 2.15 Configuración de la Instrucción Conversor A/D del LDmicro.

Fuente: Programa LDmicro. [17]

- 1- Lista de selección de objetos.
- 2- Lista de selección de posibles terminales de entradas analógicas del microcontrolador.

- **Generar salida de PWM.**

```
duty_cycle  
- {PWM 32.8 kHz} -
```

LDmicro puede generar el código para usar el módulo de PWM interno del microcontrolador. Si la condición de entrada a esta instrucción es verdadera, entonces el ciclo útil del módulo de PWM se establece en el valor de la variable "duty_cycle".

El ciclo útil debe ser un número entre 0 y 100; 0 siempre corresponde a bajo, y 100 siempre corresponde a alto.



Puede especificar la frecuencia (en Hz) de la PWM, y tal vez no se alcance ese valor exactamente, ya que depende de cómo el microcontrolador divida la frecuencia del reloj. LDmicro escogerá la frecuencia más cercana posible para alcanzar el valor deseado; si el error es muy grande se le avisará de forma visual en la pantalla del computador.

Esta instrucción debe estar en el extremo derecho del escalón que la contiene. El tiempo de ejecución de la lógica de escalera consume un Temporizador (Timer) del microcontrolador para medir el tiempo del ciclo. Esto significa que la PWM sólo está disponible en los microcontroladores con al menos dos temporizadores. La PWM utiliza el terminal CCP2 (no el CCP1).

2.6.2 La Simulación

Para utilizar el modo de simulación, seleccione en el menú *Simulación* (ver figura 2.16). El programa se verá diferente en el modo de simulación. No se muestra ningún cursor. Las instrucciones activas se muestran en rojo; las instrucciones inactivas se muestran en gris (ver figura 2.17). La pantalla del programa se actualizará en tiempo real cuando los estados del programa cambien.

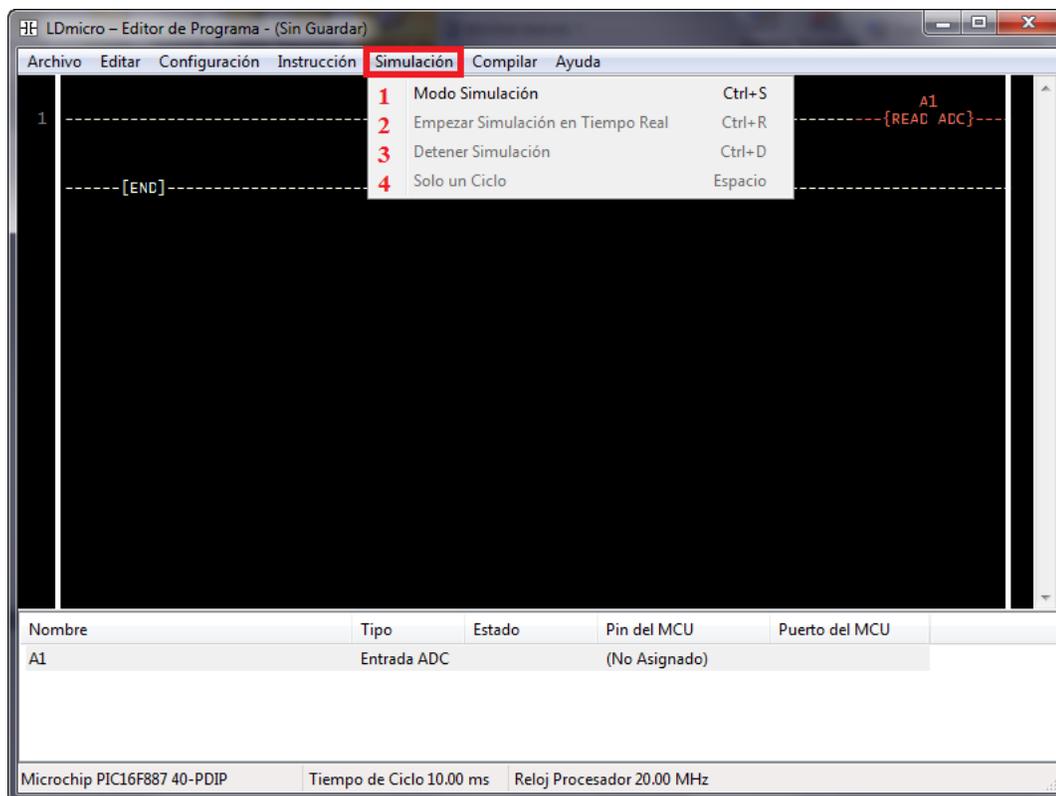


Figura 2.16 Menú para Simulaciones con el LDmicro.

Fuente: Programa LDmicro. [17]

- 1- Modo Simulación, acceso rápido <CTRL + S>.
- 2- Modo para ciclos continuos en tiempo real, acceso rápido <CTRL + R>.
- 3- Detener la Simulación, acceso rápido <CTRL + D>.
- 4- Para realizar un ciclo del PLC, acceso rápido <BARRA ESPACIADORA>.

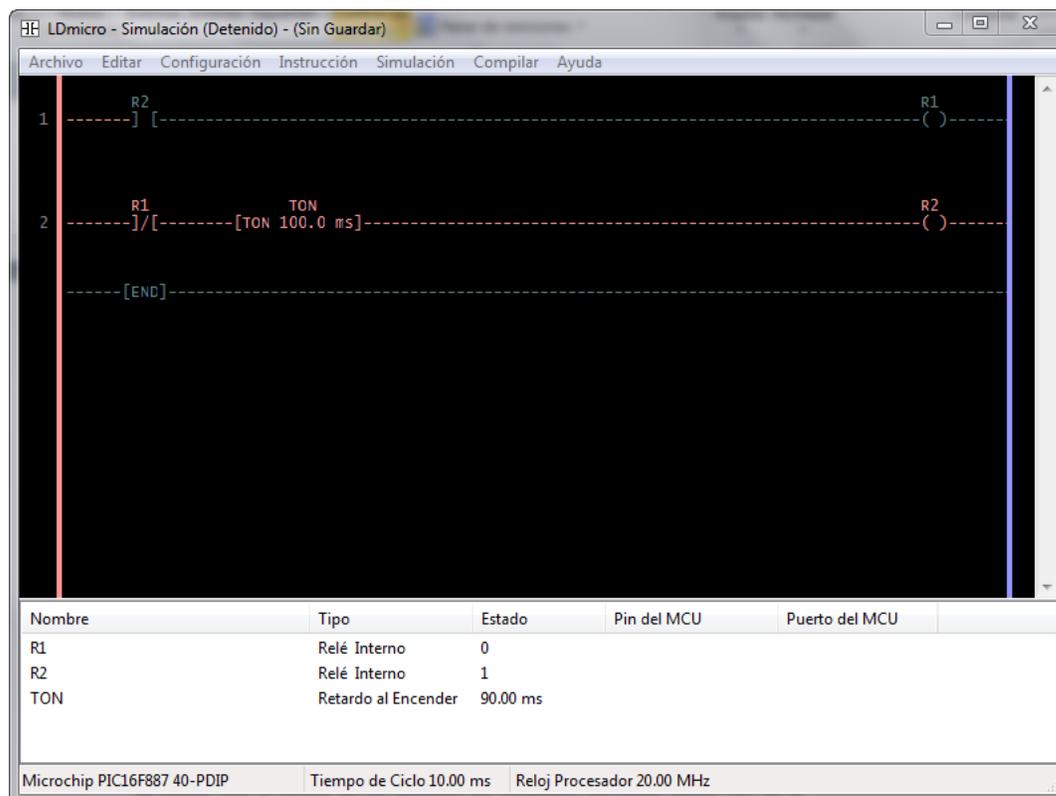


Figura 2.17 Modo de Simulación del LDmicro.

Fuente: Programa LDmicro. [17]

Se puede fijar el estado de las entradas del programa haciendo doble clic en la lista al final de la pantalla, o haciendo doble clic en la instrucción de los contactos “Xname” en el programa. Si se cambia el estado de un terminal de entrada entonces los cambios no se reflejarán hasta que los ciclos del PLC en el programa se visualicen. Esto pasará automáticamente si se está ejecutando una simulación en tiempo real, o cuando se presione la barra espaciadora [17].

2.6.3 Compilando a Código Nativo

Finalmente, el objetivo del programa LDMicro es generar un archivo .hex que se pueda usar para programar el microcontrolador. Para hacer la compilación del programa primero se debe asignar un terminal de entrada/salida a cada objeto

“Xname” o “Yname”. Esto se realiza haciendo doble clic en el nombre del objeto en la lista al final de la pantalla (ver figura 2.18). Saldrá una ventana de diálogo en dónde puede escoger de una lista un terminal que no se encuentre asignado.



Figura 2.18 Configuración de los Terminales Entrada/Salida del LDmicro.

Fuente: Programa LDmicro. [17]

- 1- Lista de selección de objetos.
- 2- Lista de selección de posibles terminales de entradas/salidas digitales del microcontrolador.

Ahora se puede generar el código del programa. Se selecciona *Compilar* → *Compilar*, o *Compilar* → *Compilar Como...*, si previamente se ha compilado el programa, se puede especificar un nombre de archivo diferente para así no sobrescribir el archivo generado en la anterior compilación (ver figura 2.19), si la compilación es exitosa se muestra al usuario una pantalla indicando los resultados (ver figura 2.20).

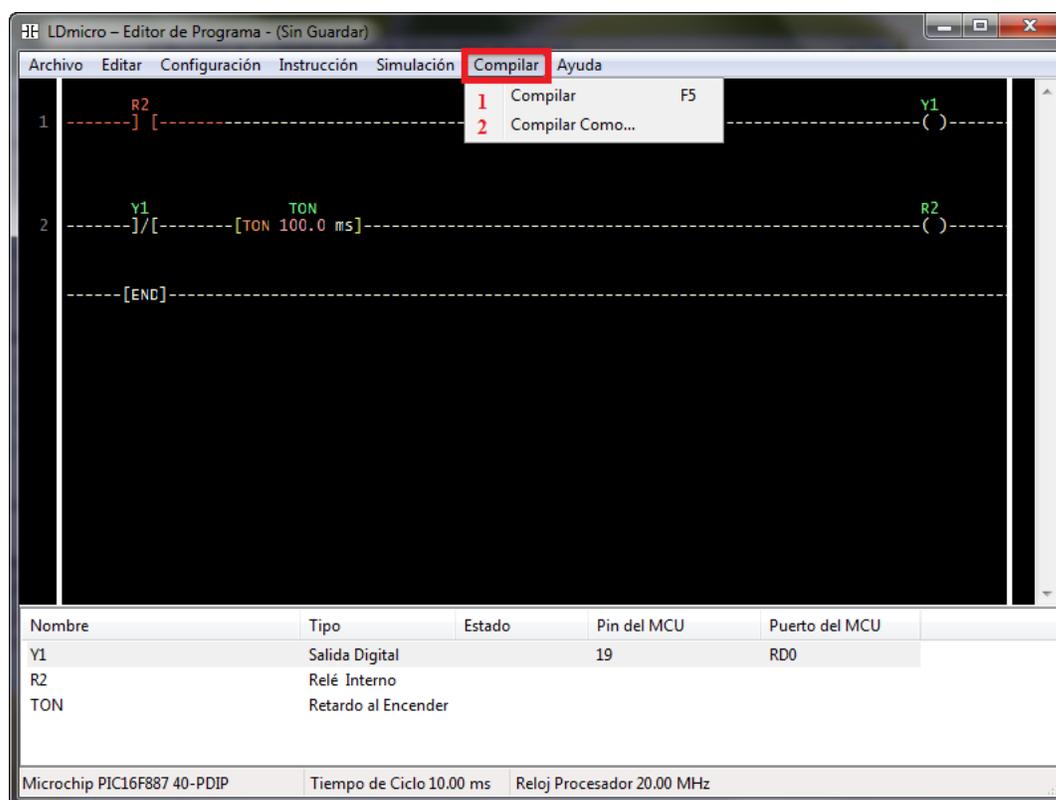


Figura 2.19 Menú de Compilación del LDmicro.

Fuente: Programa LDmicro. [17]

- 1- Opción Compilar, acceso rápido <F5>.
- 2- Opción Compilar Como.

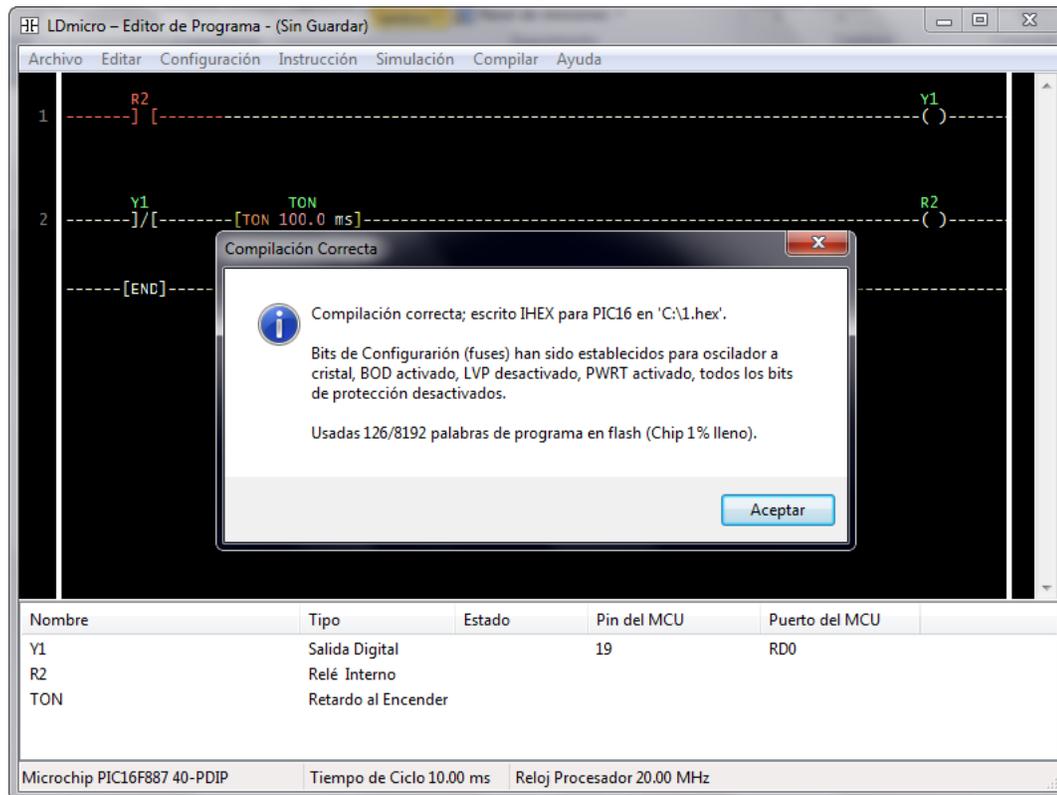


Figura 2.20 Ventana Resultado de la Compilación del LDmicro.

Fuente: Programa LDmicro. [17]

Se puede usar cualquier software y hardware que se tenga para cargar el archivo .hex en el microcontrolador [17].

Todas estas instrucciones pueden ser consultadas en el “Manual de funciones básicas del LDMicro” contenidas en el anexo B.



CAPÍTULO III

MARCO METODOLÓGICO

3.1 TIPO DE INVESTIGACIÓN

Considerando las características del presente trabajo, éste se encuentra clasificado dentro de los diferentes tipos de investigación como proyecto factible. El cual, consiste en “la investigación, elaboración y desarrollo de una propuesta de modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones, instituciones o grupos sociales; puede referirse a la formulación y desarrollo de políticas, programas, tecnologías, métodos o procesos” [26].

3.2 PROCEDIMIENTO METODOLÓGICO

En función al cumplimiento de los objetivos propuestos, el trabajo se desarrolló en seis etapas, las cuales se describen a continuación.

Etapa I: Definición de la estructura y funcionalidad del prototipo a desarrollar atendiendo a las características existentes en diferentes PLC de gama baja. Esta etapa comprende la realización de las siguientes actividades:

- Búsqueda de manuales técnicos de PLC de diferentes marcas y tipos.
- Selección de los manuales técnicos de PLC de gama baja.
- Estudio de las características de los PLC de gama baja, en cuanto a: cantidad de Entradas/Salidas digitales y analógicas, protocolos de comunicación, voltaje de alimentación y memoria de programa.



Etapa II: Diseño de los diagramas esquemáticos y de circuito impreso requeridos por el PLC para las interfaces de Entrada/Salida digitales y analógicas, protocolo de comunicación y fuente de alimentación. Durante esta etapa se realizarán las actividades siguientes:

- Estudio del microcontrolador PIC16F887.
- Selección de relé industrial con aislamiento óptico o galvánico y con voltaje de operación de 5V para Entrada digital.
- Selección de relé industrial con aislamiento óptico o galvánico y con voltaje de operación de 5V para Salida digital.
- Selección de relé industrial con aislamiento óptico o galvánico y con voltaje de salida de 0V a 5V para Entrada analógica.
- Selección de interfaz para Salida analógica de 4mA a 20mA aislada ópticamente.
- Selección de un convertidor DC-DC de 24V a 5V con aislamiento para la alimentación del PLC.
- Selección de dispositivo para comunicación 1-Wire®.
- Estudio del programa *Express PCB*.
- Elaboración de los esquemas electrónicos.

Etapa III: Construcción de los diseños preliminares, para esta etapa se realizaron las siguientes actividades:

- Montaje y conexión de microcontrolador, relés de Entrada/Salida digitales y analógicos, interfaces de salida de corriente, optoacopladores, leds y pantalla LCD en la placa de pruebas (*protoboard*).
- Estudio de los entornos de programación Proton IDE y LDmicro.
- Programación del microcontrolador.



- Pruebas de comunicación entre las diferentes interfaces de Entrada/Salida y el microcontrolador.

Etapa IV: Construcción de circuitos prototipos, los cuales se realizaron mediante las siguientes actividades:

- Construcción de circuitos impresos.
- Adaptación de los circuitos en las cajas de protección seleccionadas.
- Pruebas de operatividad y levantamiento de curva de calibración del PLC. Esta actividad se realizó en el laboratorio de Instrumentación de Procesos por medio del controlador HoneyWell UDC1000 y un miliamperímetro como elemento auxiliar para verificar la salida en corriente del PLC. El procedimiento fue proveer al equipo (UDC1000) con una entrada de corriente igual al valor mínimo de su rango, para la obtención del cero, luego se incrementó el valor de corriente hasta alcanzar el punto máximo del rango establecido por el instrumento (UDC1000) para finalmente con los datos obtenidos levantar la curva de calibración del PLC.

Etapa V: Adaptación del compilador de lenguaje escalera distribuido bajo licencia GNU LDmicro. Durante esta etapa se realizaron las siguientes actividades:

- Estudio del entorno de programación Visual Studio 2008 y del lenguaje de programación C++.
- Adaptación del código del programa LDmicro, para un funcionamiento acorde con el prototipo de PLC a construir.
- Prueba del correcto funcionamiento de la nueva versión del programa LDmicro con el prototipo de PLC.



Etapas VI: Elaboración de un conjunto de tres prácticas, las cuales se realizaron mediante las siguientes actividades:

- Recopilación de material bibliográfico para usar como apoyo teórico en las prácticas a realizar.
- Proponer las actividades a realizar en la práctica.
- Presentar la solución de la práctica tanto en planos eléctricos y esquemas de montaje, como en programación utilizando como elemento de control el PLC a construir.



CAPÍTULO IV

RESULTADOS

En este capítulo se describen el equipo y los módulos accesorios construidos en el desarrollo del presente trabajo especial de grado, sus características de operación y el software diseñado para el monitoreo del mismo.

4.1 Definición de la estructura y funcionalidad del prototipo a desarrollar atendiendo a las características existentes en diferentes PLC de Gama Baja

Como resultado del estudio realizado de diferentes tipos, marcas y modelos de PLC de gama baja, se elaboró una tabla descriptiva de cada uno de ellos (ver anexo A), atendiendo a las siguientes características de un PLC:

- Cantidad de entradas y salidas digitales y analógicas.
- Tipo de comunicación.
- Tipo de fuente de alimentación.
- Memoria de programa.
- Tamaño del dispositivo.
- Aislamiento en las entradas y salidas y en la fuente alimentación.

Teniendo en cuenta las características anteriores, se realizaron; un módulo principal y tres módulos de expansión, los cuales tienen la siguiente distribución:

Módulo Principal:

- 8 entradas/salidas digitales.



- 2 entradas analógicas aisladas.
- 2 salidas analógicas de 4 – 20 mA aisladas.
- Fuente de Alimentación de 24V aislada.
- Memoria de programa de 8KB, está definida por el microcontrolador PIC seleccionado.
- Pantalla LCD de 16x2 caracteres.
- Hardware de soporte para la comunicación utilizando el protocolo 1-Wire®.
- Aislamiento en todas las entradas y salidas.

Módulo de expansión digital:

- 6 entradas/salidas digitales.
- Aislamiento en todas las entradas y salidas.

Módulo de expansión analógico:

- 4 entradas analógicas.
- Aislamiento en todas las entradas.

Módulo de teclado:

- 5 entradas y 1 salida digital.

A continuación en la figura 4.1, se encuentran los diagramas de conexión entre el módulo principal PLC887 y los distintos módulos de expansión, de esta manera se presentan dos tipos de conexiones, en el Diagrama 1 se encuentra el PLC887, el módulo de expansión analógico y el teclado. En el Diagrama 2 se encuentra el PLC887, el módulo de expansión digital y el analógico; estos arreglos se realizan de esta manera debido a que los terminales del microcontrolador usados para el módulo de expansión digital son los mismos terminales usados por el teclado, la utilización de cada módulo de expansión va a depender del programa que se le coloque al PLC887.

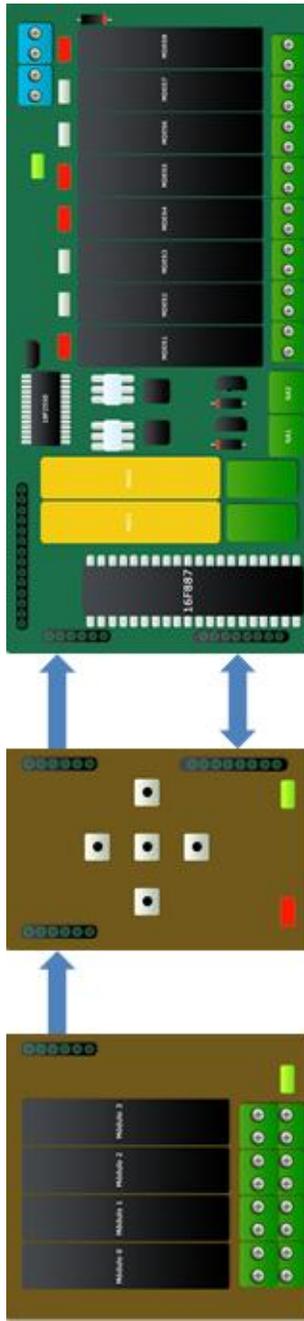


Diagrama 1.

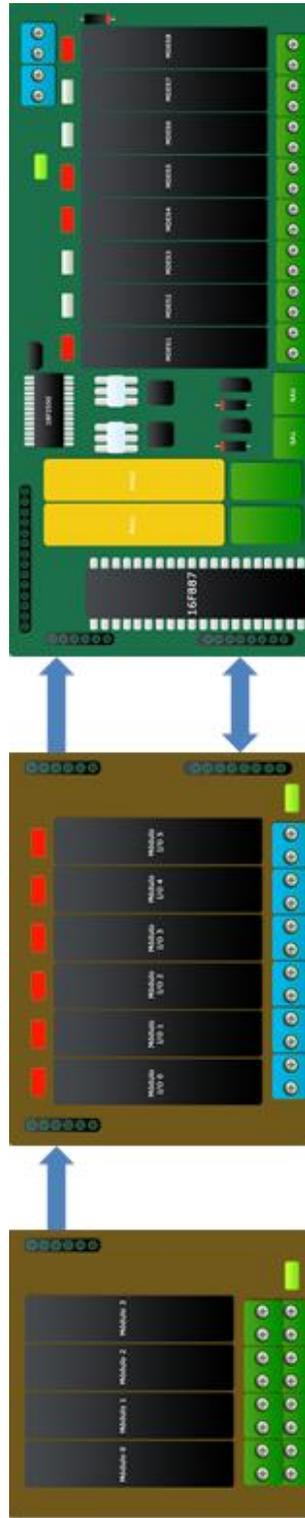


Diagrama 2.

Figura 4.1 Diagramas de Conexiones entre el PLC887 y Módulos de Expansión.



4.2 Diseño de los diagramas esquemáticos y de circuito impreso requeridos por el PLC para las interfaces de Entrada/Salida Digitales y Analógicas, Protocolo de Comunicación y Fuente de Alimentación

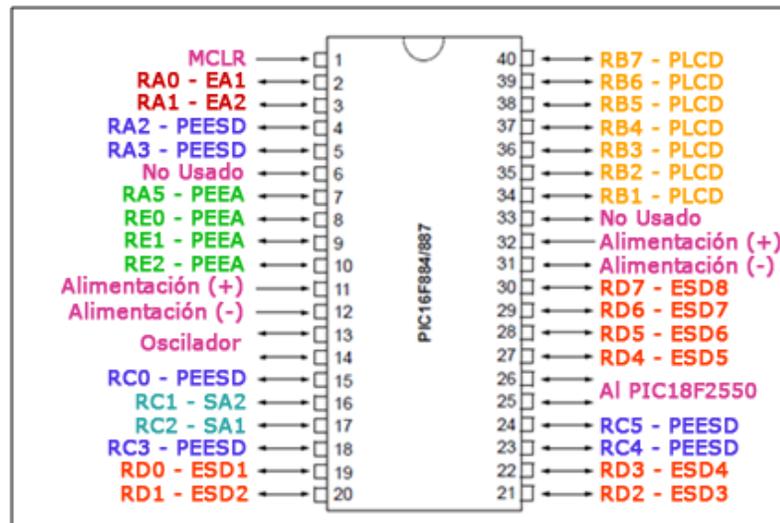
Para realizar el diseño del PLC, previamente se estudió al microcontrolador PIC16F887 y se tomaron en cuenta las siguientes características:

- 35 terminales de Entrada/Salida digital.
- 14 terminales de Entrada analógica.
- 8KB de memoria de programa.
- 2 Salidas de PWM.
- Alimentación de 2,0V a 5,5V.

De acuerdo a las características anteriores, se realizó la siguiente distribución:

- 8 Entradas/Salidas digitales, asociadas todas al puerto D, configurables desde el programa principal del microcontrolador PIC.
- 2 Entradas analógicas, asociadas a los convertidores A/D en los terminales RA0 y RA1.
- 2 Salidas analógicas, asociadas a los terminales RC1 y RC2, cada una de ellas manejadas por el hardware de PWM del microcontrolador PIC.
- 8 Salidas digitales, para manejar la pantalla LCD de 16x2 caracteres, la misma es controlada por el puerto B del microcontrolador PIC.
- 6 Entradas/Salidas digitales, asociadas a los terminales RC0, RA2, RA3, RC3, RC4 y RC5, configurables desde el programa principal del microcontrolador PIC, para el módulo de expansión digital o el módulo del teclado.

- 4 Entradas analógicas, asociadas a los terminales RA5, RE0, RE1 y RE2, configurables desde el programa principal del microcontrolador PIC, para el módulo de expansión analógico.



Leyenda:

- EA - Entrada Analógica
- PEESD - Puerto de Expansión de Entrada/Salida Digital
- PEEA - Puerto de Expansión de Entrada Analógica
- SA - Salida Analógica
- ESD - Entrada/Salida Digital
- PLCD - Pantalla LCD

Figura 4.2 Diagrama de Terminales del Microcontrolador PIC16F887 asociados al PLC887.

4.2.1 Módulos Industriales de Entrada y Salida

Estos módulos permiten hacer la interfaz de potencia entre el PLC887 (y sus accesorios de expansión) y los elementos finales de indicación y/o control. Para la selección de estos módulos industriales, se tomó como referencia lo siguiente: voltaje de operación de 5V y aislamiento óptico o galvánico; los módulos que se adecuaron a nuestros requerimientos son los fabricados por la empresa DataForth®, además de que cuentan con una variada gama de modelos.

4.2.1.1 Entrada/ Salida Digital

Los módulos de entrada/salida digital se dividen en dos grupos, los módulos de entrada para tensiones AC y para tensiones DC (ver figura 4.3), y los módulos de salida para manejar cargas en AC y para el manejo de cargas en DC (ver figura 4.4). Las especificaciones técnicas de cada uno de los módulos marca DataForth®, se muestran en las tablas 4.1 y 4.2.

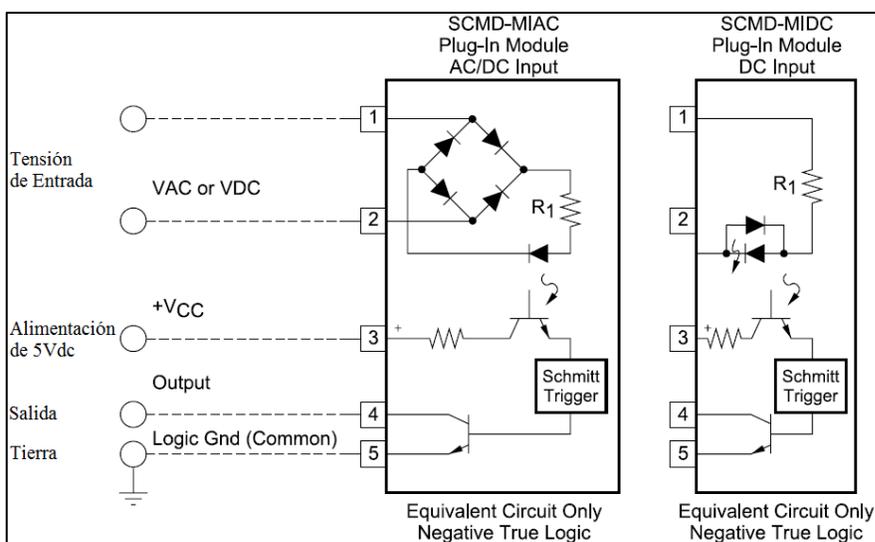


Figura 4.3 Esquema Electrónico del Módulo de Entrada Digital – DataForth®.

Fuente: www.dataforth.com - Hoja de especificaciones técnicas del módulo.

Tabla 4.1 Especificaciones de los Módulos de Entrada Digital – DataForth®.

PARÁMETRO	TÍPICO
Voltaje de Entrada AC	24V, 120V, 240V
Voltaje de Entrada DC	3.3 a 32V, 10 a 60V
Nivel de Aislamiento	4kV _{RMS}
Tipo Salida	Colector Abierto
Temperatura de Operación	-30 a +80 °C
Voltaje de Alimentación	5V _{DC}
Certificaciones	UL, CSA, CE

Fuente: www.dataforth.com - Hoja de especificaciones técnicas del módulo.

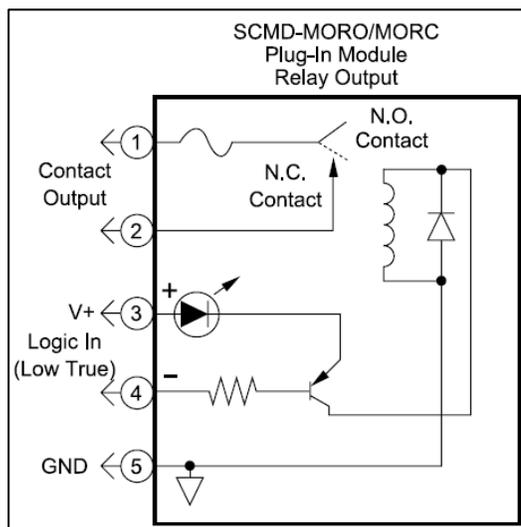


Figura 4.4 Esquema Electrónico del Módulo de Salida Digital – DataForth®.
Fuente: www.dataforth.com - Hoja de especificaciones técnicas del módulo.

Tabla 4.2 Especificaciones de los Módulos de Salida Digital – DataForth®.

PARÁMETRO	TÍPICO
Potencia Max. en la Salida	30W @ 0-110V _{DC}
	62.5VA @ 0-125V _{AC}
Nivel de Aislamiento	1000 V _{RMS}
Temperatura de Operación	-30 a +80 °C
Voltaje de Alimentación	5V _{DC}
Tiempo de Activación	2 ms
Tiempo de Desactivación	1 ms
Certificaciones	UL, CSA, CE

Fuente: www.dataforth.com - Hoja de especificaciones técnicas del módulo.

4.2.1.2 Entrada Analógica

Los módulos de entrada analógica permiten la conexión de sensores, cuya salida es variable (analógica), con el PLC887. Ofrecen un alto nivel de aislamiento (1500 V_{RMS}) y de linealidad ($\pm 0.02\%$ del span), lo cual hace que sea seguro y confiable para su uso. En esta investigación se utilizó el módulo analógico de entrada de corriente de (0 - 20) mA modelo 8B32 fabricado por la empresa DataForth®

(ver figura 4.5). Este módulo es alimentado con $5V_{DC}$ y su salida varía de $(0 - 5) V_{DC}$ según la corriente de entrada (ver tabla 4.3).

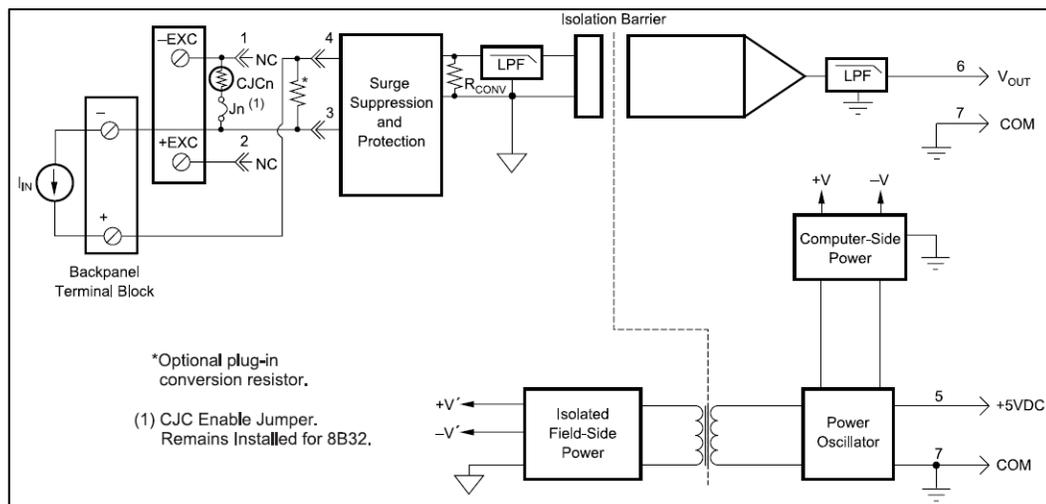


Figura 4.5 Esquema Electrónico del Módulo de Entrada Analógica – DataForth®.

Fuente: www.dataforth.com - Hoja de especificaciones técnicas del módulo.

Tabla 4.3 Especificaciones de los Módulos de Entrada Analógica – DataForth®.

PARÁMETRO	TÍPICO
Rango de Entrada	0 a 20 mA ó 4 a 20 mA
Resistencia de Entrada	<50 Ω
Precisión	$\pm 0,05\%$ Span
Linealidad	$\pm 0,02\%$ Span
Rango de Salida	0 a 5 V_{DC}
Tensión de Alimentación	5 $V_{DC} \pm 5\%$
Corriente de Alimentación	30mA
Temperatura de Operación	-40 a +85 $^{\circ}C$
Nivel de Aislamiento	1500 V_{RMS}
Certificaciones	UL (Pendiente), CE

Fuente: www.dataforth.com - Hoja de especificaciones técnicas del módulo.

4.2.2 Salida Analógica

La selección del XTR115UA (ver figura 4.6) como interfaz de salida de corriente se realizó considerando lo siguiente: es de bajo costo, su tamaño es reducido debido a su encapsulado de montaje superficial SOIC de 8 terminales, es manipulado mediante la PWM del microcontrolador y cuenta con un rango de salida estandarizado de 4 – 20 mA (ver tabla 4.4); este se encuentra aislado por un optoacoplador 4N35SR2M (ver figura 4.7) para la protección del PLC, ofreciendo un nivel de aislamiento de 5,3 kV_{RMS} (ver tabla 4.5).

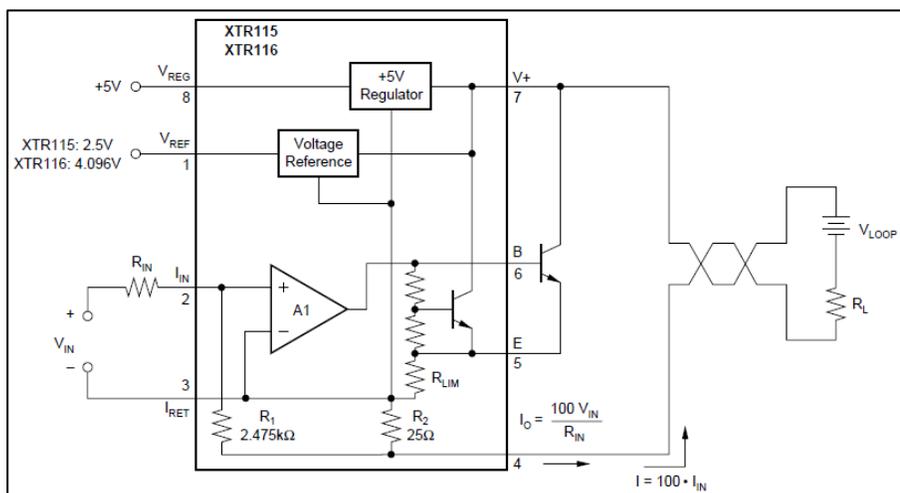


Figura 4.6 Esquema Electrónico del XTR115UA – Texas Instruments®.

Fuente: www.ti.com - Hoja de especificaciones técnicas.

Tabla 4.4 Especificaciones del XTR115UA – Texas Instruments®.

PARÁMETRO	TÍPICO
Voltaje Regulado para Elementos Externos	5 V _{DC}
Voltaje de Alimentación	24 V _{DC}
Precisión	±0,05% Span
Linealidad	±0,003% Span
Rango de Salida	4 a 20 mA _{DC}
Temperatura de Operación	-40 a +85 °C

Fuente: www.ti.com - Hoja de especificaciones técnicas.

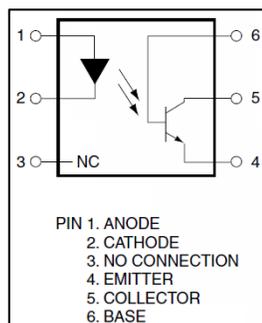


Figura 4.7 Esquema Electrónico del 4N35SR2M – Fairchild Semiconductor®.

Fuente: www.fairchildsemi.com - Hoja de especificaciones técnicas.

Tabla 4.5 Especificaciones del 4N35SR2M – Fairchild Semiconductor®.

PARÁMETRO	TÍPICO
Voltaje de Alimentación Ánodo – Cátodo	1,5 V _{DC}
Voltaje de Alimentación Colector – Emisor	30 V _{DC}
Nivel de Aislamiento	5,3 kV _{RMS}
Temperatura de Operación	-40 a +100 °C

Fuente: www.fairchildsemi.com - Hoja de especificaciones técnicas.

4.2.3 Fuente de Alimentación

Para la alimentación del PLC se seleccionó al convertidor DC-DC de 24V a 5V modelo DCP022405U de la empresa Texas Instruments (ver figura 4.8), de encapsulado de montaje superficial SO-28, por su capacidad de aislamiento, potencia de 2W y una eficiencia del 89% (ver tabla 4.6). Para el diseño del PLC se utilizaron dos de estos convertidores, debido a que tienen la capacidad de sincronizarse entre ellos mediante una configuración ya establecida (ver figura 4.9), ofreciendo así 2 veces su potencia de salida.

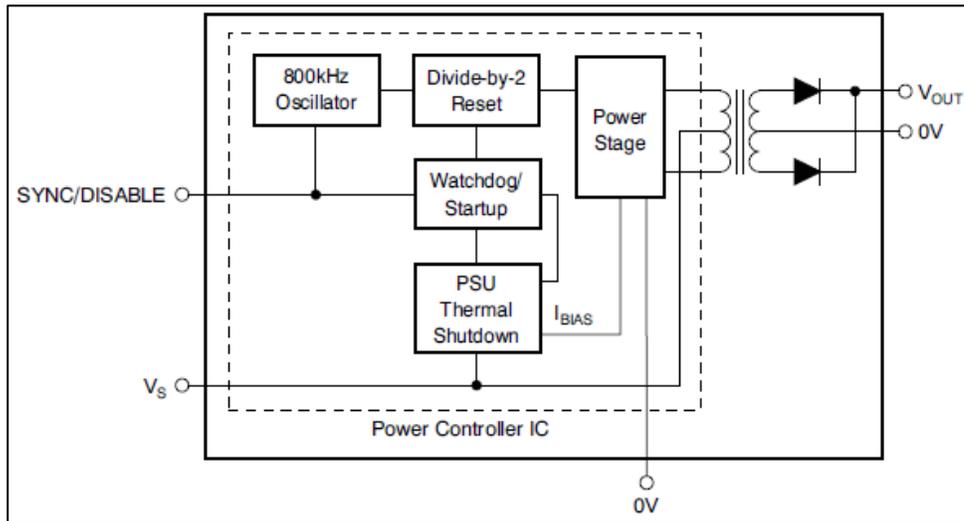


Figura 4.8 Esquema Electrónico del DCP022405U – Texas Instruments®.
Fuente: www.ti.com - Hoja de especificaciones técnicas.

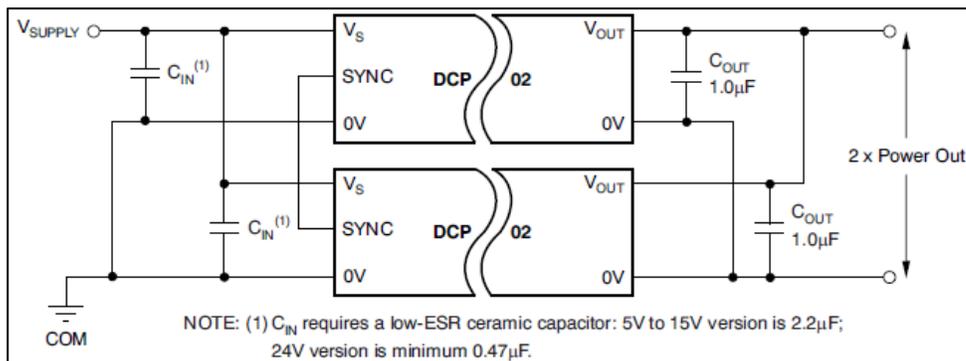


Figura 4.9 Esquema Electrónico de la Conexión en Paralelo del DCP022405U – Texas Instruments®.
Fuente: www.ti.com - Hoja de especificaciones técnicas.

Tabla 4.6 Especificaciones del 4N35SR2M – Texas Instruments®.

PARÁMETRO	TÍPICO
Voltaje de Alimentación	24 V _{DC}
Voltaje de Salida	5 V _{DC}
Potencia de Salida	2 W
Nivel de Aislamiento	1 kV _{RMS}
Temperatura de Operación	-60 a +125 °C
Eficiencia	89%

Fuente: www.ti.com - Hoja de especificaciones técnicas.

4.2.4 Interfaz de Hardware para Comunicación 1-Wire®

Para la comunicación 1-Wire®, se seleccionó al DS2409 este es un acoplador de MicroLAN, el cual es un componente esencial para construir y controlar redes 1-Wire® con varios niveles de ramificación (ver figura 4.10). A su vez se colocó un microcontrolador PIC18F2550 (ver figura 4.11 y ver tabla 4.7) para el monitoreo de los estados del PLC tanto de las entradas analógicas como de las entradas/salidas digitales, el cual se encarga de comunicarse con el DS2409 y posteriormente con el computador.

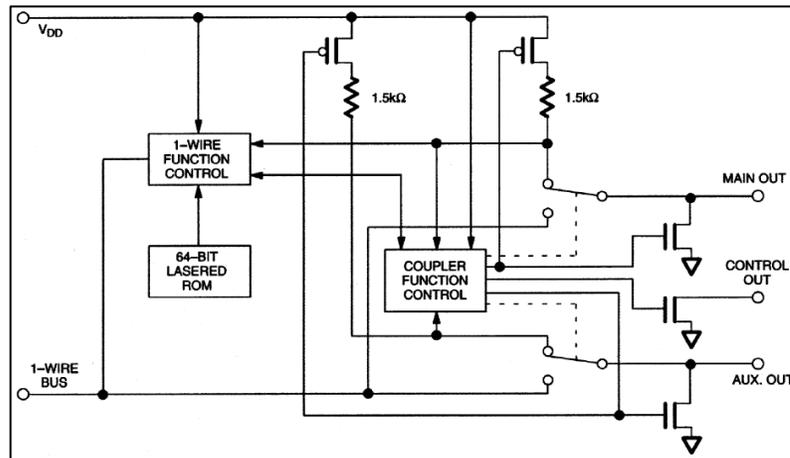


Figura 4.10 Esquema Electrónico del DS2409 – Dallas Semiconductor®.
Fuente: www.maxim-ic.com - Hoja de especificaciones técnicas.

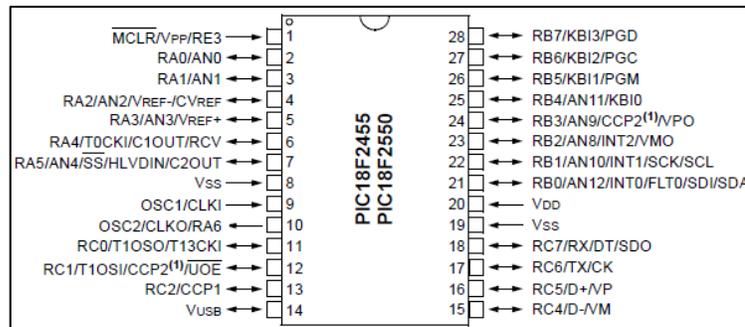


Figura 4.11 Diagrama de Terminales del Microcontrolador PIC18F2550.
Fuente: www.microchip.com - Hoja de especificaciones técnicas.



Tabla 4.7 Especificaciones del Microcontrolador Microchip® PIC18F2550.

CARACTERÍSTICAS	18F2550
Frecuencia Máxima	48MHz
Memoria de programa flash palabra de 14 bits	32KB
Posiciones RAM de datos	2048
Posiciones EEPROM de datos	256
Puertos de Entrada/Salida	A, B, C, (E)
Número de pines	28
Interrupciones	19
Temporizadores (<i>Timers</i>)	4
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicación USB	1
Líneas de entrada de CAD de 10 bits	10
Repertorio de instrucciones	75 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard
CPU	RISC
Canales PWM	2
Pila Hardware	31 niveles
Soporte para Programación en circuito, ICSP (<i>In-Circuit Serial Programming</i>)	Sí

Fuente: *www.microchip.com - Hoja de especificaciones técnicas.*

A continuación se muestran los diseños esquemáticos del módulo principal del PLC (ver figuras 4.12, 4.13, 4.14, 4.15 y 4.16), el cual fue llamado “PLC887”, de igual manera se muestran los diseños esquemáticos de los módulos de expansión digital, analógica y teclado (ver figuras 4.17, 4.18 y 4.19).

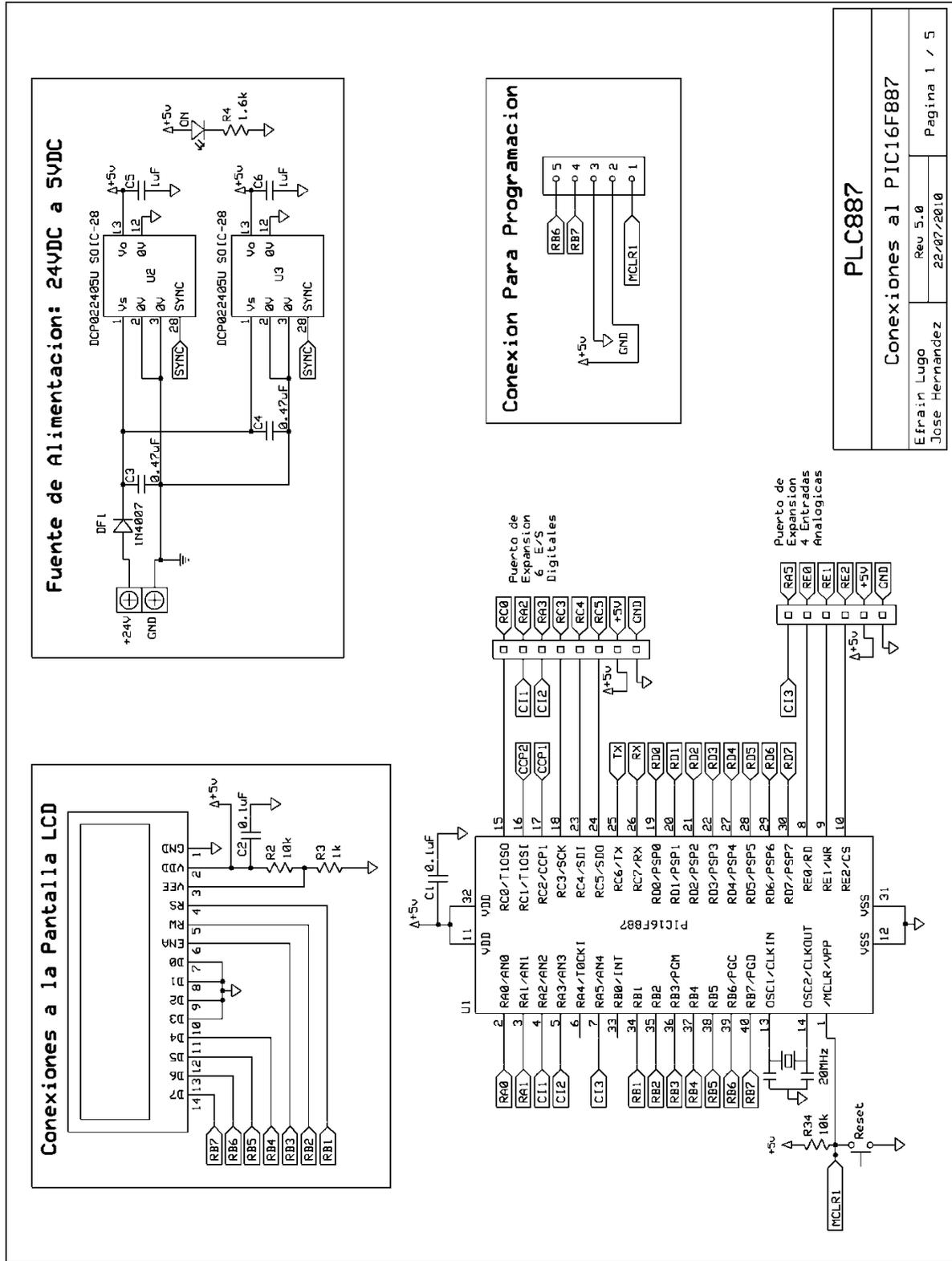


Figura 4.12 Esquemático Conexiones al PIC16F887.

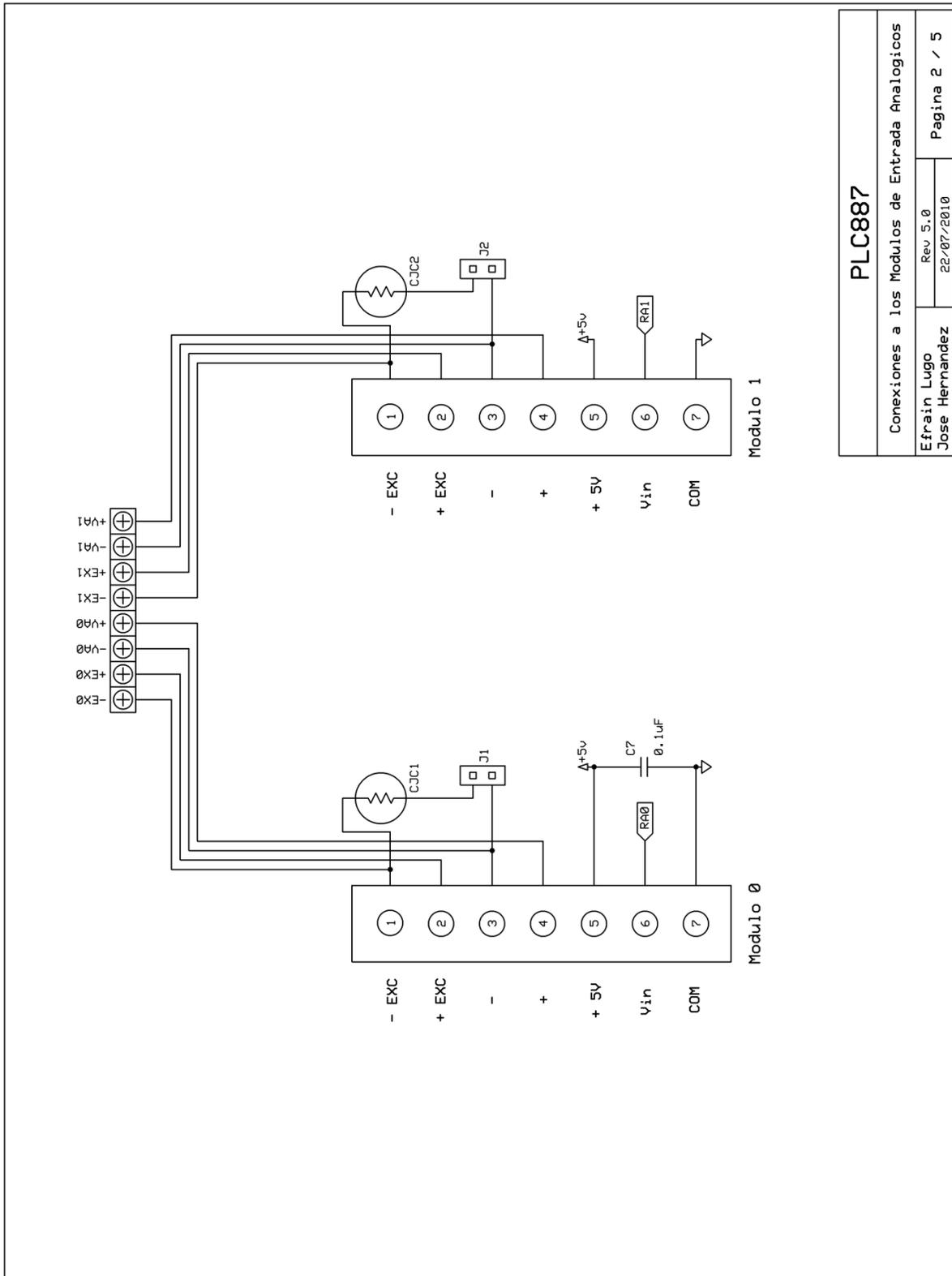
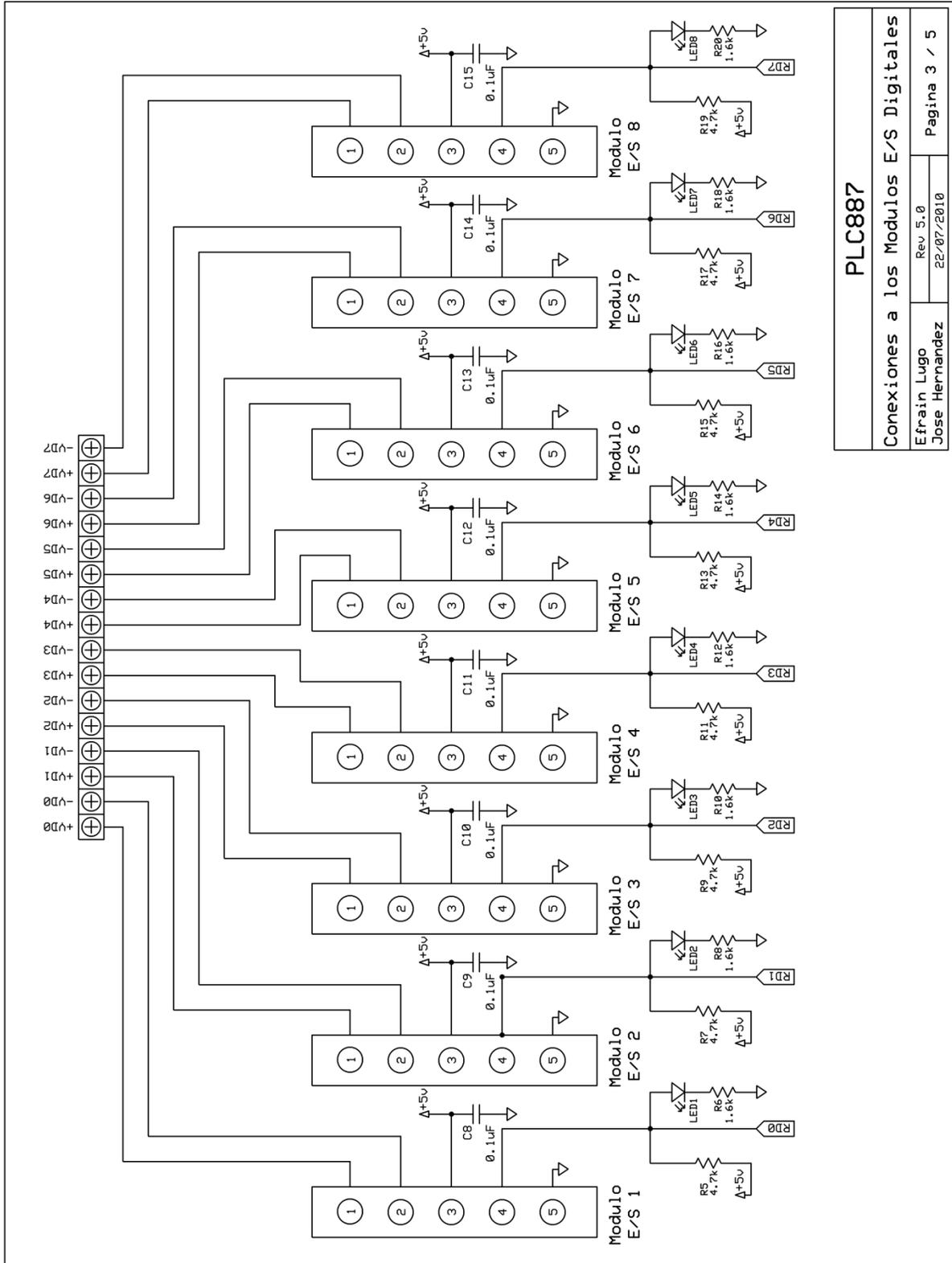
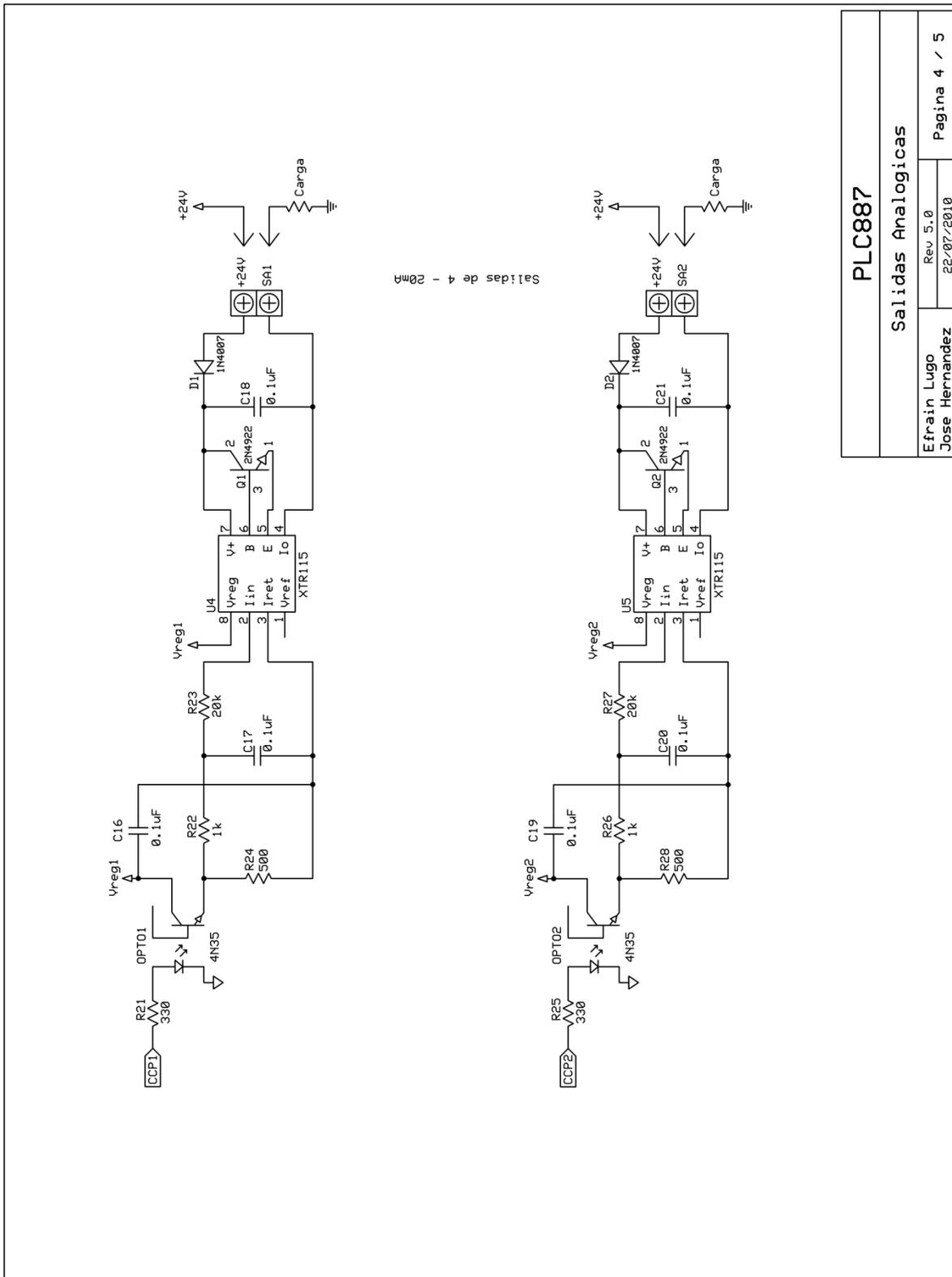


Figura 4.13 Esquemático de Conexiones a los Módulos de Entrada Analógica.



PLC887	
Conexiones a los Módulos E/S Digitales	
Efrain Lugo	Rev 5.0
Jose Hernandez	22-07/2010
Pagina 3 / 5	

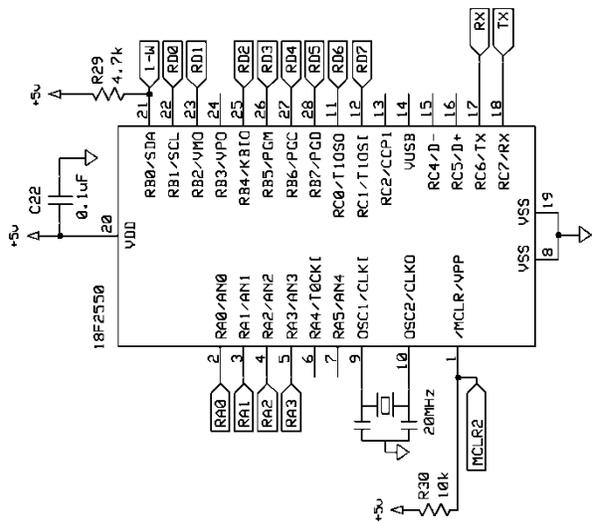
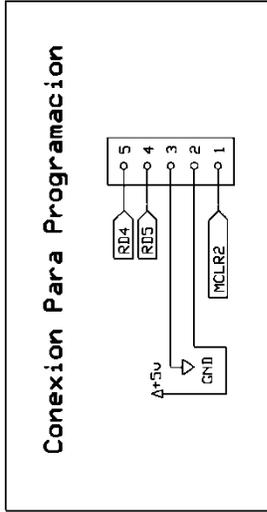
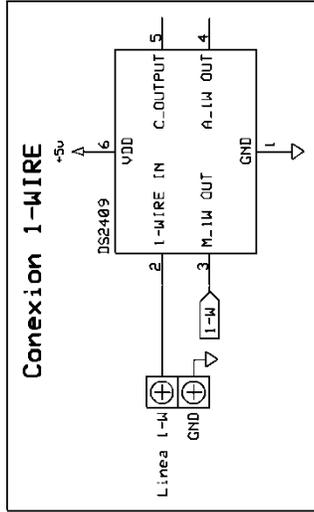
Figura 4.14 Esquemático de Conexiones a los Módulos de E/S Digital.



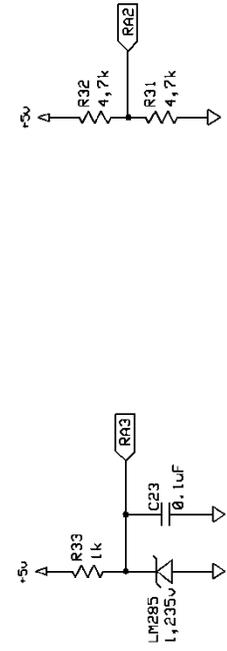
Salidas de 4 - 20mA

PLC887	
Salidas Analógicas	
Efraín Lugo	Rev 5.0
Jose Hernandez	22/07/2010
Pagina 4 / 5	

Figura 4.15 Esquemático Salidas Analógicas.



Para Voltaje de Referencia



PLC887	
Comunicacion 1-Wire	
Efraim Lugo	Rev 5.0
Jose Hernandez	22/07/2010
Pagina 5 / 5	

Figura 4.16 Esquemático Comunicación 1-Wire®.

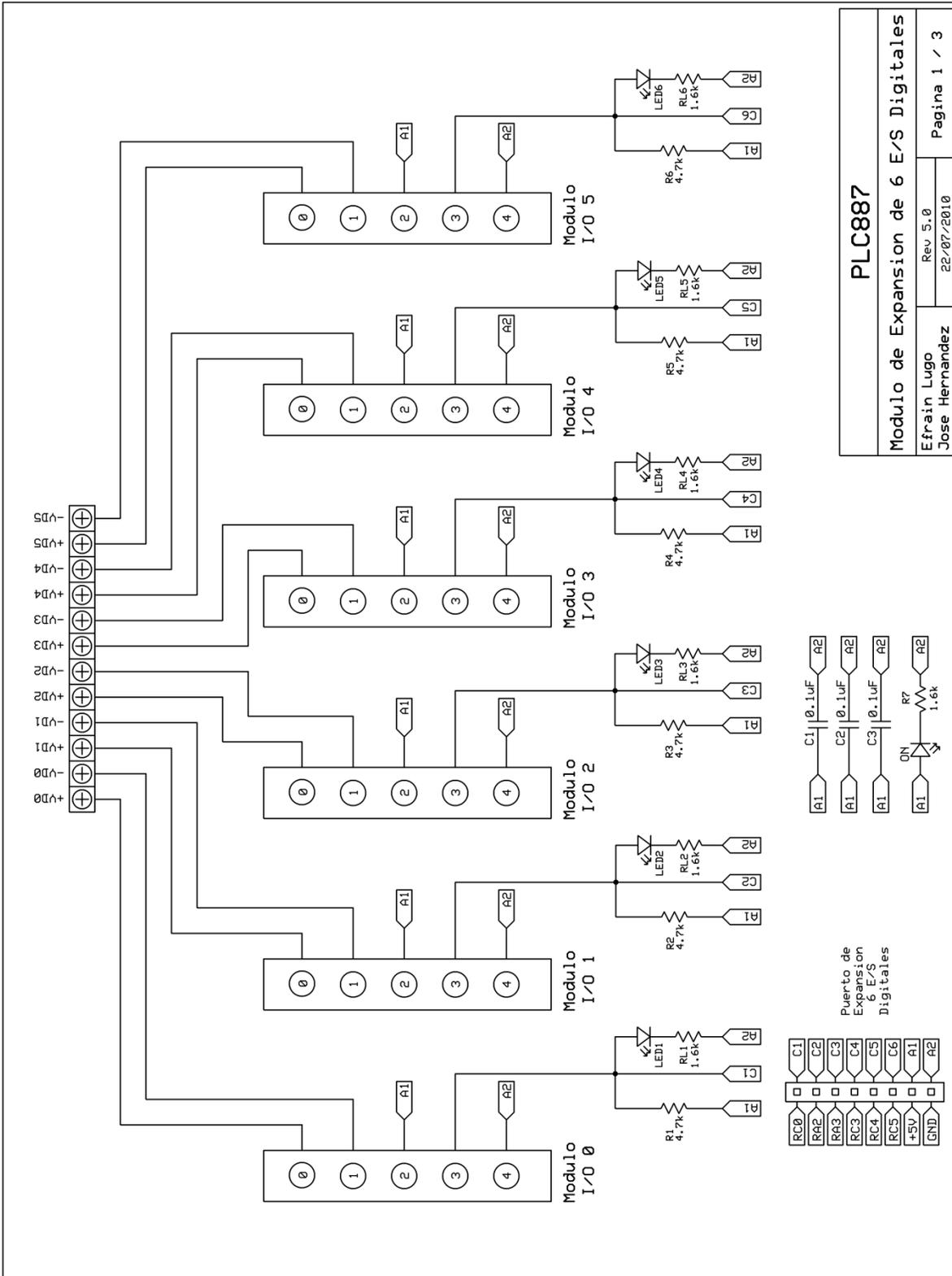


Figura 4.17 Esquemático Módulo de Expansión Digital.

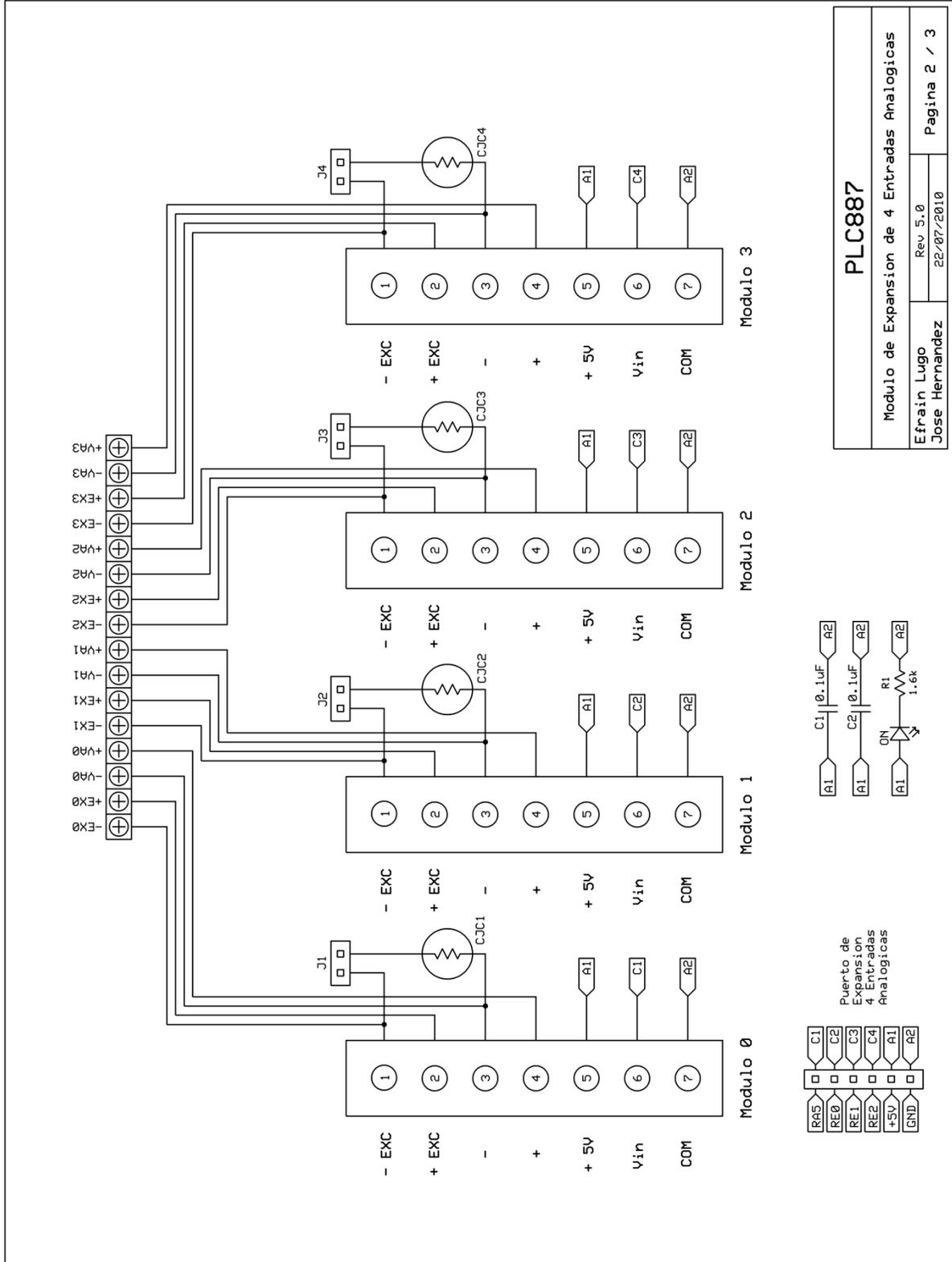


Figura 4.18 Esquemático Módulo de Expansión Analógica.

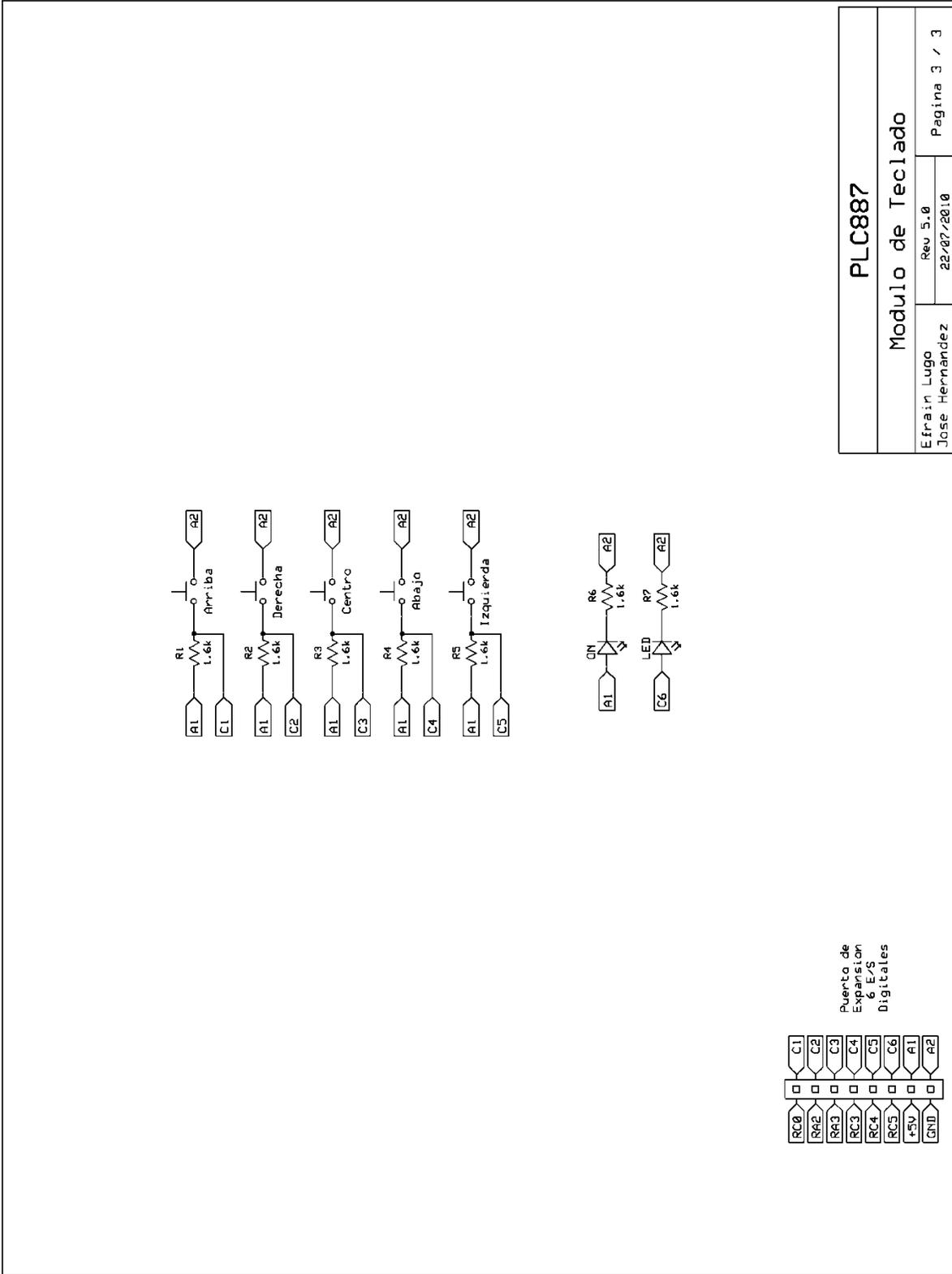


Figura 4.19 Esquemático del Módulo de Teclado.

4.3 Construcción de los diseños preliminares

En esta etapa, se hicieron los montajes experimentales del PLC (ver figuras 4.20 y 4.21) en placas de prueba sin soldaduras (ProtoBoard), con el fin de efectuar las pruebas de funcionamiento del dispositivo, pruebas de comunicación entre las diferentes interfaces de Entrada/Salida Digital/Analógica y el microcontrolador, y verificar la operatividad del equipo, esto se realizó mediante la programación del microcontrolador con algoritmos desarrollados para realizar las distintas pruebas de cada interfaz.

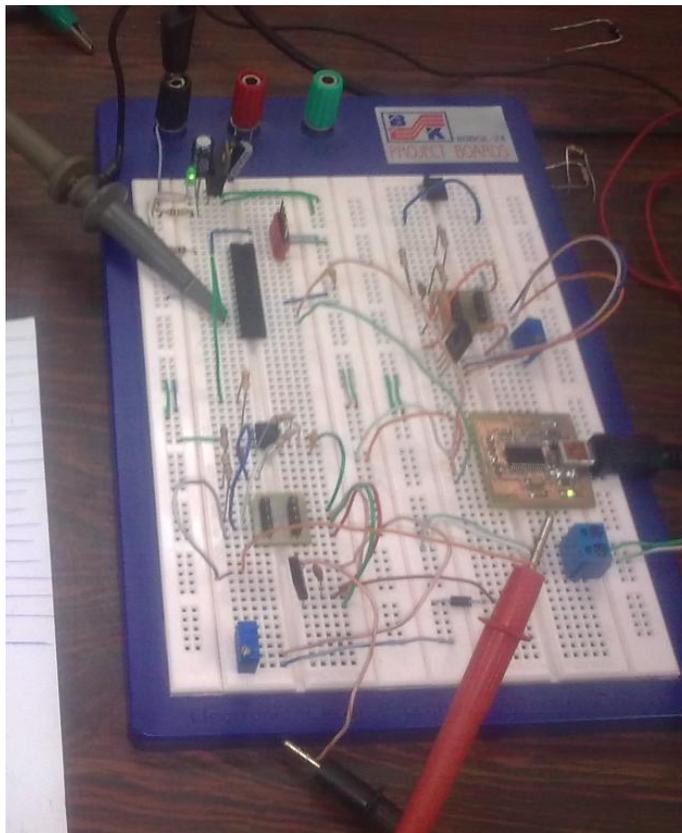


Figura 4.20 Montaje de Pruebas del PLC.

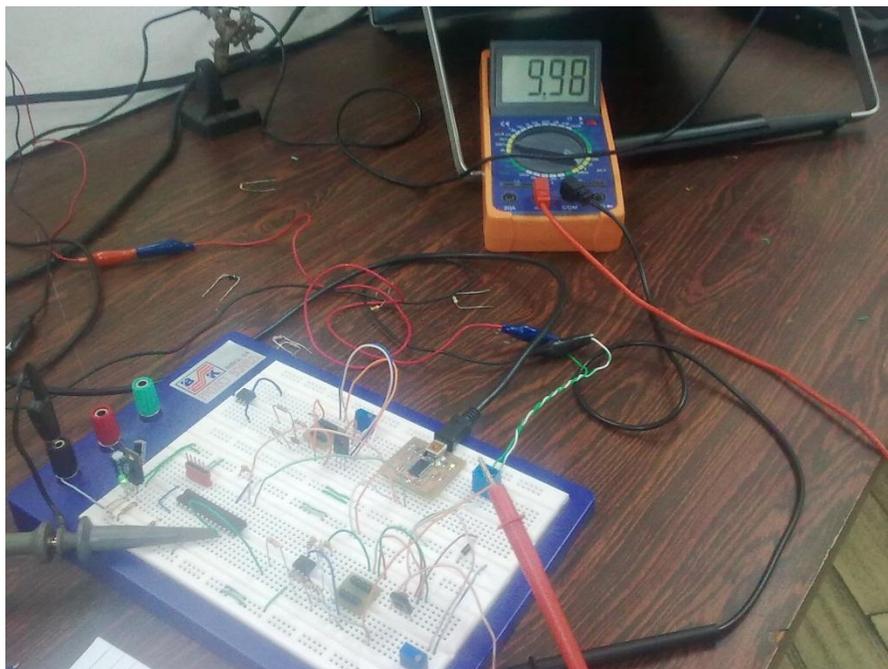


Figura 4.21 Montaje de Pruebas PLC – Interfaz de Salida de Corriente.

Los algoritmos de prueba (ver figuras 4.22 y 4.23) fueron desarrollados en el entorno de programación LDmicro, para verificar el funcionamiento y la compatibilidad del LDmicro y del microcontrolador PIC16F887; con el entorno de programación Proton IDE, se realizaron algoritmos de prueba para la verificación del funcionamiento de la pantalla LCD y para la salida en corriente, mediante la manipulación de la PWM del microcontrolador PIC.

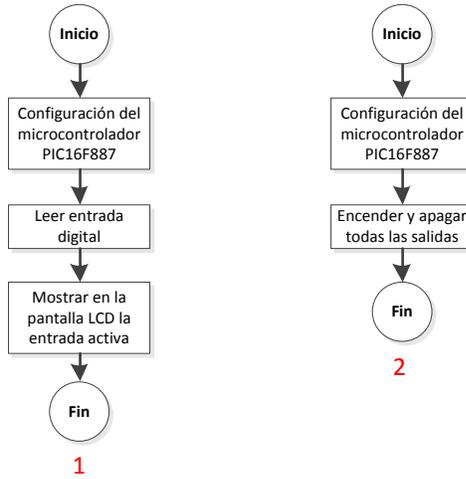


Figura 4.22 Diagrama de los algoritmos usados para las pruebas de las entradas y salidas Digitales del PLC887.

- 1- Entradas digitales.
- 2- Salidas Digitales.

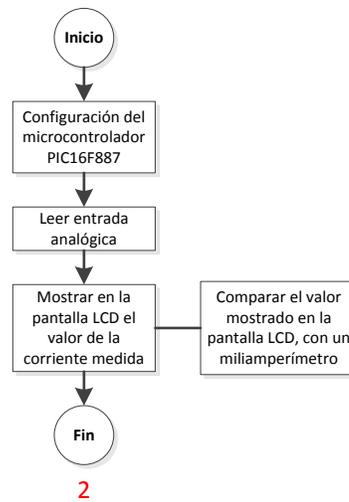
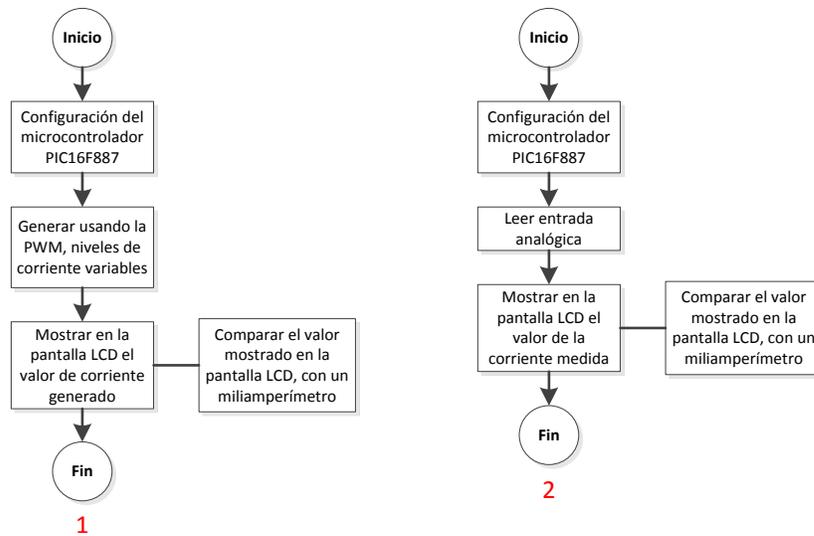


Figura 4.23 Diagrama de los algoritmos usados para las pruebas de las entradas y salidas Analógicas del PLC887.

- 1- Salidas Analógicas.
- 2- Entradas Analógicas.

La calibración y verificación de las salidas en corriente del PLC887 se realizó en el laboratorio de Instrumentación de Procesos, utilizando al controlador HoneyWell UDC1000 como elemento patrón y un miliamperímetro como elemento auxiliar.

El procedimiento fue proveer al equipo (UDC1000) con una entrada de corriente igual al valor mínimo de su rango, para la obtención del cero, luego se incrementó el valor de corriente hasta alcanzar el punto máximo del rango establecido por el instrumento (UDC1000) para finalmente con los datos obtenidos levantar la curva de calibración del PLC (ver tablas 4.8 y 4.9) (ver figuras 4.25 y 4.26).



Figura 4.24 Montaje de Pruebas PLC – Interfaz de Salida de Corriente.

Los valores ideales del ciclo útil de la PWM (mostrados a continuación en la tabla 4.8) se obtuvieron empíricamente, para ello se hicieron variaciones del ciclo útil hasta que el miliamperímetro indicara 4 mA (ciclo útil aproximadamente de 32) y 20 mA (ciclo útil aproximadamente de 240), con lo que se pudo estimar una tendencia



lineal de los valores de ciclo útil a utilizar para así obtener diferentes valores de corriente comprendidos en el rango de (4 – 20) mA.

Para la estimación de la tendencia lineal de los valores esperados a obtener por el UDC1000 se calcularon los diferentes valores de porcentaje según la corriente de entrada a este. Se tomó que el 0% corresponde a 4 mA y que el 100% corresponde a 20 mA.

Tabla 4.8 Valores Ideales para el Manejo de la PWM.

VALORES IDEALES PARA EL MANEJO DE LA PWM		
Ciclo Útil	Corriente (mA) (Vi)	UDC1000(%)
32	4	0,00
45	5	6,25
58	6	12,5
71	7	18,75
84	8	25,00
97	9	31,25
110	10	37,50
123	11	43,75
136	12	50,00
149	13	56,25
162	14	62,50
175	15	68,75
188	16	75,00
201	17	81,25
214	18	87,50
227	19	93,75
240	20	100,00

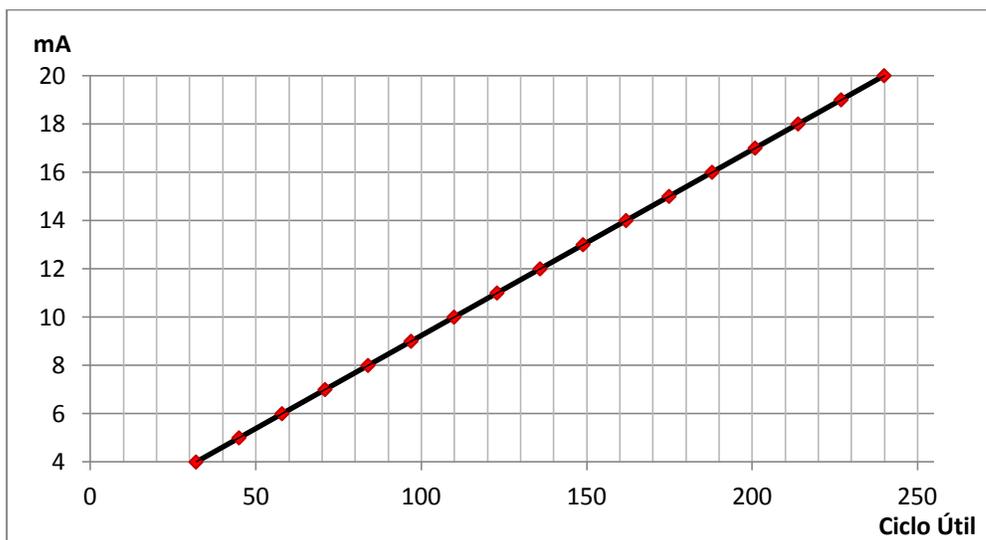


Figura 4.25 Gráfica de Valores Ideales para el Manejo de la PWM.

Tabla 4.9 Valores Reales para el Manejo de la PWM.

VALORES REALES PARA EL MANEJO DE LA PWM		
Ciclo Útil	Corriente (mA) (Vr)	UDC1000 (%)
32	4,02	0,1
43	5,01	6,3
54	5,98	12,4
66	7,04	19
78	7,97	24,8
91	8,99	31,2
104	10,02	37,6
118	11,02	43,9
131	11,97	49,8
145	12,99	56,2
159	13,98	62,4
173	15,04	69
186	16,00	75
200	17,01	81,3
215	17,95	87,2
229	18,99	93,7
244	19,97	99,8

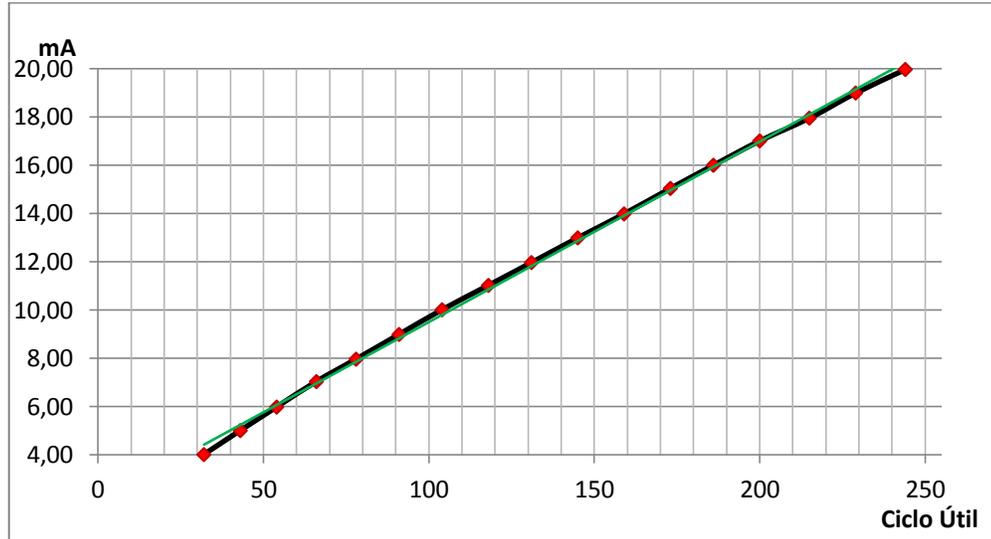


Figura 4.26 Gráfica de Valores Reales para el Manejo de la PWM.

En la tabla 4.10 se muestra el cálculo del error estático, el cual nos permite evaluar la exactitud o clase del PLC887. La exactitud o clase indica la capacidad que tiene el PLC887 (como instrumento generador de corriente) de generar corrientes cercanas al valor ideal.

El error estático se calcula como la diferencia entre el valor real (V_r) y el valor ideal (V_i):

$$\text{Error Estático} = ee = V_r - V_i$$

$$\text{Exactitud o Clase} = \frac{|ee_{MAX}|}{Span} \times 100$$

$$\text{Error Estático Porcentual} = ee(\%) = \frac{ee}{V_i} \times 100$$



Tabla 4.10 Cálculo de errores de la corriente generada por la PWM.

CÁLCULO DE ERRORES DE LA CORRIENTE GENERADA	
Error Estático (mA)	Error Estático (%)
0,02	0,50
0,01	0,20
-0,02	-0,33
0,04	0,57
-0,03	-0,38
-0,01	-0,11
0,02	0,20
0,02	0,18
-0,03	-0,25
-0,01	-0,08
-0,02	-0,14
0,04	0,27
0,00	0,00
0,01	0,06
-0,05	-0,28
-0,01	-0,05
-0,03	-0,15

Para el cálculo de la exactitud o clase, se selecciona el valor máximo del error estático, pudiendo ser este positivo o negativo, en este caso el valor máximo es -0,05mA (ver tabla 4.10), y el Span del PLC887 es de 16mA, de aquí se obtiene una exactitud o clase del $0,3125\% \approx 0,3\%$.

Los valores de corriente generados por el PLC887 (ver tabla 4.9), se obtuvieron con una exactitud del 0,3%, lo cual hace al PLC887 un instrumento exacto y confiable para el manejo de señales en el rango de (4 - 20) mA. Es importante señalar que con la realización de las pruebas se observó el comportamiento lineal de la corriente generada por la PWM del microcontrolador (ver figura 4.26).

4.4 Construcción de circuitos prototipos

Utilizando el programa Express PCB, se elaboró el circuito impreso del módulo principal del PLC, el cual fue construido con materiales de muy alta calidad con las siguientes características [27]:

- Material de construcción tipo FR4.
- Grosor del cobre de 0,007 pulgadas.
- Capa aislante protectora para las pistas de cobre.
- Agujeros con bordes de estaño para facilitar las soldaduras.
- Grosor total de la placa de 0,07 pulgadas.

A continuación se muestra el diseño de la tarjeta de circuito impreso del módulo principal del PLC, en su capa superior (ver figura 4.27) y en su capa inferior (ver figura 4.28).

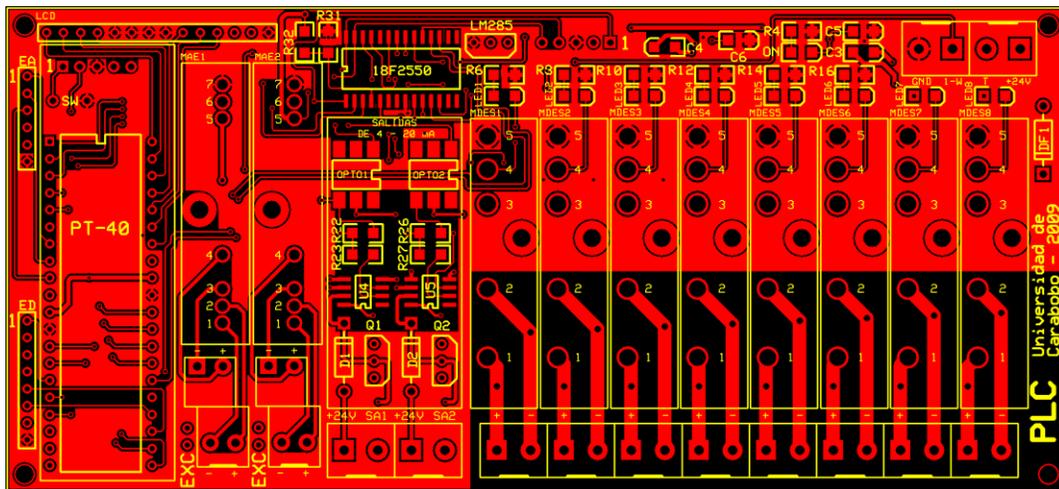


Figura 4.27 Circuito Impreso del Módulo Principal del PLC (Capa Superior).

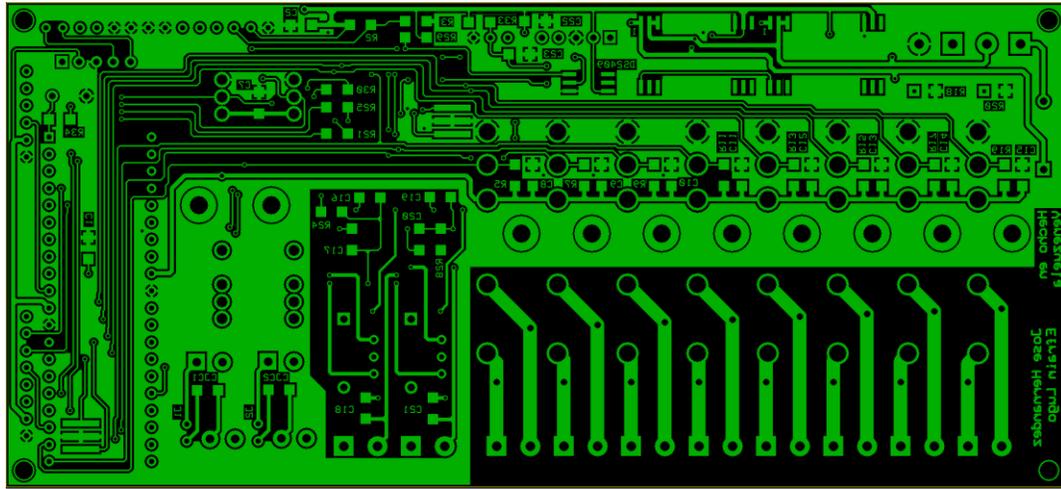


Figura 4.28 Circuito Impreso del Módulo Principal del PLC (Capa Inferior).

En las siguientes figuras se muestra el acabado final de las tarjetas de circuito impreso del módulo principal del PLC, fabricadas por Express PCB (ver figuras 4.29 y 4.30).

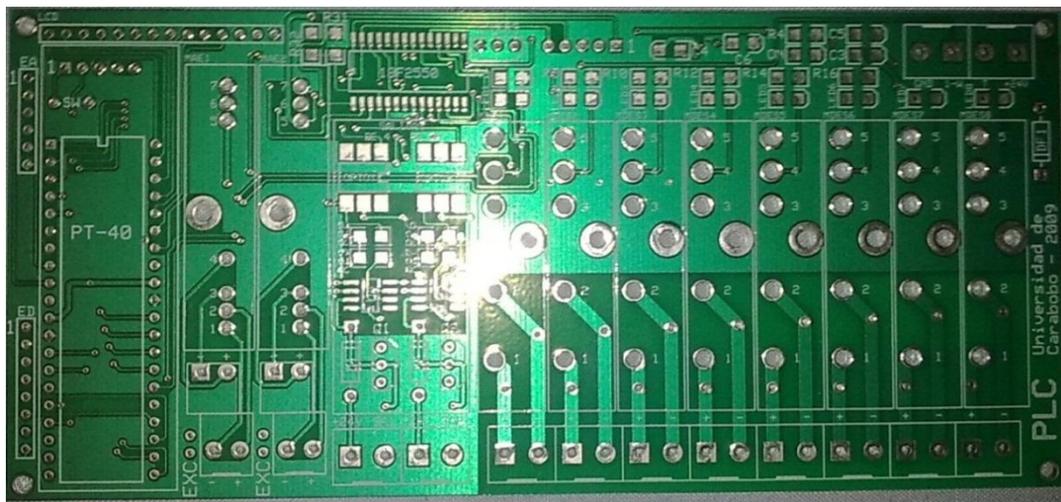


Figura 4.29 Acabado final de la Tarjeta de Circuito Impreso del Módulo Principal del PLC (Capa Superior).

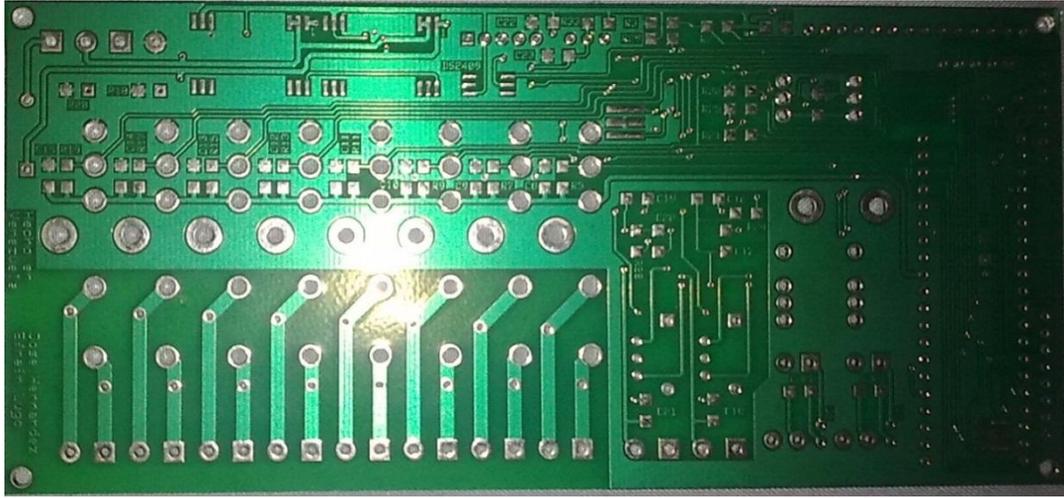


Figura 4.30 Acabado final de la Tarjeta de Circuito Impreso del Módulo Principal del PLC (Capa Inferior).

Seguidamente se muestra el módulo principal del PLC totalmente ensamblado (ver figuras 4.31 y 4.32).

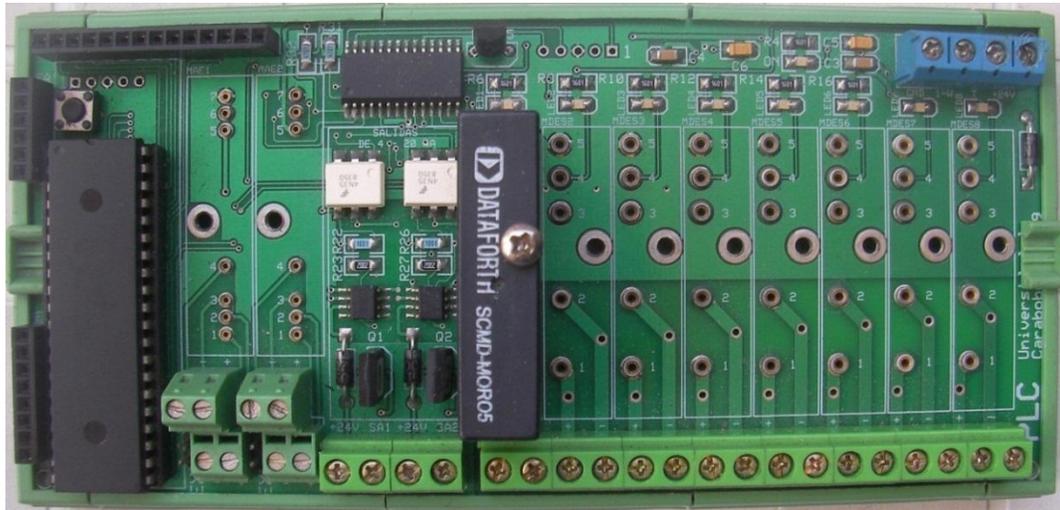


Figura 4.31 Circuito Impreso del Módulo Principal del PLC Totalmente Ensamblado (Vista Superior).

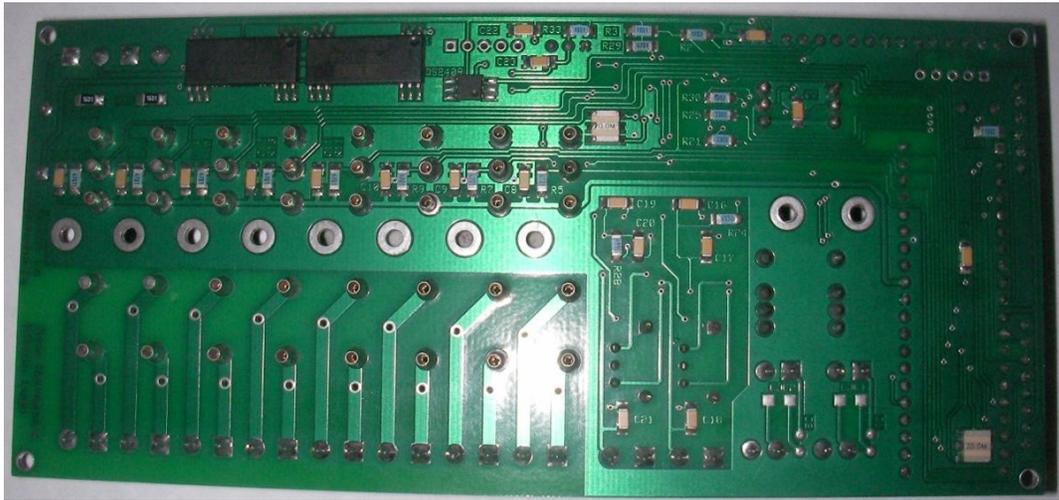


Figura 4.32 *Circuito Impreso del Módulo Principal del PLC Totalmente Ensamblado (Vista Inferior).*

A continuación en la figura 4.33, se muestran los puertos de conexión y borneras para las diferentes entradas y salidas del PLC887. Se recomienda consultar el “Manual del PLC887” contenido en el anexo B, para más detalles.

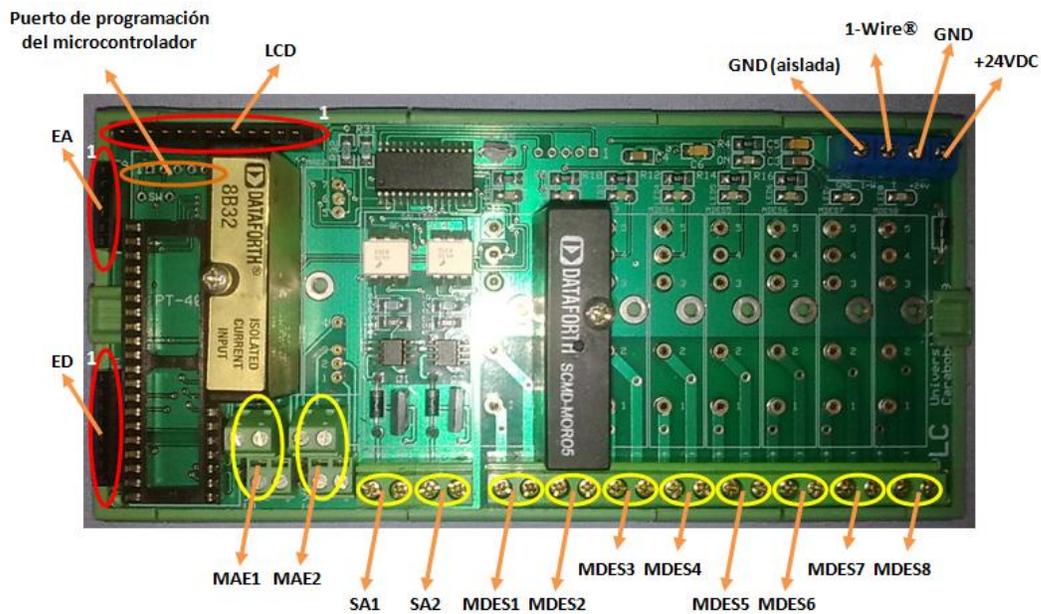


Figura 4.33 *Puertos de Conexión del PLC887.*

Para la elaboración de los módulos de expansión digital, analógica y de teclado, se utilizó el programa Express PCB, la construcción de los mismos se realizó de forma artesanal por medio de fotograbado.

A continuación se muestra el diseño de los módulos de expansión digital y analógica (ver figura 4.33) y el módulo del teclado (ver figura 4.34), y como resultado final se obtuvieron los módulos de expansión y de teclado ensamblados (ver figuras 4.35, 4.36 y 4.37)

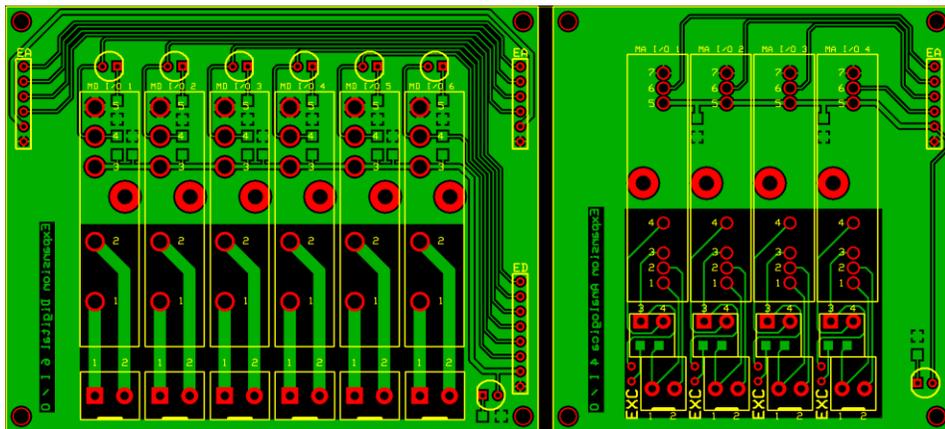


Figura 4.34 Circuito Impreso de los Módulos de Expansión.

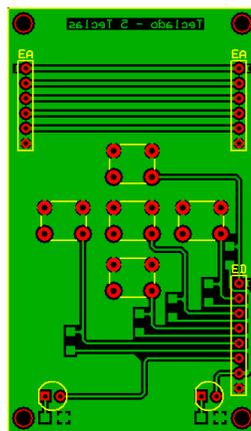


Figura 4.35 Circuito Impreso del Módulo de Teclado.

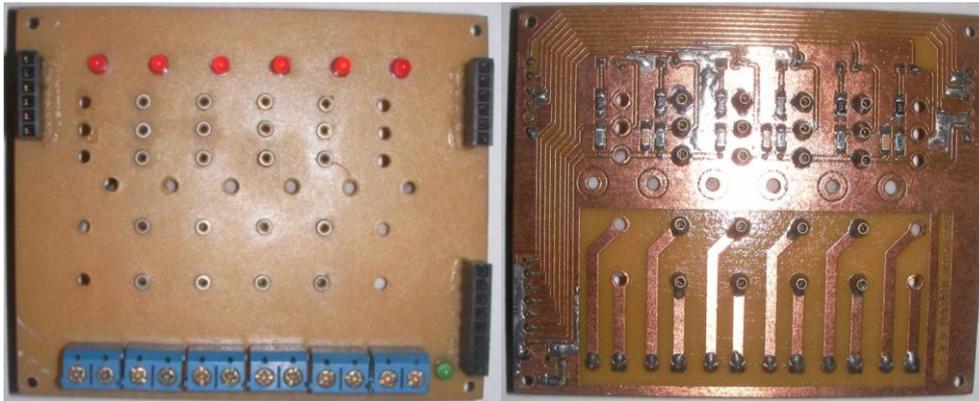


Figura 4.36 *Circuito Impreso del Módulo de Expansión Digital Totalmente Ensamblado (Vista Superior e inferior).*

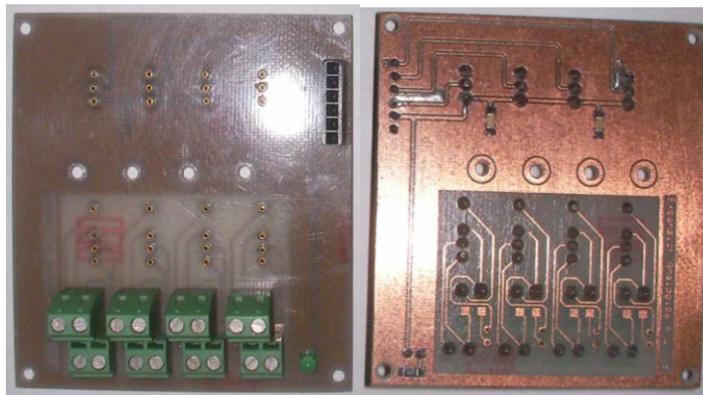


Figura 4.37 *Circuito Impreso del Módulo de Expansión Analógica Totalmente Ensamblado (Vista Superior e inferior).*

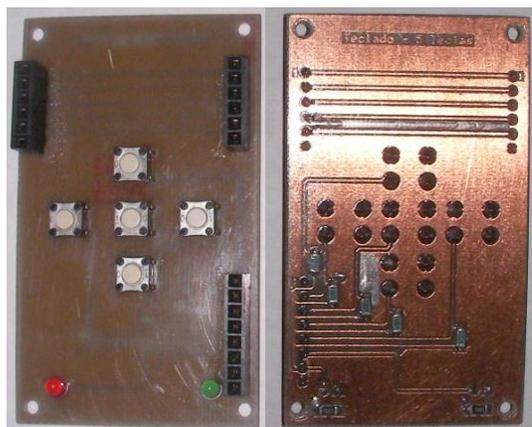


Figura 4.38 *Circuito Impreso del Módulo de Teclado Totalmente Ensamblado (Vista Superior e inferior).*



También, se incluyen las listas de componentes utilizados en la construcción del PLC887 y sus módulos accesorios, con una referencia de costos actualizados a la fecha (los precios fueron calculados en USD a paridad 6,30Bs/USD y no incluyen costos de envío). Esto se ve reflejado en las tablas 4.11, 4.12, 4.13 y 4.14.

Tabla 4.11 Elementos que componen el Modulo Principal PLC887.

ITEM	CANT	DESCRIPCIÓN	CODIGO	COSTO USD	COSTO Bs.	TOTAL Bs.	REF
C1, C2, C7..C23	19	Capacitor 0,1 μ F @ 50V SMD	311-1179-1-ND	0,11	0,69	13,17	(1)
C3, C4	2	Capacitor 0,47 μ F @ 50V SMD	445-1380-1-ND	0,24	1,51	3,02	(1)
C5, C6	2	Capacitor 1 μ F @ 50V SMD	80-C1206104K5RAC7867	0,10	0,63	1,26	(2)
N/A	54	Base Terminal Relé Ind.	ED5028-ND	0,68	4,28	231,34	(1)
BD1..BD8, SAI1,SA2, +24V, 1-W	12	Conector 5mm 2 Polos	277-1236-ND	0,86	5,42	65,02	(1)
EA1, EA2	2	Conector 3,81mm 2 Polos	651-1708026	3,02	19,03	38,05	(2)
DF1, D1, D2	3	Diodo Rectificador 1A @ 1000V _{DC}	1N4007FSCT-ND	0,14	0,88	2,65	(1)
LED1..LED8	8	Led rojo corto 635nm SMD	67-1552-1-ND	0,49	3,09	24,70	(1)
ON	1	Led verde corto 635nm SMD	67-1553-1-ND	0,47	2,96	2,96	(1)
OPTO1, OPTO2	2	Optoacoplador 4N35 SMD	512-4N35SR2M	0,58	3,65	7,31	(1)
LCD	1	Conector receptáculo 16 pos 0,1"	A32549-ND	3,35	21,11	21,11	(1)
EA	1	Conector receptáculo 6 pos 0,1"	A32907-ND	1,43	9,01	9,01	(1)
ED	1	Conector receptáculo 8 pos 0,1"	A32909-ND	2,69	16,95	16,95	(1)
Q1, Q2	2	Transistor bipolar de 3A 60V	863-2N4922G	0,58	3,65	7,31	(2)
R2, R30, R34	3	Resistencia 10k SMD 1% 1/4W	660-RK73H2BTTD1002F	0,10	0,63	1,89	(1)



Tabla 4.11 Elementos que componen el Modulo Principal PLC887 (Continuación).

ITEM	CANT	DESCRIPCIÓN	CODIGO	COSTO USD	COSTO Bs.	TOTAL Bs.	REF
R4, R6, R8, R10, R12, R14, R16, R18, R20	9	Resistencia 1,6k SMD 1% 1/4W	71-CRCW1206-1.6K-E3	0,22	1,39	12,47	(2)
R5, R7, R9, R11, R13, R15, R17, R19, R29, R31, R32	11	Resistencia 4,7k SMD 1% 1/4W	660-RK73H2BTDD4701F	0,10	0,63	6,93	(2)
R3, R22, R26, R33	4	Resistencia 1k SMD 1% 1/4W	660-RK73H2BTTE1001F	0,10	0,63	2,52	(2)
R21, R25	2	Resistencia 330 SMD 1% 1/4W	660-RK73H2BTDD3300F	0,10	0,63	1,26	(2)
R24, R28	2	Resistencia 499 SMD 1% 1/4W	660-RK73H2BTDD4990F	0,10	0,63	1,26	(2)
R23, R27	2	Resistencia 20k SMD 1% 1/4W	311-20.0KFRCT-ND	0,10	0,63	1,26	(1)
LM285	1	Referencia de voltaje 1,235V	660-RK73H2BTDD3300F	0,10	0,63	0,63	(2)
SW	1	Pulsador	P12228SCT-ND	0,29	1,83	1,83	(1)
U1	1	Microcontrolador PIC16F887 DIP	PIC16F887-I/P-ND	2,64	16,63	16,63	(1)
18F2550	1	Microcontrolador PIC18F2550 SOIC	PIC18F2550-I/SO-ND	4,96	31,25	31,25	(1)
U2, U3	2	Convertidor DC-DC 24-5V SOIC	296-12367-5-ND	11,39	71,76	143,51	(1)
U4, U5	2	Transmisor de corriente 4-20mA SOIC	XTR115UA-ND	3,78	23,81	47,63	(1)
XTAL1, XTAL2	2	Resonador de 20MHz SMD	815-AWSCR-20.00MTD-T	0,49	3,09	6,17	(2)
PCB	1	Tarjeta de circuito impreso	N/A	41,50	261,45	261,45	(3)
Total Bs.						980,53	

(1) www.digikey.com (2) www.mouser.com (3) www.expresspcb.com



Tabla 4.12 Elementos que componen el Modulo de Expansión Digital.

ITEM	CANT	DESCRIPCION	CODIGO	COSTO USD	COSTO Bs.	TOTAL Bs.	REF
C1..C3	3	Capacitor 0,1 μ F @ 50V SMD	311-1179-1-ND	0,11	0,69	2,08	(1)
N/A	30	Base Terminal Relé Ind.	ED5028-ND	0,68	4,28	128,52	(1)
BD1..BD6	6	Conector 5mm 2 Polos	277-1236-ND	0,86	5,42	32,51	(1)
LED1..LED6	6	Led rojo 3mm	751-1129-ND	0,53	3,34	20,03	(1)
ON1	1	Led verde 3mm	751-1101-ND	0,47	2,96	2,96	(1)
EA	2	Conector receptáculo 6 pos 0,1"	A32907-ND	1,43	9,01	18,02	(1)
ED	1	Conector receptáculo 8 pos 0,1"	A32909-ND	2,19	13,80	13,80	(1)
RL1..RL6, R7	7	Resistencia 1,6k SMD 1% 1/4W	71-CRCW1206-1.6K-E3	0,22	1,39	9,70	(2)
R1..R6	6	Resistencia 4,7k SMD 1% 1/4W	660-RK73H2BTDD4701F	0,10	0,63	3,78	(2)
PCB	1	Tarjeta de circuito impreso	N/A	20,75	130,73	130,73	(3)
Total Bs.						362,12	

(1) www.digikey.com (2) www.mouser.com (3) www.expresspcb.com

Tabla 4.13 Elementos que componen el Modulo de Expansión Analógica.

ITEM	CANT	DESCRIPCION	CODIGO	COSTO USD	COSTO Bs.	TOTAL Bs.	REF
C1, C2	2	Capacitor 0,1 μ F @ 50V SMD	311-1179-1-ND	0,11	0,69	1,39	(1)
N/A	28	Base Terminal Relé Ind.	ED5028-ND	0,68	4,28	119,95	(1)
BA1..BA4	4	Conector 3,81mm 2 Polos	651-1708026	3,02	19,03	76,10	(2)
ON2	1	Led verde 3mm	751-1101-ND	0,47	2,96	2,96	(1)
EA	1	Conector receptáculo 6 pos 0,1"	A32907-ND	1,43	9,01	9,01	(1)
R1	9	Resistencia 1,6k SMD 1% 1/4W	71-CRCW1206-1.6K-E3	0,22	1,39	12,47	(2)
PCB	1	Tarjeta de circuito impreso	N/A	13,83	87,15	87,15	(3)
Total Bs.						309,04	

(1) www.digikey.com (2) www.mouser.com (3) www.expresspcb.com



Tabla 4.14 Elementos que componen el Módulo de Teclado.

ITEM	CANT	DESCRIPCION	CODIGO	COSTO USD	COSTO Bs.	TOTAL Bs.	REF
LED	1	Led rojo 3mm	751-1129-ND	0,53	3,34	3,34	(1)
ON	1	Led verde 3mm	751-1101-ND	0,47	2,96	2,96	(1)
EA	2	Conector receptáculo 6 pos 0,1"	A32907-ND	1,43	9,01	18,02	(1)
ED	1	Conector receptáculo 8 pos 0,1"	A32909-ND	2,19	13,80	13,80	(1)
R1..R7	7	Resistencia 1,6k SMD 1% 1/4W	71-CRCW1206-1.6K-E3	0,22	1,39	9,70	(2)
SW1..SW2	5	Pulsador	SW403-ND	0,41	2,58	12,92	(1)
PCB	1	Tarjeta de circuito impreso	N/A	10,38	65,36	65,36	(3)
Total Bs.						126,09	

(1) www.digikey.com (2) www.mouser.com (3) www.expresspcb.com

4.5 Adaptación del compilador de lenguaje escalera distribuido bajo licencia GNU LDmicro

Para entender el código fuente del programa LDmicro desarrollado por Jonathan Westhues [17], se estudió lo esencial del lenguaje de programación C++ (lenguaje en el que se desarrolló el LDmicro), con la ayuda del entorno de programación Visual Studio 2008, se propuso realizar cambios pertinentes que estuviesen acordes a nuestros intereses, los cambios a realizar fueron:

- Se realizó la traducción del programa a español en su totalidad incluyendo la ayuda del mismo.
- Se cambiaron las teclas de acceso rápido a funciones especiales de manera que fuesen más intuitivos al español.
- Se eliminaron de la lista de microcontroladores aquellos que no se van a utilizar.
- Se eliminó el menú de configuración de parámetros del microcontrolador, y quedaron ajustados en los siguientes valores:
 - Velocidad del Oscilador = 20MHz

- Tiempo de Ciclo = 10ms
- Microcontrolador predeterminado = Microchip PIC16F887 40-PDIP
- Microcontrolador opcional = Microchip PIC16F877 40-PDIP

Como tarea final se realizó un programa de prueba para verificar el funcionamiento del PLC con la nueva versión del LDmicro (ver figura 4.39).

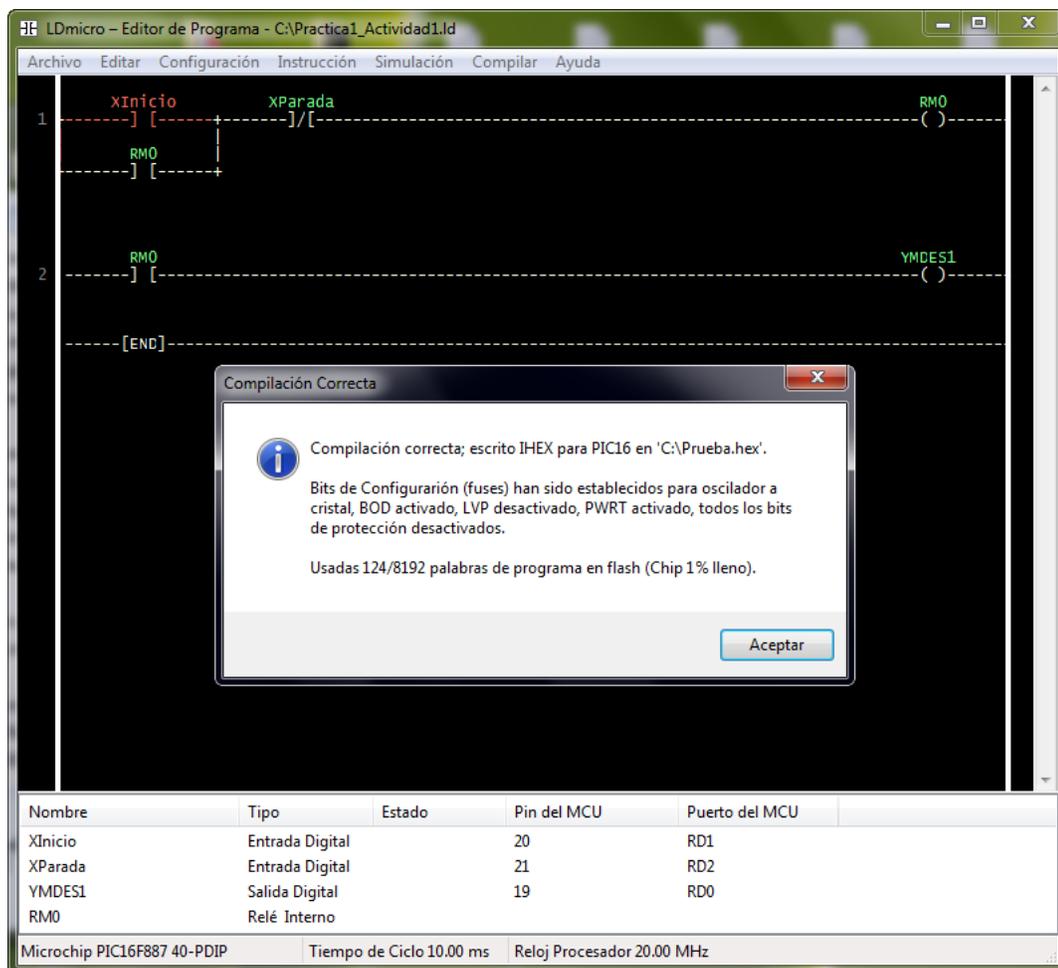


Figura 4.39 Programa de Prueba del LDmicro después de los cambios.

4.6 MonitorPLC887

El MonitorPLC887 es un programa que fue diseñado para establecer la comunicación entre el PLC887 y el computador, empleando para ello el protocolo 1-Wire®; este permite realizar el monitoreo de las entradas y salidas del PLC887. La aplicación fue desarrollada en ambiente Microsoft™ Visual Basic Express 2008 (ver figura 4.40).



Figura 4.40 Ventana Principal del MonitorPLC887.

La interfaz de hardware para comunicación 1-Wire® del PLC887 está conformada por un acoplador de MicroLAN DS2409 y un microcontrolador PIC18F2550.

Para poder utilizar el MonitorPLC887 se instaló en el computador el servidor OPC OmniServer Pro, el cual se encarga de gestionar y controlar la comunicación 1-Wire® entre el computador y el PLC887 (ver figura 4.41) y se configuró según lo indicado en el Manual del Programa MonitorPLC887 (ver anexo B).

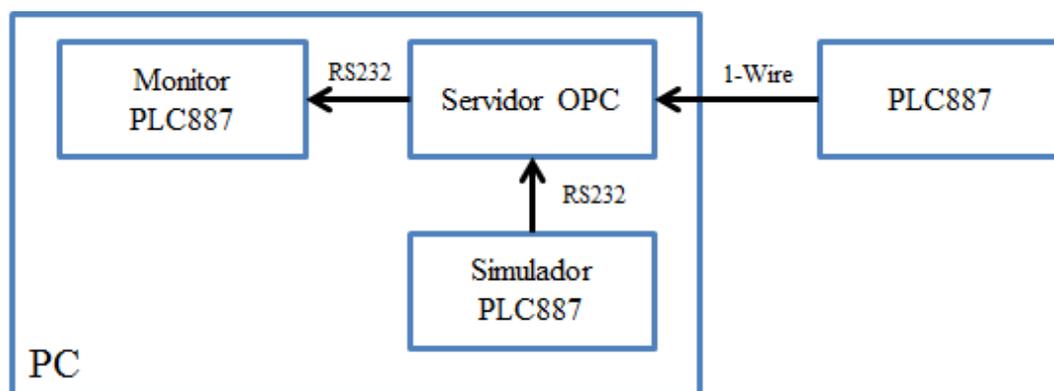


Figura 4.41 Diagrama de Bloques de la Topología de la Comunicación del PLC887 y el Computador.

Este proyecto contempló el diseño de la plataforma de hardware necesaria para permitir la comunicación 1-Wire® y el desarrollo de la aplicación de monitoreo, no incluyendo la implementación del servidor de datos 1-Wire® a OPC. Para verificar el correcto funcionamiento del programa MonitorPLC887, se realizaron pruebas de comunicación utilizando un simulador del PLC887 (ver figura 4.41), el cual envía los datos correspondientes de las variables simuladas al servidor OPC OmniServer Pro por medio del protocolo de comunicación RS232, para posteriormente ser procesadas por el programa MonitorPLC887. El simulador se elaboró empleando el software de simulación de electrónica, PROTEUS Design Suite (ver figura 4.42).

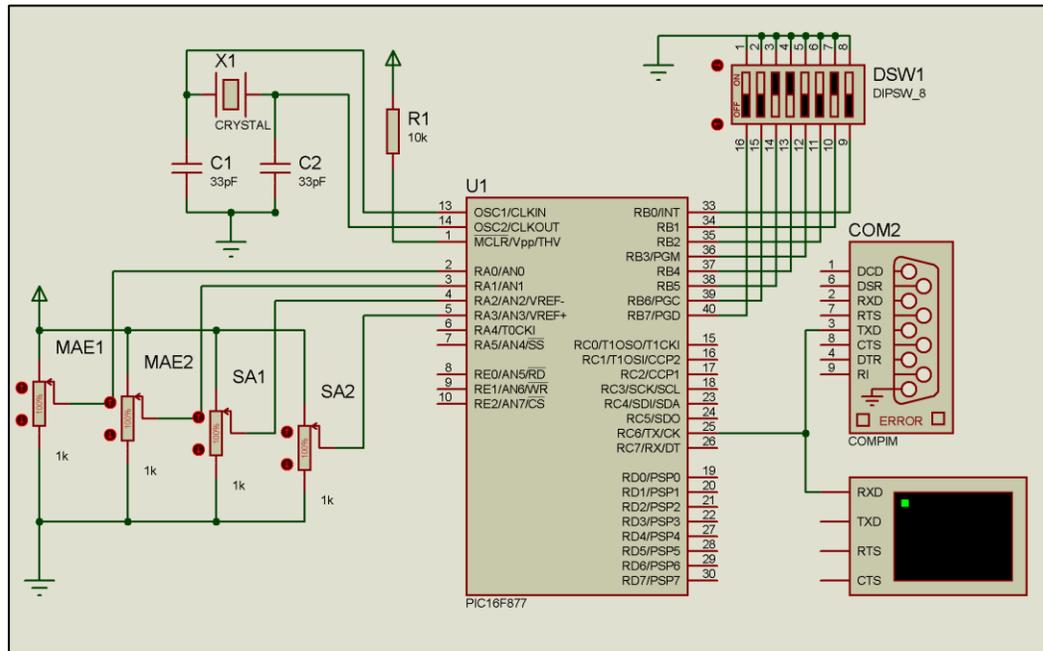


Figura 4.42 Simulador del PLC887 realizado con el software de simulación **PROTEUS Design Suite**.

Una vez finalizado el desarrollo de la aplicación se facilita al usuario un entorno visual donde puede monitorear y graficar las variables de interés capturadas por el PLC887, a continuación se muestra la ventana principal de la aplicación en detalle (ver figura 4.43).

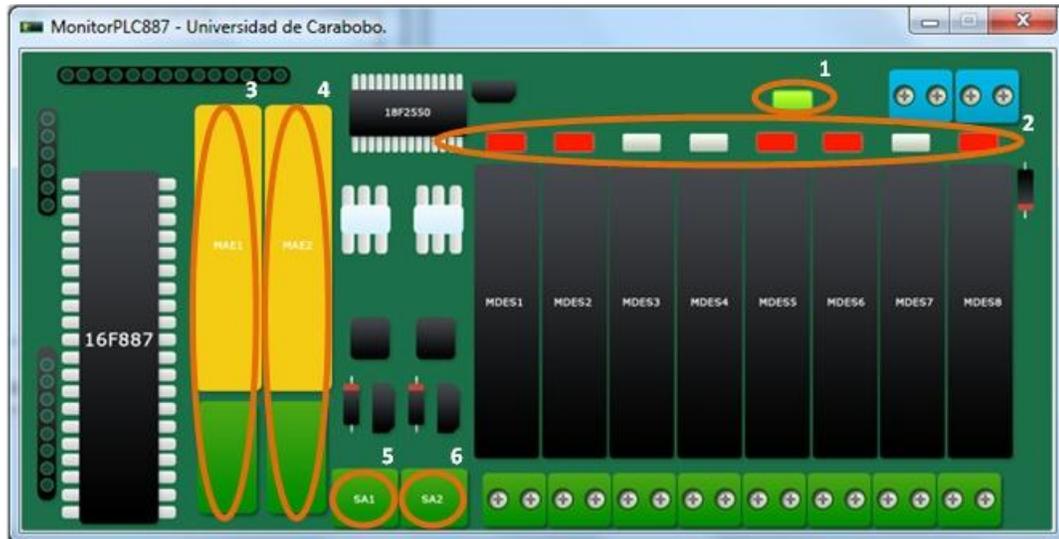


Figura 4.43 Descripción de la Ventana Principal del MonitorPLC887.

- 1- Indica si el PLC887 está conectado al computador. Se mostrará de color rojo si está desconectado y color verde si está conectado.
- 2- Indica el estado de las variables digitales. Se mostrarán en gris cuando estén desactivadas y en rojo cuando estén activadas.
- 3- Al hacer doble clic en una de ellas se desplegará una nueva ventana donde se observará el comportamiento (de forma gráfica) de la entrada analógica seleccionada.
- 4- Al hacer doble clic en una de ellas se desplegará una nueva ventana donde se observará el comportamiento (de forma gráfica) de la salida analógica seleccionada.

En la ventana de gráficas (ver figura 4.44) se observará el comportamiento de cada una de las salidas y/o entradas analógicas del PLC887.

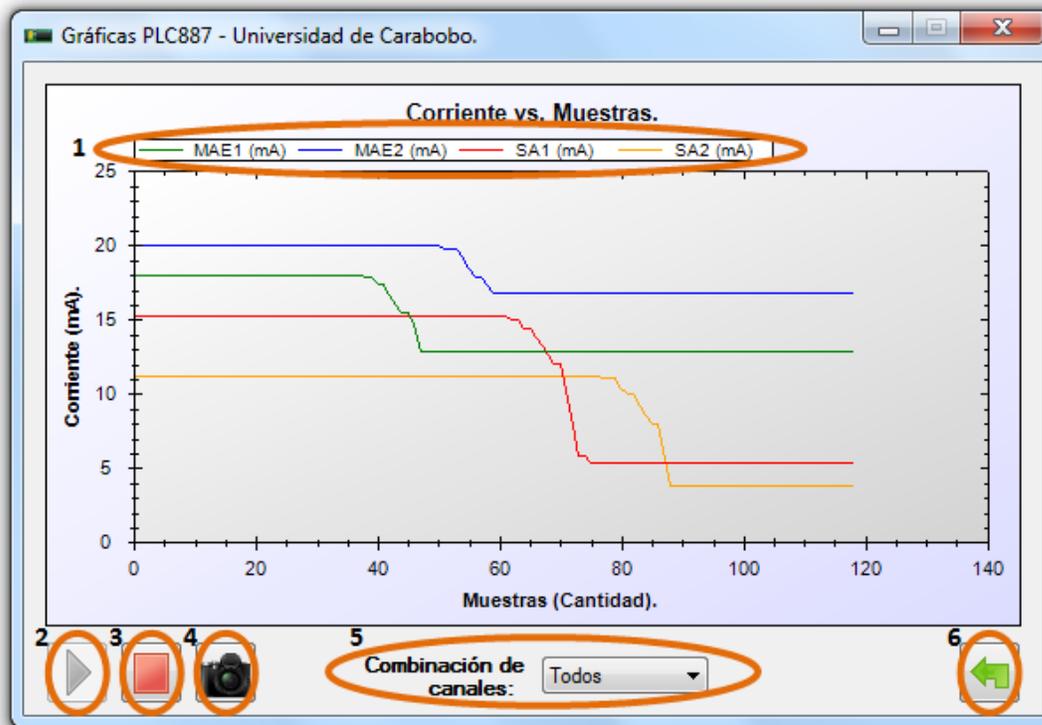


Figura 4.44 Descripción de la Ventana de Gráficas del Monitor PLC887.

- 1- Indica el o los canales de entradas o salidas analógicas seleccionadas en la ventana principal. Mostrará el nombre y el color correspondiente para cada canal seleccionado.
- 2- Botón para empezar a graficar la información que está siendo muestreada por el PLC887. Se estará graficando desde que el usuario seleccione algún canal en la ventana principal. Se mostrará en gris cuando esté deshabilitada esta función y en verde cuando esté habilitada.
- 3- Botón para detener el graficado de información que está siendo muestreada por el PLC887. Se mostrará en gris cuando esté deshabilitada esta función y en rojo cuando esté habilitada.
- 4- Botón de captura de imagen. Con esta función el usuario podrá tomar y almacenar una foto, en ella se mostrará lo que en ese momento el usuario estaba viendo. Las imágenes estarán almacenadas en la carpeta “Capturas”



ubicada en el directorio de instalación del programa (también habrá un acceso directo en el escritorio).

- 5- Selección de canales. En esta sección el usuario podrá seleccionar la combinación de canales que más le convenga para el estudio del comportamiento de las entradas o salidas analógicas del PLC887.
- 6- Botón para regresar a la ventana principal del programa y salir de la ventana de gráficas.

4.6.1 Pruebas de funcionamiento

Las pruebas de funcionamiento del programa MonitorPLC887 se realizaron utilizando un simulador del PLC887. Las pruebas fueron las siguientes:

- Se simularon las señales de entradas y salidas digitales con el simulador PLC887, a través de un dip switch, dichas señales se presentaron de forma correspondiente en el programa MonitorPLC887 (ver figura 4.45).
- Se simularon las señales de entradas y salidas analógicas con el simulador PLC887, a través de resistencias variables, dichas señales se evidenciaron de forma correspondiente en el programa MonitorPLC887 (ver figura 4.46).

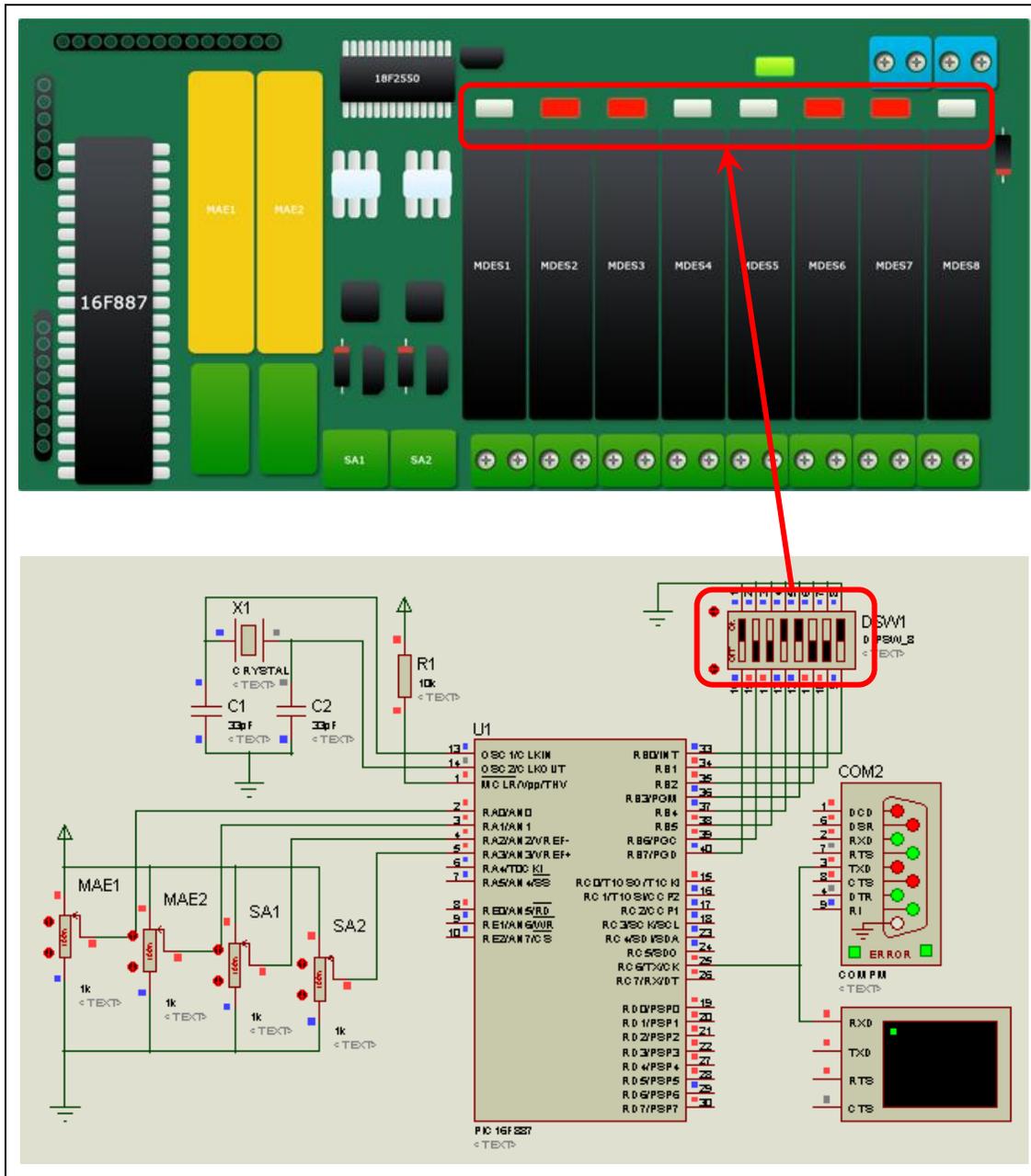


Figura 4.45 Prueba de monitoreo de entradas y salidas digitales con el MonitorPLC887.

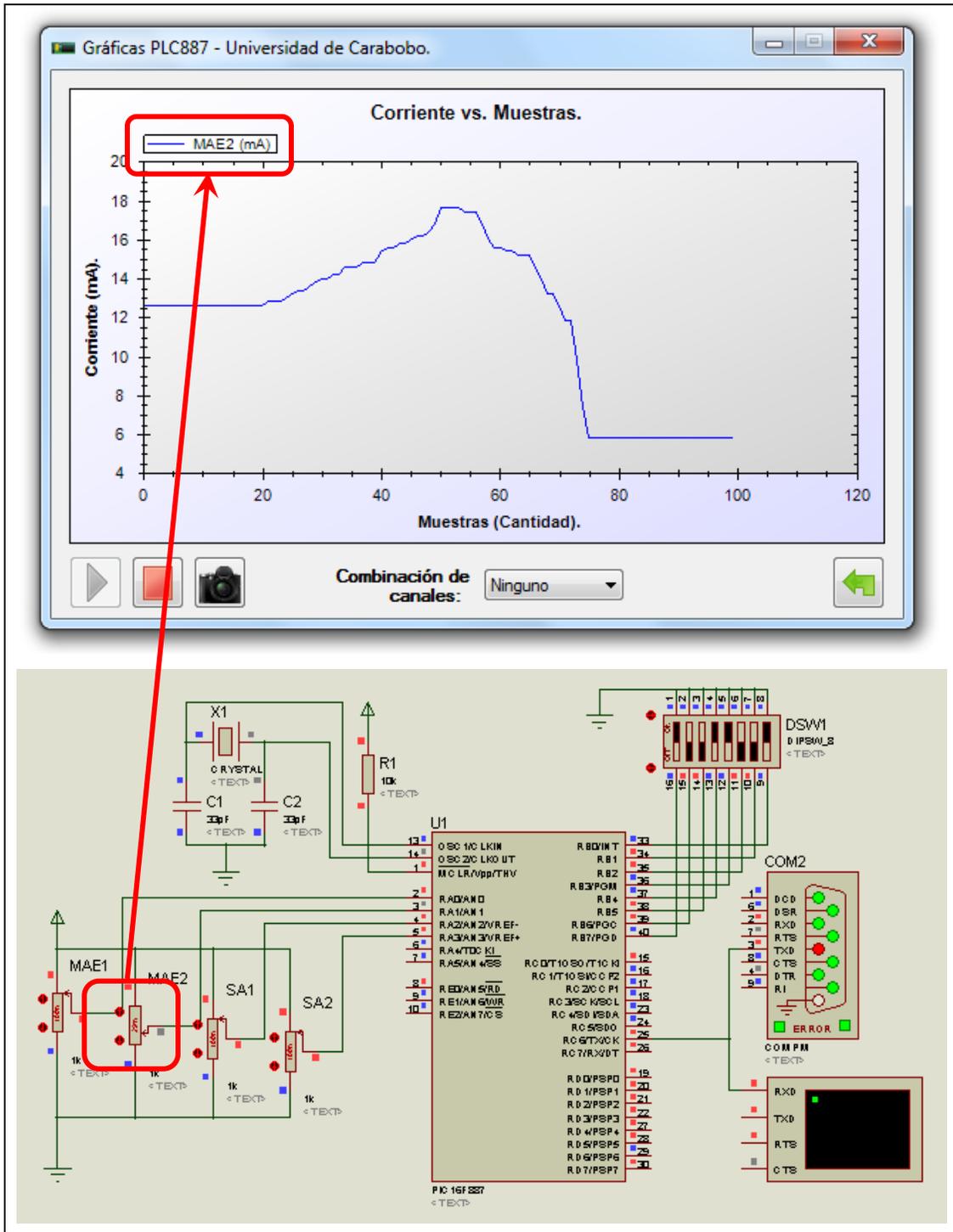


Figura 4.46 Prueba de monitoreo de entrada analógica con el MonitorPLC887.



4.7 Elaboración de las prácticas de laboratorio

Se realizaron tres prácticas de laboratorio para dar a conocer las bondades presentes en el PLC887 como dispositivo de control. Cada práctica está estructurada de la siguiente manera; objetivos, pre-laboratorio, laboratorio e informe, con el propósito de entrenar a los estudiantes de Ingeniería Eléctrica de la Universidad de Carabobo a emplear herramientas de programación y dispositivos alternativos para el desarrollo de actividades de control.

El contenido de cada práctica se explica a continuación:

- Práctica 1, Introducción al PLC887 y a su programación en lenguaje escalera utilizando el LDmicro. En esta práctica se pretende dar a conocer las características del PLC887 e introducir al LDMicro como herramienta de programación.
- Práctica 2, Arranque, parada y cambio de giro de un motor trifásico. El objetivo de esta experiencia es, utilizando al PLC887 y el LDmicro, arrancar y hacer la inversión de giro a un motor trifásico.
- Practica 3, Introducción al *Proton IDE* como herramienta alternativa de programación del PLC887. Esta práctica tiene como objetivo, presentar los comandos básicos de programación del *Proton IDE* para el manejo de las entradas y salidas tanto digitales como analógicas del PLC887.

Las prácticas y sus respectivas soluciones se encuentran en el anexo C.



4.8 Cargar archivos hexadecimales (.hex) al PLC887

Una vez realizada la compilación del programa que se quiere cargar en el PLC887 y obtenido el archivo hexadecimal (.hex), éste se puede grabar en el microcontrolador utilizando cualquier software y hardware que se tenga para programar microcontroladores.

En el "Manual del PLC887" contenido en el anexo B, se muestra con detalle el procedimiento usado por los autores para la programación del microcontrolador del PLC887.



CAPÍTULO V

CONCLUSIONES

Los microcontroladores son ampliamente conocidos por su versatilidad en la ejecución de diversas tareas, en este trabajo de investigación se propuso su aplicación para una función específica; demostrar que es posible construir un PLC con un microcontrolador como dispositivo de control, ya que la arquitectura de un PLC está constituida por entradas y salidas digitales y analógicas, características que comparte con el microcontrolador.

Al culminar el diseño y construcción del prototipo de PLC, para ser usado en aplicaciones didácticas e industriales, se obtuvo lo siguiente:

Los equipos obtenidos como producto final fueron: Módulo principal de control llamado PLC887, el cual cuenta con 8 entradas/salidas digitales, 2 entradas analógicas (pueden ser en corriente o en tensión dependiendo del módulo seleccionado), 2 salidas analógicas en corriente de 4 a 20 mA y comunicación 1-Wire®. Módulo de expansión digital de 6 entradas/salidas. Módulo de expansión analógica de 4 entradas (pueden ser en corriente o en tensión dependiendo del módulo seleccionado). Módulo de teclado de 5 teclas (entradas) y 1 led (salida). Los dispositivos construidos son de bajo costo, alta funcionalidad y fueron diseñados con prestaciones de hardware especiales para su aplicación en laboratorios didácticos y en ambientes industriales, cabe destacar que los módulos fueron diseñados (en cuanto a dimensiones) para estar contenidos en carcasas electrónicas modulares, las cuales pueden ser instaladas fácilmente en rieles tipo DIN. En los ensayos de generación de corriente (usando la PWM del microcontrolador), el PLC887 presentó un comportamiento lineal y una exactitud del 0,3%, demostrando ser un instrumento exacto y confiable en el rango de (4 - 20) mA.



Las entradas/salidas digitales y las entradas analógicas del PLC887 se configuran de acuerdo a las necesidades del proceso a controlar; se podrán seleccionar los módulos de entrada digital (para tensiones AC y DC) y salida digital (cargas AC y DC) a utilizar, así como también los de entrada analógica dependiendo del tipo de variable física a sensor (temperatura, voltaje, corriente, frecuencia, etc).

Es importante resaltar que para programar el microcontrolador no hace falta tener conocimientos de Assembler, ya que se seleccionaron dos poderosas herramientas de programación (LDmicro y Proton IDE) que facilitan la realización de programas de cualquier nivel de complejidad.

El compilador de lenguaje escalera, LDmicro, fue adaptado a las características del prototipo de PLC construido, obteniendo una versión del programa completamente en español, con teclas de acceso rápido a las funciones especiales y con parámetros de configuración fijos acordes con el PLC887; por otra parte, también se seleccionó al Proton IDE como una herramienta profesional de programación de alto nivel del PLC887, por ser un entorno de programación amigable e intuitivo para el desarrollo de programas que requieran altos niveles de complejidad.

Con la elaboración del software para el monitoreo del PLC887, cuyo canal de comunicación es el protocolo 1-Wire®, se diseñó una herramienta virtual con la cual se puede llevar a cabo una representación gráfica en tiempo real de las variables, sean digitales o analógicas, conectadas al dispositivo. El uso de la tecnología 1-Wire® permitirá utilizar una interfaz de gran alcance y bajo costo para la conexión entre un dispositivo de campo y el computador, además de construir y controlar redes con varios niveles de ramificación, permitiendo la conexión de múltiples PLC en cascada. Este proyecto contempló el diseño de la plataforma de hardware necesaria para permitir la comunicación 1-Wire® y el desarrollo de la aplicación de monitoreo, no



incluyendo la implementación del servidor de datos 1-Wire® a OPC.

Para dar a conocer las bondades presentes en el PLC887 como dispositivo de control se realizaron tres prácticas de laboratorio. Cada práctica está constituida por objetivos, pre-laboratorio, laboratorio e informe, con el propósito de entrenar a los estudiantes de Ingeniería Eléctrica de la Universidad de Carabobo, en el uso de herramientas de programación y dispositivos alternativos para el desarrollo de actividades de control.



RECOMENDACIONES

Una vez culminada la presente investigación, se plantean las siguientes sugerencias a fin de poder obtener el mayor beneficio en la aplicación del siguiente trabajo de investigación:

- Leer detalladamente los manuales elaborados en el desarrollo de este trabajo donde se explican los tipos de conexiones y las características de entrada – salida del dispositivo, antes de utilizarlo en alguna aplicación.
- Verificar cuidadosamente antes de encender el equipo la polaridad del voltaje de alimentación y del lazo de salida en corriente de 4 – 20 mA.
- Desarrollar programas que permitan utilizar la pantalla LCD y el módulo de expansión de teclado como interfaz hombre máquina para obtener e ingresar datos del proceso durante la operación del PLC.
- Desarrollar el servidor de datos 1-Wire® a OPC.
- Realizar convenios con otras universidades interesadas en utilizar estos dispositivos en el proceso de enseñanza de sus estudiantes en el área de automatización y control de procesos.
- Elaborar otras experiencias prácticas en asignaturas como; automatización, instrumentación de procesos, control de procesos y laboratorio de diseño digital, basadas en el prototipo construido.



BIBLIOGRAFÍA

- [1] V. Bravo, "Curso de: Automatas Programables (PLC)," Intelligent Electromechanical Systems Manual Teórico Práctico, sf.
- [2] R. Bliesener, et al., *Programmable Logic Controllers*. Denkendorf, Alemania: Festo Didactic GmbH & Co, 2002.
- [3] PARALLAX, Inc., "BASIC Stamp Programmable Logic Controller Package," Manual Stamp PLC, 2003.
- [4] O. Boix Aragonès, M. A. Saigí Grau, and F. Zabaleta Alaña, *Automatismos Eléctricos Programables*. Barcelona, España: Edicions UPC, 1998.
- [5] J. D. Peña, J. Gámiz Caro, A. Grau i Saltes, and H. Martínez García, *Introducción a los autómatas programables*, 6th ed. Barcelona, España: UOC, 2003.
- [6] J. M. Angulo Usategui and I. Angulo Martínez, *MICROCONTROLADORES «PIC» Diseño Práctico de Aplicaciones. Primera parte: El PIC16F84. Lenguajes PBASIC y Ensamblador.*, 6th ed. Madrid, España: McGraw-Hill, 2003.
- [7] Asamblea Nacional de la República Bolivariana de Venezuela, *Ley Orgánica de Ciencia, Tecnología e Innovación*. Caracas, Venezuela, 2005.
- [8] Presidencia de la República, *Ley de Contrataciones Públicas*. Caracas, Venezuela, 2008.
- [9] ABB Automation Products, S.A., "Advant Controller AC010 / AC31 / S500 / CP500," Manual de uso AC010, 2000.
- [10] Siemens AG, "LOGO! Manual," Manual de uso LOGO!, 2001 - 2003.
- [11] B. Linke, "Overview of 1-Wire® Technology and Its Use," Maxim Integrated Products, Inc. Notas de Aplicación, 2008.



- [12] K.-H. John and M. Tiegelkamp, *IEC 61131-3: Programming Industrial Automation Systems*. Berlin, Alemania: Springer-Verlag, 2001.
- [13] Microchip Technology Inc., *PIC16F87X: Data Sheet, 28/40-Pin 8-Bit CMOS FLASH Microcontrollers*. USA, 2001.
- [14] MikroElektronika. (2010, May) PICPLC8A PLC System. [Online]. <http://www.mikroe.com/eng/products/view/8/picplc8a-plc-system/>
- [15] MikroElektronika. (2010, May) PICPLC16B PLC System. [Online]. <http://www.mikroe.com/eng/products/view/3/picplc16b-plc-system/>
- [16] MikroElektronika. (2010, May) PICPLC16 BOX PLC System. [Online]. <http://www.mikroe.com/eng/products/view/38/picplc16-box-plc-system/>
- [17] J. Westhues. (2009, Nov.) Ladder Logic for PIC and AVR. [Online]. <http://www.cq.cx/ladder.pl>
- [18] J. Pinto. (2008, Sep.) ISA. [Online]. http://www.isa.org/Content/ContentGroups/News/2006/February24/History_of_the_PLC.htm
- [19] J. Hugh. (2008, Sep.) Automating Manufacturing Systems with PLCs. Libro Digital.
- [20] L. A. Bryan and E. A. Bryan, *Programmable Controllers Theory and Implementation*, 6th ed. USA: Industrial Text Company, 1997.
- [21] J. Balcells and J. L. Romeral, *Autómatas Programables*, 6th ed. España: Marcombo, S.A., 2000.
- [22] Crownhill Associates Ltd. (2008, Sep.) Proton BASIC Compiler - Proton Development Suite. [Online]. http://www.picbasic.org/proton_development_suite.php
- [23] Maxim. (2008, Nov.) 1-Wire Devices - Maxim. [Online]. <http://www.maxim-ic.com/products/1-wire/>



-
- [24] Maxim Inc. (2008, Nov.) Overview of 1-Wire® Technology and Its Use. [Online]. <http://www.maxim-ic.com/app-notes/index.mvp/id/1796>
- [25] Maxim. (2008, Nov.) 1-Wire Tutorial Presentation. [Online]. <http://www.maxim-ic.com/products/1-wire/flash/overview/index.cfm>
- [26] M. Barrios Y., *Manual de trabajos de Grado de Especialización y Maestría y Tesis Doctorales*. FEDUPEL, 2006.
- [27] ExpressPCB. (2010, Sep.) [Online]. <http://www.expresspcb.com/>



ANEXOS



ANEXO B

Manuales de Usuario



ANEXO C

Prácticas y Soluciones



ANEXO D

Hojas de Especificaciones Técnicas de los Circuitos Integrados