

ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE
INGENIERÍA EN MECATRÓNICA



"FABRICACIÓN DE PLC UTILIZANDO MICROCONTROLADORES PARA USO DIDACTICO
CON CARACTERISTICAS INDUSTRIALES"

TRABAJO DE INVESTIGACIÓN PARA OPTAR AL GRADO DE INGENIERO EN
MECATRÓNICA

POR

OSCAR DANIEL ARAUJO AYALA

EDGARDO ALEJANDRO GOCHEZ ZELAYA

CRISTO EDGARDO NAVARRO MASFERRER

ASESOR

EVER SIGFREDO ABREGO PREZA

JULIO 2017

SANTA TECLA, LA LIBERTAD, EL SALVADOR, C. A

ESCUELA ESPECIALIZADA EN INGENIERIA
ITCA-FEPADE
ESCUELA DE INGENIERIA EN MECATRONICA

RECTORA
ELSY ESCOBAR SANTODOMINGO

ICERRECTOR ACADÉMICO
CARLOS ALBERTO ARRIOLA MARTÍNEZ

DIRECTOR DE ESCUELA
MARIO ALFREDO MAJANO GUERRERO

COORDINADOR INGENIERIA EN MECATRONICA
JOSE MANUEL TREJO PERAZA

ASESOR DE TRABAJO DE INVESTIGACIÓN
EVER SIGFREDO ABREGO PREZA

Escuela Especializada en Ingeniería ITCA-FEPADE

Carta de Cesión de Derechos de Autor

Nosotros: OSCAR DANIEL ARAUJO AYALA DUI: 04821221-9
EDGARDO ALEJANRO GÓCHEZ ZELAYA DUI: 04383965-0
CRISTO EDGARDO NAVARRO MASFERRER DUI: 04721775-3

Estudiantes de la carrera de INGENIERIA EN MECATRONICA de la Escuela Especializada en Ingeniería ITCA-FEPADE

Manifestamos:

- 1) Que somos los autores del trabajo de investigación que lleva por título: "FABRICACIÓN DE PLC UTILIZANDO MICROCONTROLADORES PARA USO DIDACTICO CON CARACTERISTICAS INDUSTRIALES", y que en adelante denominaremos la obra, presentado como requisito de graduación de la carrera antes mencionada, el cual fue dirigida y asesorada por EVER SIGFREDO ABREGO PREZA quien desempeña como docente de la Escuela de: MECATRÓNICA en esta institución.
- 2) Que la obra es una creación dirigida original y que no infringe los derechos de propiedad intelectual, ni los derechos de publicidad, comerciales, de propiedad industrial u otros, y que no constituye una difamación, ni una invasión a la privacidad o de la intimidad, ni cualquier injuria hacia terceros.
- 3) Nos responsabilizamos ante cualquier reclamo que se le haga a la Escuela Especializada en Ingeniería ITCA-FEPADE, en este sentido
- 4) Que estamos debidamente legitimados para autorizar la divulgación de la obra mediante las condiciones de la licencia de Creative Commons.

- Reconocimiento (cc by)
- Reconocimiento-Compartir (cc by-sa)
- Reconocimiento-SinObraDerivada (cc by-nd)
- Reconocimiento-NoComercial (cc by-nc)
- Reconocimiento-NoComercial-CompartirIgual (cc by-nc-sa)
- Reconocimiento-NoComercial-SinObraDerivada (cc by-nc-nd)

De acuerdo con la legalidad vigente

- 5) Que conocemos y aceptamos las condiciones de preservación y difusión, establecidas en la Escuela Especializada en Ingeniería ITCA-FEPADE

En atención a lo antes expuesto solicitamos:

Que la obra quede depositada en las condiciones establecidas en la licencia de difusión anteriormente seleccionada, por lo tanto y en base a lo artículos 5, 7 y 8 de la Ley de Propiedad Intelectual, cedemos los derechos de autor de orden patrimonial.

Firma

Firma

Firma

Santa Tecla, a los 19 días del mes de Julio del año 2017

Índice

Contenido	Página
1. Capítulo I	
1.1. Definición o planteamiento del problema	1
1.1.1. Definición del problema.	2
1.1.2. Antecedentes.	4
1.1.3. Justificación.	7
1.2. Objetivos:	8
1.2.1. Objetivo General.	8
1.2.2. Objetivos específicos.	8
1.3. Alcances y limitaciones:	9
1.3.1. Alcances.	9
1.3.2. Limitaciones.	10
2. Capítulo II	
2.1. Marco Teórico de la Investigación.	12
2.1.1 Introducción a los PLC	13
2.1.2 Controlador lógico programable	14
2.1.3 Estructura interna de un PLC	16
2.1.3.1 Procesador	17
2.1.3.2 Memoria monitor del sistema	19
2.1.3.3 Entradas digitales	20
2.1.3.4 Entradas analógicas	20
2.1.3.5 Salidas digitales	21
2.1.3.6 Salidas analógicas	22
2.1.3.7 Fuente de alimentación	23
2.1.3.8 Interfaces	24
2.1.3.9 Unidad de programación	25
2.1.4 Estructura externa	26
2.1.4.1 Estructura compacta	26
2.1.4.2 Estructura semi modular	27

2.1.4.3 Estructura modular	28
2.1.5 Modo de funcionamiento de un PLC	28
2.1.6 Lenguajes de programación	30
2.1.6.1 Lista de instrucciones (IL)	30
2.1.6.2 Texto estructurado (ST)	31
2.1.6.3 KOP	32
2.1.6.4 FUP	32
2.1.7 PLC con microcontroladores	33
2.1.7.1 Microcontroladores	34
2.1.7.2 CPU	35
2.1.7.3 Memoria	35
2.1.7.4 Unidad de entrada salida	36
2.1.8 Arquitectura de los microcontroladores	37
2.1.9 Arquitectura de Von Neuman	38
2.2.0 Arquitectura Harvard	40
2.2.1 arquitectura del set de instrucciones	42
2.2.2 Comparativa microprocesadores y microcontroladores	45
2.2.3 Plataformas Arduino y pingüino	
2.2.3.1 Placa pingüino	47
2.2.3.2 Estructura de Arduino (ATMEL AVR)	49
2.2.3.3 Estructura de Arduino ATmega 328/P)	49
2.2.4 Criterio de diseño para sistemas automatizados	53
2.2.4.1 Entorno industrial para sistemas automatizados	54
2.2.4.2 Compatibilidad electromagnética	54
2.2.4.3 Segregación de circuitos	55
2.2.4.4 Diseños de sistemas de alimentación	55
2.2.4.5 Estructura de masas	55
2.2.4.6 Estructura de alimentación	56
2.2.4.7 Entradas ,salidas y protección	56
2.2.4.8 Diseño del oscilador del reloj	56
2.2.4.9 Desacoplo de los circuitos integrados	56

2.2.5 Normativa IEC 61131	57
2.2.5.1 Finalidad de la normativa	57
2.2.5.2 Condiciones de servicio normales	58
2.2.5.3 Requisitos eléctricos	58
2.2.5.4 Requisitos mecánicos	59
2.2.6 Normativa para diseño de PLC y equipos de automatización	59
2.2.7 Estructura básica de un PLC con microcontroladores	62
2.2.7.1 Fuentes de alimentación DC/DC	62
2.2.7.2 Definición de entradas a AC	65
2.2.7.3 Definición de entradas a DC	65
2.2.7.4 Acondicionamiento de entradas digitales	65
2.2.7.5 Diseño con aislamiento galvánico	66
2.2.7.6 Elementos de aislamiento de potencial-opto acopladores	67
2.2.7.7 Filtro pasa bajas en entradas DC	67
2.2.8.0 Diseño de entradas analógicas	69
2.2.8.1 Procesamiento digital de señales analógicas	69
2.2.8.2 Resolución y lectura de señales analógicas	71
2.2.8.3 Filtrado de entradas analógicas	72
2.2.8.4 Diseño de salidas digitales	77
2.2.8.5 Salidas a relé	77
2.2.8.6 salidas transistorizadas	78
2.2.9 Software libre	79
2.2.9.0 Que es el software libre	79
2.2.9.1 Software libre comercial	80
2.2.9.2 Libertad de ejecutar el programa como se desee	81
2.2.9.3 Copyleft	83
2.2.9.4 Empaquetamiento y distribución	83
2.2.9.5 Reglas admisibles	84
2.2.9.6 Consideraciones legales	84
2.2.9.7 Licencias basadas en contrato	85

2.2.9.8 Por qué proporcionar software libre para programar un PLC	86
2.2.9.9 Costos de adquisición	86
2.3.0 Software libre de programación de PLC con microcontroladores	87
2.3.1 Programación escalera en microcontroladores	88
2.3.2 Funciones de softwares y plataformas compatibles	89
2.4.0 Software (Waltech Ladder Maker)	91
2.4.1.1 Estructura de software	92
2.4.1.2 Interfaz de línea de comandos	93
2.4.1.3 Instrucciones disponibles en software	94
2.4.1.4 Observaciones del software	96
CAPITULO III	
3.1 METODOLOGIA DE LA INVESTIGACION	100
3.1.1 Pasos a seguir para desarrollar una metodología	100
3.1.2 Metodología científica aplicada tecnológica	101
3.1.3 Metodología participativa	102
3.1.3.1 Estudio de casos	103
3.1.3.2 Estudio etnográfico	103
3.1.4 Encuesta proyecto PLC software y hardware libre	104
3.1.5 Investigación acción	105
3.1.6 Enfoque cualitativo	105
CAPITULO IV	
4.1 ALCANCES-ANALISIS DE RESULTADOS	107
4.1.1 Resultados que se esperan obtener con el proyecto	108
4.1.2 Resultados obtenidos de aplicar metodología participativa y enfoque cualitativo	109
4.1.3 Resultados obtenidos de aplicar metodología científica aplicada y metodología acción	118
4.1.4 Diseño y cálculos del micro PLC	118
4.1.5 Regulador de voltaje 24vdc a 12vdc	119

4.1.5.1 Descripción	119
4.1.5.2 Diseño esquemático	119
4.1.5.3 Calculos de diseño	120
4.1.6 Entradas digitales	121
4.1.6.1 Descripción	121
4.1.6.2 Diseño esquemático	122
4.1.6.3 Calculos de diseño	122
4.1.7 Entradas analógicas	124
4.1.7.1 Descripción	124
4.1.7.2 Diseño esquemático	125
4.1.7.3 Calculos de diseño	125
4.1.8 Circuito base arduino NANO	127
4.1.8.1 Descripción	127
4.1.8.2 Diseño esquemático	127
4.1.8.3 Calculos de diseño	127
4.1.9 PCB Regulador de voltaje 24vdc a 12vdc	128
4.1.10 Esquema circuito base arduino	128
4.1.11 PCB circuito base arduino	129
4.1.12 Botton PCB circuito base arduino	129
4.1.13 PCB circuito interfaz IN/OUT	130
CAPITULO V	
5.1 CONCLUSIONES Y RECOMENDACIONES	132
Bibliografía	135
Sitiografia	137
Glosario	138
Anexos	140

INDICE DE IMÁGENES

CAPITULO II	
Imagen 2.1 Ciclo de accionamiento de un PLC	14
Imagen 2.2 Estructura interna de un PLC	16
Imagen 2.3 Tipos de señales	19
Imagen 2.4 Estructura compacta de un PLC	27
Imagen 2.5 Estructura modular de un PLC	28
Imagen 2.6 Ciclo de trabajo de un PLC	29
Imagen 2.7 Sintaxis de programación Lista de Instrucciones	30
Imagen 2.8 Sintaxis de programación SCL	31
Imagen 2.9 Programación en lenguaje de escalera	32
Imagen 2.10 Programación en Diagrama de funciones	33
Imagen 2.11 Elementos principales de un microprocesador	34
Imagen 2.12 Arquitectura de Von Neuman	38
Imagen 2.13 Secuencia para coordinar el procesamiento de datos	40
Imagen 2.14 Arquitectura Harvard generalizada	40
Imagen 2.15 Diagrama secuencial de Sistemas de Instrucciones	43
Imagen 2.16 Placa Pingüino	47
Imagen 2.17 Estructura interna de PIC 18f2550	48
Imagen 2.18 Tarjeta Arduino NANO	49
Imagen 2.19 Diseño de Circuito con ATmega 328P en Arduino	51
Imagen 2.20 Circuito regulador de voltaje Arduino NANO	51
Imagen 2.21 Procesador de Comunicación UART/USB	52
Imagen 2.22 Micro PLC de Allen Bradley - PICO Controller	53
Imagen 2.23 Etiqueta CE de certificación	62
Imagen 2.24 Circuito Típico de un regulador lineal LM317	64
Imagen 2.25 Comportamiento de la señal de salida sin amortiguamiento	68
Imagen 2.26 Tipos de PDS	70
Imagen 2.27 Formatos Aritméticos de PDS	70
Imagen 2.28 Toma de valores de una frecuencia especifica	73

Imagen 2.29 Graficas de problemas de Aliasing	74
Imagen 2.30 Comportamiento de un filtro pasa bajas	75
Imagen 2.31 Diseño de entrada analógica configurable	76
Imagen 2.32 Diseño de salidas a relé	77
Imagen 2.33 Diseño de salidas Open Drain a transistor	78
Imagen 2.34 Encabezado de Interfaz de Software Waltech Ladder Maker	93
Imagen 2.35 Interfaz de línea de comandos	93
CAPITULO IV	
Imagen 4.1 Circuito LM317	119
Imagen 4.2 Circuito de entradas digitales	122
Imagen 4.3 Ratio de Transferencia de Corriente vs. Corriente Directa	123
Imagen 4.4 Circuito de entradas analógicas	125
Imagen 4.5 Circuito base de arduino nano	127
Imagen 4.6 Diseño de PCB's	128
Imagen 4.7 Esquema Circuito Base Arduino	128
Imagen 4.8 TOP PCB Circuito Base Arduino	129
Imagen 4.9 Bottom PCB Circuito Base Arduino	129
Imagen 4.10 PCB CIRCUITO INTERFAZ IN/OUT - TOP	130
Imagen 4.11 PCB CIRCUITO INTERFAZ IN/OUT - BOTTOM	131

INDICE DE TABLAS

TABLA 1 Comparativa entre microprocesadores y microcontroladores	45
Tabla 2 Comparación de funciones de softwares y sus plataformas compatibles	89
Tabla 3 Descripción de las instrucciones del software Waltech Ladder Maker	94
Tabla 4 Barra de edición en programación escalera	96
Tabla 5 Encuesta Proyecto: PLC - Software y Hardware Libre	104

CAPITULO 1

PLANTEAMIENTO DEL PROBLEMA

DEFINICIÓN DEL PROBLEMA

La automatización actualmente está evolucionando rápidamente y cada vez incrementa más la demanda por parte de las empresas de contratar personal capacitado, que cumpla con las competencias y destrezas para poder controlar sus procesos utilizando otras herramientas o equipos que tengan similares características, y una muy buena funcionabilidad comparados con equipos costosos (gama media y alta), todo esto con el fin de generar un ahorro económico, tiempos de producción, calidad, etc., en sus plantas o líneas de producción.

El análisis que hemos desarrollado en ITCA FEPADE, es que la institución en su currícula de desarrollo del estudiante de técnico Mecatrónica, lo guía a adquirir competencias que le permitan aplicarse como analista y desarrollador de sistemas automatizados, utilizando autómatas programables. En esta perspectiva, se plasma la necesidad de analizar el uso eficiente de las herramientas y equipos, utilizadas para lograr el objetivo anteriormente mencionado. Tal análisis nos arroja que, la introducción que se le da al alumno para desarrollar sus competencias de lógica aplicada a la programación de algoritmos, se le imparte en la materia de "Lógica de Programación", que va orientada a uso del programa DFD para el diseño de flujogramas.

Sin embargo, como propuesta que resultaría viable sería, desarrollar un autómata programable de bajo costo, con microcontroladores, basándose en la programación en lenguaje KOP con software de Uso libre, como un concepto que se puede implementar en los últimos temas de la materia de lógica de programación, para la implementación real de algoritmos y el uso del lenguaje KOP como introducción a las materias de programación de PLC, creando así nociones y conceptos básicos necesarios para la programación en este lenguaje.

En la actualidad en nuestro país no se cuenta con un dispositivo de fabricación local que cumpla con las características y funciones de un PLC, que pudiera ser una alternativa económica y fiable para el desarrollo de aplicaciones básicas en la industria. En el marco de este otro problema se pretenderá implementar el uso de hardware y software libre lo que significa que no serían equipos que requieran comprar y/o renovar licencias cada cierto tiempo (6 meses, un año, etc.), pero que estos equipos a pesar de su diseño básico, y de fabricación local, sean eléctricamente robustos, electromagnéticamente compatibles, y tengan funcionabilidad aceptable.

Todo esto nos permitirá introducir los conceptos y la integración de las herramientas de diseño electrónico, y de software necesarias para fabricar autómatas programables básicos, que ayuden a la innovación y competitividad tecnológica científica de ITCA FEPADE, en el desarrollo de prototipos para aplicaciones industriales y de uso didáctico.

ANTECEDENTES

En nuestro país El Salvador no se ha abordado una investigación específica en el uso, fabricación e implementación industrial de autómatas programables basados en microcontroladores, mediante hardware libre y software de código abierto, sin embargo, si se han desarrollado sistemas dedicados o embebidos en aplicaciones textiles, imprenta, codificación, de fabricación local con uso de microcontroladores por parte de departamentos de investigación de universidades y empresas privadas de mantenimiento y diseño de circuitos electrónicos, mediante lenguajes de bajo nivel en Protocolos SPI(Serial Peripheral Interface).

A nivel de investigaciones previas por parte de las universidades solamente hemos podido encontrar una Tesis de la Universidad Nacional para Grado de Ingeniería Eléctrica, con el tema: "Diseño y construcción de un (PLC) Control Lógico Programable basado en la tecnología del microcontrolador", sin embargo esta tesis solo se centró en la implementación de un microcontrolador PIC 16F877p emulando un PLC, que no contaban con circuitos que en su diseño tomaran en cuenta una proyección al entorno industrial. Y utilizando un software de programación que interpretaba los estados lógicos de las salidas respecto a las entradas mediante tablas de verdad, que simbolizaban las posibles combinaciones de las entradas.

En la investigación se encontraron tres softwares de código abierto, ya compilados en diferentes plataformas de programación, lo que nos permitió identificar que ya existía un precedente con el desarrollo de proyectos de control de sistemas en general, basados en microcontroladores.

Sin embargo, no todos son sencillos, algunos muy completos, los cuales piden configurar otros elementos de otras fuentes para trabajar (Pidgets) o bien modificar y recargar el bootloader (firmware del microcontrolador).

Además del software en países desarrollados, ya se dieron cuenta de los alcances y la potencia del uso de microcontroladores en el diseño de PLC's industriales y se dispusieron a la tarea de fabricarlos, como por ejemplo las empresas:

1. **Industrial Shields**, empresa española dedicada al diseño, producción y comercialización de productos basándose en hardware Open Source, como por ejemplo PLC's basados en placa Arduino (ATmega328), utilizando el IDE de arduino como plataforma de programación y pantallas HMI basadas en Raspberry PI.

2. **Controllino**, empresa alemana, dedicada a la fabricación de PLC's industriales basados en ATmega2560, utilizando diferentes entornos como el IDE de arduino, Programino, LabView o ATMEL Studio como software de programación.

Sin embargo, ninguna de estas empresas ofrece una alternativa específica de programación en lenguajes como LAD (escalera), FBD (Bloques de funciones), SFC (Funciones secuenciales), los que comúnmente se utilizan para la enseñanza de programación de PLC's y aplicaciones industriales en general. Esto es debido a que en lenguajes de alto nivel como el ST (Texto estructurado), se le saca mejor partida al hardware según la aplicación, mediante el uso de técnicas de programación, funciones, como la programación orientada a objetos, etc., que nacen de los lenguajes computacionales.

Todos estos datos y antecedentes del tema, nos da el preámbulo y los contextos en que se ha venido desarrollando el tema, y que nos generar el conocimiento básico y esencial que se necesita para la implementación y fabricación de autómatas programables basados en microcontroladores, particularmente en placa de Arduino con Micro ATmega328.

JUSTIFICACIÓN

Nuestro estudio de investigación se realiza con el primer propósito de demostrar que se puede desarrollar un prototipo empleando microcontroladores que nos permita dar un aporte al enriquecimiento didáctico a los estudiantes haciendo uso de un microPLC de fabricación local, utilizando hardware (microcontroladores de uso libre), brindando una herramienta de aprendizaje que facilite el uso de lenguajes de programación KOP, ya no solo de forma general aplicando la plataforma de Arduino, sino también aplicando los algoritmos en un PLC básico como introducción, que les facilite posteriormente en las materias de Programación de PLC, en la familiarización de los conceptos, técnicas y el manejo eléctrico adecuado de estos dispositivos, además de los criterios de diseño electrónicos de los mismos. Pudiendo ser esta una propuesta económica y viable, en su implementación a largo plazo; de manera que localmente en ITCA-FEPADE se pueda ejecutar y desarrollar más detalladamente a futuro estos sistemas.

Y al mismo tiempo junto con la parte didáctica poder diseñar un microPLC que posea características eléctricas y electrónicas similares a las de los PLC industriales y que puedan contar con lenguajes de programación estandarizados que puedan ser ejecutados por sistemas operativos comunes. Logrando con todo esto, el desarrollo de conceptos y la aplicación de tecnología en el tema de diseño de sistema y/o prototipos electrónicos de esta forma incentivar el desarrollo científico de estos sistemas, que a un futuro puedan ser desarrolladas con estándares y certificaciones internacionales.

OBJETIVOS

General

Desarrollar y crear un microPLC utilizando un microcontrolador en placa de Arduino que posea características de un PLC industrial y que sirva como equipo didáctico a los estudiantes de ITCA-FEPADE.

Específicos

- Implementar un entorno de desarrollo integrado (IDE) de uso libre, basado en lenguaje grafico KOP (esquema de contactos), para la lógica programación del microcontrolador, con la capacidad de instalarse en los sistemas operativos más comunes en la actualidad.
- Desarrollar la adaptación de un sistema o plataforma de microcontrolador Arduino como un sistema PLC robusto electricamente y práctico, para aplicaciones didácticas
- Diseñar un manual que aporte al objetivo didáctico del proyecto, permitiéndole al estudiante, el uso del software y una buena manipulación de hardware del microPLC a diseñar, para su implementación en los laboratorios de ITCA FEPADE.
- Diseñar un prototipo que pueda reflejar ahorro en costos de fabricación, al utilizar un microcontrolador, programándolo con software libre en lenguaje grafico KOP y que no requiera la compra de licencias para su uso.

ALCANCES

- ✓ Se busca con la investigación que con nuestro Micro PLC que ITCA-FEPADE adquiriera una herramienta didáctica a las carreras afines para la enseñanza de los conceptos básicos y las aplicaciones que se pueden desarrollar en un PLC, utilizando un lenguaje estándar de programación (KOP).

- ✓ Desarrollar la investigación en la fabricación y diseño de autómatas programables a la comunidad técnica científica de nuestro país para motivar a desarrollar prototipos más avanzados y softwares más completos con base en la tecnología ya existente.

- ✓ Una de las características con las que contara nuestro micro PLC es que podrá contar con un mínimo de 5 entradas digitales ópticamente aisladas galvánicamente de 24 a 5v ,7 salidas digitales a Relé con capacidad de manejar cargas inductivas de hasta 20Amp y 4 entradas analógicas normalizadas de 4-20 mA.

- ✓ Para facilitar el aprendizaje y uso correcto del software de programación, así como la operación del PLC se proporcionara un manual didáctico en donde se dará a conocer a los estudiantes de forma gráfica el uso de los elementos más importantes del software como simbología de contactos.

- ✓ Se tendrá la capacidad de contar con características electrónicas de un PLC industrial y en futuro poder desarrollar el prototipo para que trabaje en entornos industriales con un grado de compatibilidad electromagnética que le permita funcionar de forma estable y segura en el proceso a controlar.

LIMITACIONES

- ✓ El tiempo es un factor crítico, para la fabricación del PLC por lo que no se puede llegar a fabricar un equipo estandarizado con todas las "normas" que aplican a un PLC industrial con respecto a su CEM (Compatibilidad Electro Magnética), tener una mayor capacidad de control de periféricos y obtener acceso a conectarse a distintas redes de comunicación con procesadores de comunicación.
- ✓ La variedad de diferentes lenguajes de programación que se podrá utilizar para crear diferentes programas y rutinas estará determinado por el software de uso libre que se seleccione.
- ✓ El software libre seleccionado para la fabricación del PLC determinara el tipo de "instrucciones posibles" a utilizar en la programación así como también la configuración física de los pines de entrada y salida.
- ✓ El micro PLC será una versión que operara con alimentación externa de 24v no auto configurable 12/24vdc.
- ✓ No poseerá un procesador de comunicaciones (CP) para conexión a red sea Ethernet o serial RS232/485.
- ✓ No será posible integrar módulos de expansión para incrementar el número de entradas y salidas; Estará limitado a la configuración predeterminada del software de programación en escalera.

- ✓ El tamaño del programa estará limitado por la memoria Flash del micro controlador: 32KBytes de los cuales 2KBytes son determinados para el Bootloader.

- ✓ No será posible la fabricación de un chasis estandarizado por norma DIN debido a las limitaciones de fabricación; y por precio y costo de importación desde Italia no es posible emplear un chasis genérico que nos permita integrar la placa electrónica del micro PLC.

- ✓ No se cuentan con las herramientas de fabricación de circuitos CMD para realizar un diseño con elementos compactos que permitan reducir el tamaño del prototipo y así homogenizarlo con los sistemas de uso comercial.

- ✓ No se cuenta con un taller o centro mecánico que nos permita crear el diseño y la fabricación de la carcasa que contendrían el PLC, con los estándares internacionales.

CAPITULO 2

MARCO TEORICO DE LA INVESTIGACIÓN

INTRODUCCIÓN A LOS PLCS

El Controlador Lógico Programable (PLC) surgió como una solución al control de circuitos complejos de automatización, tiene su origen en la industria de fabricación de coches. A finales de los años 60 la industria busco soluciones más eficientes para poder remplazar los sistemas de control que estaban basados en circuitos eléctricos como los relés, interruptores y otros componentes.

En 1968 la empresa General Motors recibió diferentes propuestas con el fin de sustituir los sistemas cableados de relés, Bedford Associates propuso la propuesta ganadora, como resultado surgió el primer PLC designado MODICON 084 (Modular Digital Controller) llamado así porque era el proyecto número 84 de la compañía, de esa manera la empresa comenzó la fabricación, venta y mantenimiento de estos productos, Una de las personas que trabajaron en ese proyecto fue Dick Morley, quien es considerado como el padre del PLC

En 1973 apareció el primer sistema de comunicación Modbus de Modicon que les daba la capacidad a los PLC poder estar alejados de la máquina que estaban controlando, pero la falta de estandarización debido al constante cambio en la tecnología hizo que la comunicación se tornara difícil.

En 1980 e intentó estandarizar la comunicación entre PLCs con el protocolo de automatización de manufactura de la General Motors (MAP).en esta época se logró reducir el tamaño de lo PLC y la forma de programación se realizaba mediante computadoras personales.

Los primeros PLC fueron programados en lenguaje llamado listado de instrucciones en donde las órdenes del control se transmitían al procesador por medio de un listado secuencial de códigos en lenguajes de máquina.

Posteriormente apareció el lenguaje Ladder conocido también como diagrama de escaleras, parecido mucho a un diagrama esquemático de la lógica de relés

En 1990 se mejoraron y se introdujeron nuevos protocolos. El estándar IEC 1131-3 intentó combinar los lenguajes de programación de los PLC en un solo estándar internacional. Ahora se tienen PLC's que se programan en función de diagrama de bloques, listas de instrucciones, lenguaje C etc.

En la actualidad los PLC's pueden realizar operaciones aritméticas, manejar señales analógicas para realizar estrategias de control, tales como los controladores proporcional integral derivativo (PID). Además poseen la capacidad de comunicarse con otros controladores y computadoras en redes de área local, y son una parte fundamental de los modernos sistemas de control distribuido.

CONTROLADOR LOGICO PROGRAMABLE

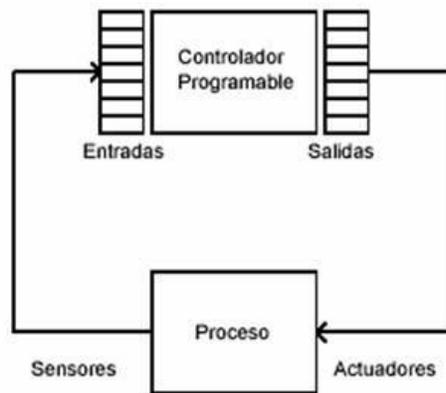


Imagen 2.1: Ciclo de accionamiento de un PLC

Un Controlador Lógico Programable (PLC) es un dispositivo electrónico operado digitalmente mediante una unidad de procesamiento, que posee las herramientas necesarias, tanto de hardware como de software, para comprobar el estado de sensores y controlar dispositivos externos como actuadores a través de un programa elaborado por el usuario.

Utiliza una memoria para el almacenamiento interno de instrucciones con el fin de implementar funciones específicas tales como lógicas secuenciales, registro y control de tiempos, conteo y operaciones aritméticas para controlar por medio de entradas/salidas digitales o analógicas varios tipos de máquinas o procesos.

El autómata programable satisface las exigencias tanto de procesos continuos como discontinuos. Regula presiones, temperaturas, niveles y caudales así como todas las funciones asociadas de temporización, cadencia, conteo y lógica. También incluye una tarjeta de comunicación adicional, el autómata se transforma en un poderoso satélite dentro de una red de control distribuida

Sus reducidas dimensiones, su facilidad de su montaje, la posibilidad de almacenar programas para su posterior utilización, la modificación o alteración de los mismos, etc., hace que su eficacia se aprecie fundamentalmente en procesos en que se producen necesidades tales como:

- Espacio reducido.
- Procesos de producción periódicamente cambiantes.
- Procesos secuenciales.
- Maquinaria de procesos variables.
- Instalaciones de procesos complejos y amplios.

- Comprobación de programación centralizada de las partes del proceso.
- Maniobra de máquinas e instalaciones.
- Señalización y control

Ventajas de los controladores lógico programables

- Una automatización relativamente económica
- Una complejidad relativamente sencilla
- Permite la grabación, copia e impresión de programas ya sea desde un pc o desde los módulos de memoria
- Disponen de salidas a relé con una gran capacidad de corte
- El mantenimiento es nulo
- Protección del programa de usuario **(Pulido, 2004, p. 3)**

Estructura interna de un PLC

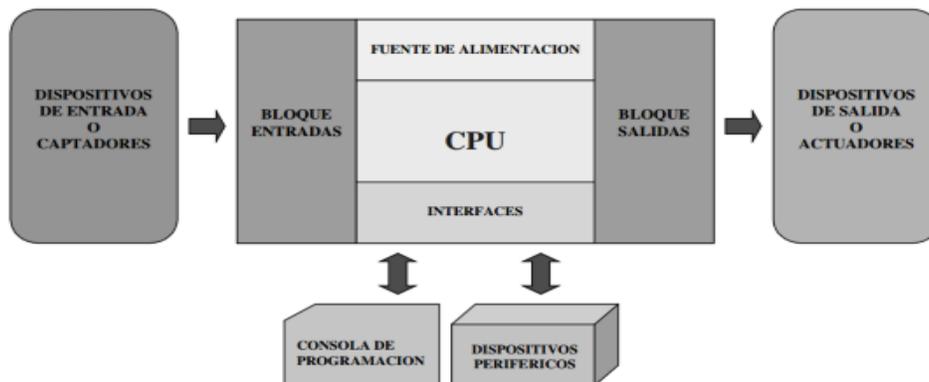


Imagen 2.2: Estructura interna de un PLC

El autómata está constituido por diferentes elementos, pero tres son los básicos:

- CPU (Unidad Central de Procesamiento puede ser microprocesador o Microcontrolador)
- Entradas (Digitales, Analógicas)
- Salidas (Digitales y Analógicas)

La CPU es la parte inteligente del sistema, es el encargado de Interpretar las instrucciones del programa de usuario y consulta el estado de las entradas. Dependiendo de dichos estados y del programa, ordena la activación de las salidas deseadas.

La CPU está constituida por los siguientes elementos:

- Procesador
- Memoria monitor del sistema
- Circuitos auxiliares

Procesador

Es una computadora que ejecuta un programa para desarrollar las operaciones específicas en un diagrama de escalera o bien un conjunto de ecuaciones booleanas, el procesador desarrolla operaciones aritméticas y lógicas sobre los datos de las variables de entrada y determinar el dato de las variables de salida **(Harper, 1999, p. 93)**

Está constituido por el microprocesador, el reloj (generador de onda cuadrada) y algún chip auxiliar que comúnmente coordina funciones de comunicación buses internos de comunicación o de bastidor en el caso de los PLCs modulares.

El microprocesador es un circuito integrado (chip), que realiza una gran cantidad de operaciones, que podemos agrupar en:

- Operaciones de tipo lógico.
- Operaciones de tipo aritmético.
- Operaciones de control de la transferencia de la información dentro del autómata.

Para que el microprocesador pueda realizar todas estas operaciones está dotado de unos circuitos internos que son los siguientes:

Circuitos de la unidad aritmética y lógica o ALU: Es la parte del μp donde se realizan los cálculos y las decisiones lógicas para controlar el autómata.

Circuitos de la unidad de control (UC) o Decodificador de instrucciones: Decodifica las instrucciones leídas en memoria y se generan las señales de control

Acumulador: Es la encargada de almacenar el resultado de la última operación realizada por el ALU.

Flags: o indicadores de resultado, que pueden ser consultados por el programa.

Contador de programa: Encargada de la lectura de las instrucciones de usuario.

Sin embargo no en todos los casos los PLC cuentan con un microprocesador como unidad de procesamiento, esto dependerá de los alcances y capacidades de procesamiento de las CPU, en algunos casos un PLC puede contar con un microcontrolador como unidad central de procesamiento, pero comúnmente esto ocurre con los relés inteligentes con bajas capacidades.

Memoria monitor del sistema

Es una memoria de tipo ROM, y además del sistema operativo del autómata contiene las siguientes rutinas, incluidas por el fabricante.

- Inicialización tras puesta en tensión o reset.
- Rutinas de test y de respuesta a error de funcionamiento.
- Intercambio de información con unidades exteriores
- Lectura y escritura en las interfaces de E/S

La sección de entradas mediante el interfaz, adapta y codifica de forma comprensible para la CPU las señales procedentes de los dispositivos de entrada o captadores.

Hay dos tipos de entradas:

- Entradas digitales
- Entradas analógicas

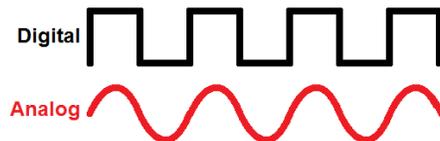


Imagen 2.3: Tipos de señales

La sección de salida también mediante interfaz trabaja de forma inversa a las entradas, es decir, decodifica las señales procedentes de la CPU, y las amplifica y manda con ellas los dispositivos de salida o actuadores como lámparas, relés... aquí también existen unos interfaces de adaptación a las salidas de protección de circuitos internos.

Hay dos tipos de salidas:

- Salidas digitales
- Salidas analógica

Entradas digitales

Los módulos de entrada digitales permiten conectar al autómata, captadores de tipo todo o nada como finales de carrera, pulsadores etc.

Los módulos de entrada digitales trabajan con señales de tensión, por ejemplo cuando por una vía llegan 24 voltios se interpreta como un "1" y cuando llegan cero voltios se interpreta como un "0"

El proceso de adquisición de la señal digital consta de varias etapas

- Protección contra sobretensiones
- Filtrado
- Puesta en forma de la onda
- Aislamiento galvánico o por opto acoplador.

Entradas analógicas

Son aquellas que detectan valores de tensión o corriente para producir eventos por comparación cuando se alcanzan los prefijados en el programa de usuario **(Martin, p. 268)**

Estándares de tensión

- De 0 a 10vcc
- De -10 a +10vcc
- De 2 a 10vcc

Estándares de corriente

- De 4 a 20mA
- De 0 a 20mA
- De 1 a 5mA
- De 0 a 5mA

Los módulos de entrada analógicas permiten que los autómatas programables trabajen con actuadores de mando analógico y lean señales de tipo analógico como pueden ser la temperatura, la presión o el caudal.

Los módulos de entradas analógicas convierten una magnitud analógica en un número que se deposita en una variable interna del autómata. Lo que realiza es una conversión A/D, puesto que el autómata solo trabajar con señales digitales. Esta conversión se realiza con una precisión o resolución determinada (número de bits) y cada cierto intervalo de tiempo (periodo muestreo).

Los módulos de entrada analógica pueden leer tensión o intensidad. El proceso de adquisición de la señal analógica consta de varias etapas:

- Filtrado
- Conversión A/D
- Memoria interna

Salidas digitales

Las funciones digitales son aquellas que funcionan con todo o nada, es decir en los bornes de salida existe o no existe tensión y pueden ser de 2 tipos (relé, transistor) **(Pulido, 2004, p. 102)**

El valor binario de las salidas digitales se convierte en la apertura o cierre de un relé interno del autómata en el caso de módulos de salidas a relé.

En los módulos estáticos (bornero), los elementos que conmutan son los componentes electrónicos como transistores o triacs, y en los módulos electromecánicos son contactos de relés internos al módulo.

Los módulos de salidas estáticas al suministrar tensión, solo pueden actuar sobre elementos que trabajan todos a la misma tensión, en cambio los módulos de salida electromecánicos, al ser libres de tensión, pueden actuar sobre elementos que trabajen a tensiones distintas.

El proceso de envío de la señal digital consta de varias etapas:

Puesta en forma

- Aislamiento
- Circuito de mando (relé interno)
- Protección electrónica
- Tratamiento cortocircuitos

Salidas analógicas

Los módulos de salida analógica permiten que el valor de una variable numérica interna del autómatas se convierta en tensión o intensidad.

Lo que realiza es una conversión D/A, puesto que el autómatas solo trabaja con señales digitales.

Esta Conversión se realiza con una precisión o resolución determinada (número de bits) y cada cierto intervalo de tiempo (periodo muestreo).

Esta tensión o intensidad puede servir de referencia de mando para actuadores que admitan mando analógico como pueden ser los variadores de velocidad, las etapas de los tiristores de los hornos, reguladores

de temperatura. Permitiendo al autómata realiza funciones de regulación y control de procesos continuos.

El proceso de envío de la señal analógica consta de varias etapas:

- Aislamiento galvánico
- Conversión D/A
- Circuitos de amplificación y adaptación
- Protección electrónica de la salida

Como hemos visto las señales analógicas sufren un gran proceso de adaptación tanto en los módulos de entrada como en los módulos de salida. Las funciones de conversión A/D y D/A que realiza son esenciales. Por ello los módulos de E/S analógicos se les consideran módulos de E/S especiales.

Con las partes mencionadas podemos decir que tenemos un autómata pero para que sea operativo son necesarios otros elementos tales como:

- Fuente de alimentación
- Interfaces
- La unidad o consola de programación
- Los dispositivos periféricos

La fuente de alimentación

Proporciona las tensiones necesarias para el funcionamiento de los distintos circuitos del sistema.

La alimentación a la CPU puede ser de continua a 24 Vcc, tensión muy frecuente en cuadros de distribución, o en alterna a 110/220 Vca. En cualquier caso es la propia CPU la que alimenta las interfaces conectadas a través del bus interno.

La alimentación a los circuitos E/S puede realizarse, según tipos, en alterna a 48/110/220 Vca o en continua a 12/24/48 Vcc.

La fuente de alimentación del autómeta puede incorporar una batería tampón, que se utiliza para el mantenimiento de algunas posiciones internas y del programa usuario en memoria RAM, cuando falla la alimentación o se apaga el autómeta.

Interfaces

Interfaces

En el control de un proceso automatizado, es imprescindible un dialogo entre operador-máquina junto con una comunicación entre la máquina y el autómeta, estas comunicaciones se establecerán por medio del conjunto de entradas y salidas del citado elemento.

Los autómetas son capaces de manejar tensiones y corrientes de nivel industrial, gracias a que disponen un bloque de circuitos de interfaz de E/S muy potente, que les permite conectarse directamente con los sensores y accionamientos del proceso.

De entre todos los tipos de interfaces que existen, las interfaces específicas permiten la conexión con elementos muy concretos del proceso de automatización.

Se pueden distinguir entre ellas tres grupos bien diferenciados:

- Entradas / salidas especiales.
- Entradas / salidas inteligentes.

- Procesadores periféricos inteligentes.

Las interfaces especiales del primer grupo se caracterizan por no influir en las variables de estado del proceso de automatización. Únicamente se encargan de adecuar las E/S, para que puedan ser inteligibles por la CPU, si son entradas, o para que puedan ser interpretadas correctamente por actuadores (motores, cilindros, etc.), en el caso de las salidas.

Las del segundo grupo admiten múltiples modos de configuración, por medio de unas combinaciones binarias situadas en la misma tarjeta. De esta forma se descarga de trabajo a la unidad central, con las ventajas que conlleva.

Los procesadores periféricos inteligentes

Son módulos que incluyen su propio procesador, memorias y puntos auxiliares de entrada / salida.

Estos procesadores contienen en origen un programa especializado en la ejecución de una tarea concreta, a la que le basta conocer los puntos de consigna y los parámetros de aplicación para ejecutar, de forma autónoma e independiente de la CPU principal, el programa de control

Unidad de programación

Es el conjunto de medio hardware y software mediante los cuales el programador introduce y depura sobre las secuencias de instrucciones (en uno u otro lenguaje) que constituyen el programa a ejecutar.

Los autómatas poseen dos sistemas de conexión diferentes a la entrada y salida, que se basan en el aislamiento de la señal del resto del circuito para no dañar los circuitos internos del PLC.

El módulo de entrada conecta las entradas físicas del PLC al resto del sistema. Cada terminal está eléctricamente aislada de los componentes electrónicos internos a través del uso de opto acopladores. De esta forma, podemos pasar el estatus de la entrada exterior (On/Off) con un diodo emisor de luz y un fototransistor. Es una medida preventiva para efectos magnéticos y altos voltaje

El módulo de salidas contiene interruptores (transistores o relés) activados por la CPU para conectar dos terminales y así permitir el paso de corriente a los circuitos externos.

Estructura externa

El término estructura externa o configuración externa de un autómata programable industrial se refiere al aspecto físico exterior del mismo, bloques o elementos en que está dividido.

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

- Estructura compacta
- Estructura semimodular (Estructura Americana)
- Estructura modular (Estructura Europea)

Estructura compacta.

Todos los elementos necesarios están agrupados y se integran en único dispositivo, son ideales para pequeñas aplicaciones con pocas entradas y salidas **(Cerde, 2014, p. 320)**

Este tipo de autómatas se distingue por presentar en un solo bloque todos sus elementos, esto es, fuente de alimentación, CPU, memorias, entradas/salidas, etc.

Son los PLC de gama baja o micro-PLCs 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.



Figura 2.4: Estructura compacta de un PLC

Ventajas de una estructura compacta

- Los autómatas compactos son más económicos dentro de su variedad.
- estos autómatas ocupan un menor espacio gracias a su construcción compacta.
- la programación de este tipo de autómatas es básica
- No requiere conocimientos profundos para su selección.
- Gracias a su estructura permiten realizar una fácil instalación. Soportan contingencias extremas de funcionamiento tales como, temperaturas $<60^{\circ}\text{C}$, fluctuaciones de tensión, vibraciones mecánicas, humedad, etc.

Estructura Semi modular.

Se caracterizan por separar las E/S del resto del Controlador Lógico Programable, 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 unidades de E/S .

Estructura modular

Su característica principal es la de que existe un módulo para cada uno de los diferentes elementos que componen el PLC como puede ser una fuente de alimentación, CPU, E/S, etc.

La sujeción de los mismos se hace por riel DIN, placa perforada o sobre RACK, en donde van alojado el BUS externo de unión de los distintos módulos que lo componen.

Son los PLC de gama alta los que suelen tener una estructura modular, que permiten una gran flexibilidad en su constitución.



Figura 2.5: Estructura modular de un PLC

MODO DE FUNCIONAMIENTO DE UN PLC

El proceso de funcionamiento de un autómata es el comportamiento que se quiere replicar. La importancia de este proceso es importante para comprender como funciona un PLC y así poder simular su modo de funcionamiento

La CPU se ha previsto para que se ejecute cíclicamente una serie de tareas, dicha ejecución se denomina ciclo del autómata (figura 2), durante el cual, la CPU ejecuta la mayoría de las tareas siguientes

- Lee las entradas
- Ejecuta el programa de usuario
- Procesa las peticiones de comunicación
- Ejecuta un autodiagnóstico
- Escribe las salidas

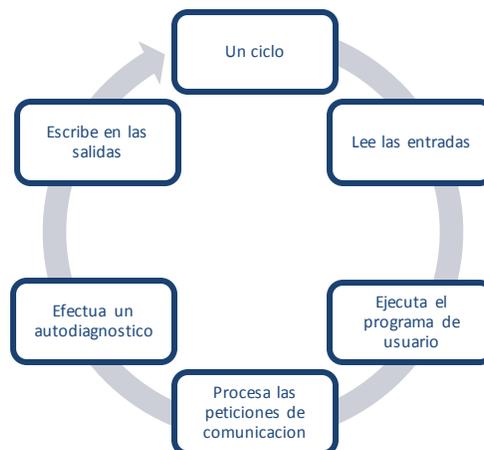


Imagen 2.6: Ciclo de trabajo de un PLC

Al principio de cada ciclo se leen los valores de las entradas digitales y se escriben en la imagen del proceso de las entradas. La CPU no actualiza las entradas analógicas como parte del ciclo normal, a menos que se haya habilitado.

Una vez guardados los valores de las entradas, se ejecuta el programa desde la primera operación hasta la última resolviendo los estados de salida y se guardan estos valores en la imagen del proceso

A continuación se efectúa el procesamiento de peticiones de comunicación que la CPU haya recibido por el puerto de comunicación y se efectúa un autodiagnóstico del firmware de la CPU y la memoria del programa.

Una vez realizados estos procesos, se escriben los valores de la imagen del proceso de las salidas en las salidas digitales.

Lenguajes de programación de PLC

Cuando hablamos de los lenguajes de programación nos referimos a diferentes formas de poder escribir el programa usuario.

Los softwares actuales nos permiten traducir el programa usuario de un lenguaje a otro, pudiendo así escribir el programa en el lenguaje que más nos conviene.

A la hora de programar un PLC se pueden usar diferentes tipos de lenguajes de programación. Todos ellos tienen elementos en común, pero también tienen sus características y ventajas individuales. Entre los lenguajes de programación más habituales se encuentran:

- Lista de instrucciones (AWL)
- Lenguaje de alto nivel (texto estructurado)
- Diagrama de contactos (KOP)
- Diagrama de Bloques funcionales (FUP)
- Lenguaje AWL

El lenguaje IL (Lista de instrucciones)

Permite crear programas de control introduciendo los nemónicos de las operaciones, como se puede observar en la figura 3. Por lo general, este lenguaje se adecúa especialmente para los programadores expertos ya familiarizados con los sistemas de automatización (PLCs) y con la programación lógica. Este lenguaje usa una pila lógica para trabajar.

```
NETWORK
LD    I0.0
LD    I0.1
LD    I2.0
A     I2.1
OLD
ALD
=     Q5.0
```

Imagen 2.7. Sintaxis de programación Lista de Instrucciones

El editor AWL también permite crear ciertos programas que, de otra forma, no se podrían programar con los lenguajes KOP ni FUP. Ello se debe a que AWL es el lenguaje nativo de la CPU, a diferencia de los editores gráficos en los que son aplicables ciertas restricciones para poder dibujar los diagramas correctamente y provienen de una traducción del lenguaje nativo

Texto estructurado (ST)

En el lenguaje de texto estructurado las instrucciones son líneas de texto que utilizan palabras o símbolos reservados. Las operaciones se definen por los símbolos matemáticos habituales. También se dispone de funciones trigonométricas, logarítmicas y de manipulación de variables complejas. Sin embargo, lo que distingue realmente estos lenguajes avanzados de las listas de instrucciones son las tres características siguientes:

- Son lenguajes estructurados, donde es posible la programación por bloques con definición de variables locales o globales.
- Incluyen estructuras de cálculo repetitivo y condicional. □ Disponen de instrucciones de manipulación de cadenas de caracteres, muy
- En aplicaciones de gestión, estadística, etc.

```
IF value < 7 THEN
  WHILE value < DO
    Value := value + 1;
  END_WHILE;
END_IF;
```

Imagen 2.8. Sintaxis de programación SCL

Lenguaje KOP

El lenguaje KOP (diagrama de contactos) permite crear programas con componentes similares a los elementos de un esquema de circuitos. KOP es probablemente el lenguaje predilecto de numerosos programadores y encargados del mantenimiento de sistemas de automatización.

Los programas KOP hacen que la CPU emule la circulación de corriente eléctrica desde una fuente de alimentación, a través de una serie de condiciones lógicas de entrada que, a su vez, habilitan condiciones lógicas de salida. El programa se ejecuta segmento por segmento, de izquierda a derecha y luego de arriba a abajo.

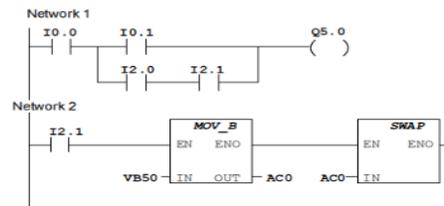


Imagen 2.9. Programación en lenguaje de escalera

Lenguaje FUP

El lenguaje FUP (Diagrama de funciones) permite visualizar las operaciones en forma de cuadros lógicos similares a los circuitos de puertas lógicas. En FUP no existen contactos ni bobinas como en el editor KOP, pero sí hay operaciones equivalentes que se representan en forma de cuadros.

La lógica del programa se deriva de las conexiones entre dichas operaciones de cuadro. Ello significa que la salida de una operación se puede utilizar para habilitar otra operación para crear la lógica de control necesaria. Dichas conexiones permiten solucionar numerosos problemas lógicos.

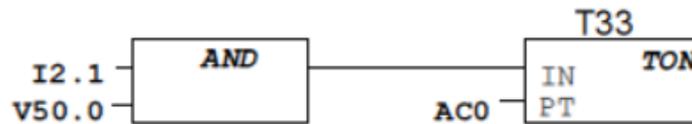


Imagen 2.10. Programación en Diagrama de funciones

PLC con Microcontroladores

En el diseño de PLC el sistema de unidad de procesamiento que más es utilizado es el de los microprocesadores, debido a su gran capacidad de manejo de información, sin embargo, existen sistemas embebidos con unidades de procesamiento con microcontroladores con una gran capacidad de procesamiento. ¿De qué depende en el diseño de un PLC avanzado el uso de un microprocesador y no de un microcontrolador?

Comparativa entre micro procesador y micro controlador

Analizaremos las diferencias, ventajas y desventajas que existen entre un micro procesador y un micro controlador además de dar a conocer sus principales características con respecto al desarrollo de PLC y como se evalúan con respecto a los siguientes elementos, funciones y especificaciones como: la CPU, las Memorias RAM y ROM, la Velocidad de Operación, el Tamaño, los Costos, las Interferencias (ruido), el Tiempo de desarrollo, etc.

Micro procesador

Son circuitos integrados que contienen millones de transistores en su interior los cuales crean circuitos complejos encargados de realizar diferentes tareas. **(Benchimol, 2011, p. 92)**

Un microprocesador como tal, es un chip integrado, diseñado para procesar datos binarios el cual incorpora en su interior una Unidad

Central de Proceso (CPU), que contiene un circuito de reloj, que puede ser programado para llevar a cabo múltiples tareas, Mayor es el número de transistores , porque requiere mayor capacidad de procesamiento. Para que un microprocesador pueda funcionar, necesita ser conectada con la memoria externa (RAM) y puertos de salida y de entrada sus elementos principales



Figura 2.11: Elementos principales de un microprocesador

Microcontroladores

Son circuitos integrados (chips) que contienen memorias (para almacenar programas), un procesador (para procesar y llevar a cabo los programas y pines de entradas y salidas para conectar interruptores, sensores y dispositivos de salida tales como motores **(Salzar, 2009, p. 6)**

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica:

- CPU (Unidad central de proceso)
- Memorias ROM (Memoria de solo lectura) y RAM (Memoria de acceso aleatorio)
- Líneas de entrada y salida (Periféricos)

CPU (unidad central de proceso) :

Podemos decir que la CPU, siglas en inglés de unidad central de proceso, es el núcleo del microcontrolador. Se encarga de ejecutar las instrucciones almacenadas en la memoria, de la que hablaremos más adelante. Es lo que habitualmente llamamos procesador o microprocesador, término que a menudo se confunde con el de microcontrolador.

En esta línea cabe aclarar que, tal y como estamos viendo, ambos términos no son lo mismo: el microprocesador es una parte de un microcontrolador y sin él no sería útil; un microcontrolador, en cambio, es un sistema completo que puede llevar a cabo de forma autónoma una labor.

Memoria

Entendemos por memoria los diferentes componentes del microcontrolador que se emplean para almacenar información durante un periodo determinado de tiempo. La información que necesitaremos durante la ejecución del programa será, por un lado, el propio código, y por otro, los diferentes datos que usemos durante la ejecución del mismo. Hablaremos por tanto de memoria de programa y de memoria de datos, respectivamente.

La diferente naturaleza de la información que hay que almacenar hace necesario el uso de diferentes tipos memorias. Sin hacer especial énfasis en este apartado, sí habrá que tener en cuenta una

clasificación básica, que distingue entre memoria volátil y no volátil. La primera es aquella que pierde la información que almacena al desconectarla de la alimentación; la segunda, como resulta obvio, no. Por lo tanto, se hace evidente que al menos la memoria de programa deberá ser no volátil

No sería práctico que el programa grabado en el microcontrolador se borrara cada vez que apagáramos el dispositivo. Con respecto a la memoria de datos, diremos por el momento según la situación puede interesarnos una u otra.

Unidades de entrada/salida

Las unidades de entrada/salida son los sistemas que emplea el microcontrolador para comunicarse con el exterior.

Imaginemos una televisión: por un lado tiene un dispositivo de salida, como es la pantalla, y por otro lado, de entrada, como son los botones de subir o bajar volumen y de cambio de canal. Así, los dispositivos de entrada nos permitirán introducir información en el microcontrolador y los de salida nos servirán para que éste la saque al exterior.

Algunos microcontroladores pueden utilizar palabras de 8 bits e incluso algunos llegan a manejar hasta 16 bits en palabras y funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia (mW).

Por lo general, tendrá la capacidad de mantenerse a la espera de un evento como pulsar un botón o de otra interrupción; así, el consumo de energía durante el estado de reposo (reloj de la CPU y los periféricos de la mayoría) puede ser sólo de nano Watts, lo que hace que muchos de ellos sean muy adecuados para aplicaciones con batería de larga duración.

Cuando es fabricado el microcontrolador, no contiene datos en la memoria ROM. Para que pueda controlar algún proceso es necesario generar o crear y luego grabar en la EEPROM o equivalente del microcontrolador algún programa; sin embargo, para que el programa pueda ser grabado en la memoria del microcontrolador, debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

Arquitecturas de los Microcontroladores

En fin un microcontrolador es un sistema completo, con unas prestaciones limitadas que no pueden modificarse y que puede llevar a cabo las tareas para las que ha sido programado de forma autónoma.

De forma general podemos definir la arquitectura de un microprocesador como un modelo y una descripción funcional de los requerimientos y las implementaciones de diseño para varias partes de una computadora, con especial interés en la forma en que la unidad central de proceso (CPU) trabaja internamente y accede a las direcciones de memoria.

“Es el diseño conceptual y la estructura operacional fundamental de un sistema de computadora” (Montaje de Componentes y Periféricos Microinformáticos), sería otra definición.

Con el concepto en mente podemos decir que en la actualidad existen dos arquitecturas que definen el uso y la distribución de la memoria interna del microcontrolador: La arquitectura de Von Neumann y Arquitectura Harvard, estas arquitecturas en principio, nos permitirán definir la forma en como los microcontroladores manejan el flujo de datos, y nos dará una perspectiva de cómo trabaja el microcontrolador en la ejecución un programa, en base a la arquitectura o composición

de las etapas internas que participan en el tratamiento de la información.

Arquitectura de Von Neumann

Tradicionalmente los sistemas con microprocesadores y microcontroladores se basan en esta arquitectura, en la cual la unidad central de proceso (CPU), está conectada a una memoria principal única (casi siempre sólo RAM) donde se guardan las instrucciones del programa y los datos. A dicha memoria se accede a través de un sistema de buses único (control, direcciones y datos).

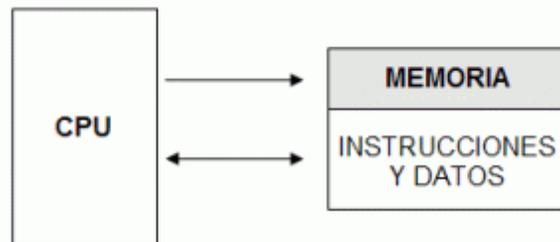


Imagen 2.12: Arquitectura de Von Neuman

En un sistema con arquitectura Von Neumann el tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus que comunica la memoria con la CPU. Así un microprocesador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits (bytes) de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria.

El tener un único bus hace que el microprocesador sea más lento en su respuesta, ya que no puede buscar en memoria una nueva instrucción mientras no finalicen las transferencias de datos de la instrucción anterior.

Las dos principales limitaciones de esta arquitectura tradicional son:

1. Que la longitud de las instrucciones está limitada por la unidad de longitud de los datos, por lo tanto el microprocesador debe hacer varios accesos a memoria para buscar instrucciones complejas.
2. La velocidad de operación (o ancho de banda de operación) está limitada por el efecto de cuello de botella que significa un bus único para datos e instrucciones que impide superponer ambos tiempos de acceso.

La arquitectura von Neumann permite el diseño de programas con código automodificable, práctica bastante usada en las antiguas computadoras que solo tenían acumulador y pocos modos de direccionamiento, pero innecesaria, en las computadoras modernas.

Esta arquitectura realiza los siguientes pasos secuencialmente para coordinar el procesamiento de los datos:

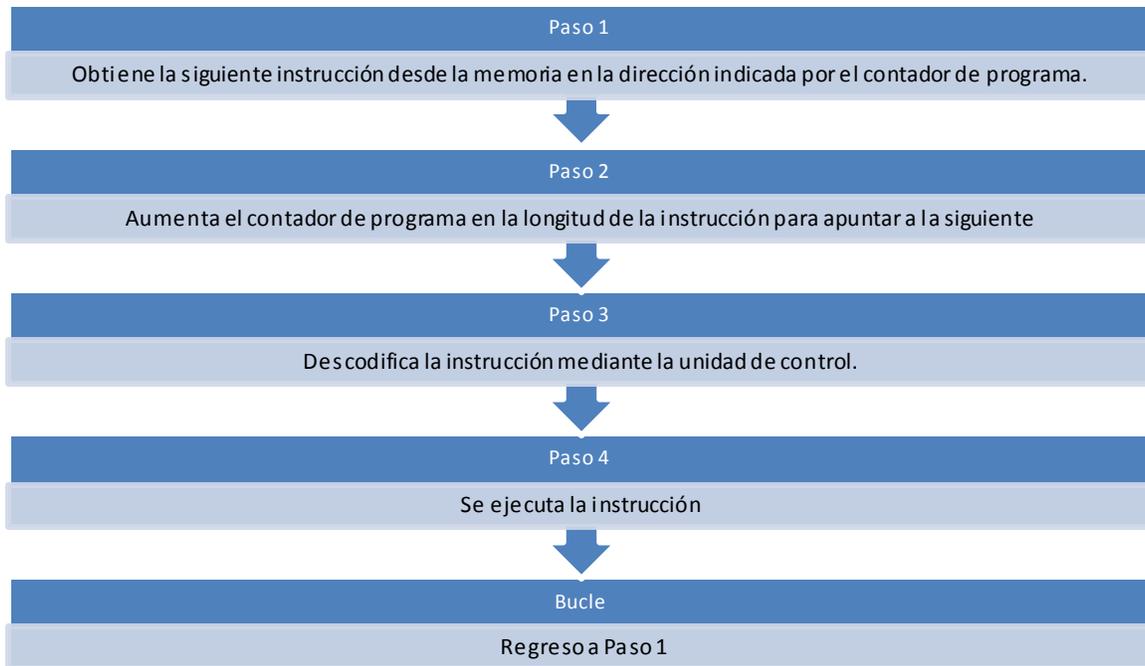


Figura 2.13: Secuencia para coordinar el procesamiento de datos

Arquitectura Harvard

Este modelo, que utilizan los Microcontroladores PIC, tiene la unidad central de proceso (CPU) conectada a dos memorias (una con las instrucciones y otra con los datos) por medio de dos buses diferentes.

Una de las memorias contiene solamente las instrucciones del programa, y es llamada a Memoria de Programa. La otra memoria solo almacena los datos y es llamada Memoria de Datos.

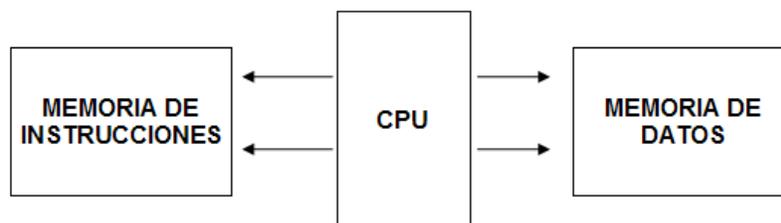


Figura 2.14 Arquitectura Harvard generalizada.

Ambos buses son totalmente independientes y pueden ser de distintos anchos. Para un procesador de Set de Instrucciones Reducido, o RISC (Reduced Instrucción Set Computer), el set de instrucciones y el bus de la memoria de programa pueden diseñarse de manera tal que todas las instrucciones tengan una sola posición de memoria de programa de longitud. Además, como los buses son independientes, el CPU puede estar accediendo a los datos para completar la ejecución de una instrucción, y al mismo tiempo estar leyendo la próxima instrucción a ejecutar.

Podemos observar claramente que las principales ventajas de esta arquitectura son:

- a) El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.
- b) El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad de operación.

Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontraran físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

Arquitectura del Set de Instrucciones de un Microcontrolador

Una de las primeras decisiones que toma el fabricante a la hora de diseñar un microprocesador es decidir cuál será su juego de instrucciones.

La decisión por dos razones; primero, el juego de instrucciones decide el diseño físico del conjunto; segundo, cualquier operación que deba ejecutarse en el microprocesador deberá poder ser descrita en términos de un lenguaje de estas instrucciones.

En principio, son una serie de instrucciones que una unidad central de procesamiento puede entender y ejecutar, o el conjunto de todos los comandos implementados por el fabricante en el diseño particular de una CPU en el microcontrolador.

Estos set de instrucciones describen los aspectos del procesador sumamente útiles para el programador del dispositivo o el programador del entorno de desarrollo integrado, incluyendo los tipos de datos nativos, las instrucciones, los registros, la arquitectura de memoria y las interrupciones, entre otros aspectos. Existen tres tipos de arquitecturas del conjunto de en los microcontroladores son.

- RISC (Reduced Instruction Set Computer)
- CISC (Complex Instruction Set Computer)
- SISC (Simple Instruction Set Computing)

Pero las más comunes en los microcontroladores son las arquitecturas RISC y CISC.

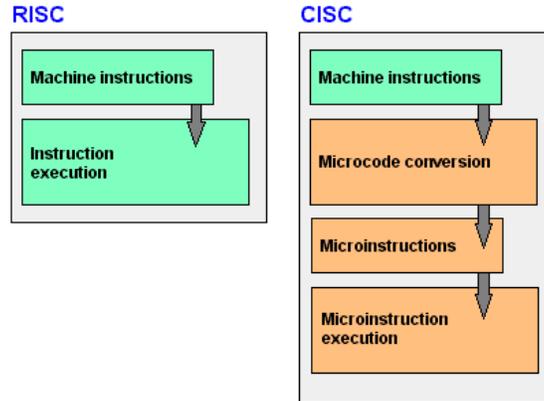


Figura 2.15 Diagrama secuencial de Sistemas de Instrucciones RISC y CISC

RISC (Reduced Instruction Set Computer) - Computadora con Juego de Instrucciones Reducidas.

RISC (Reduced Instruction Set Computer) en español "Computador con Conjunto de Instrucciones Reducidas", es un tipo de diseño de CPU generalmente utilizado en microprocesadores o microcontroladores con las siguientes características fundamentales:

- Instrucciones de tamaño fijo y presentado en un reducido número de formatos.
- Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

En términos generales, RISC es una filosofía de diseño de CPU para un microprocesador/microcontrolador que está a favor de conjuntos de instrucciones pequeñas y simples que toman menor tiempo para ejecutarse.

CISC (Complex Instruction Set Computer) - Computadoras con un juego de instrucciones complejo.

En la arquitectura computacional, CISC (complex instruction set computer). Los microprocesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y permiten operaciones complejas entre operando situados en la memoria o en los registros internos, en contraposición a la arquitectura RISC.

Este tipo de arquitectura dificulta el paralelismo entre instrucciones, por lo que, en la actualidad, la mayoría de los sistemas CISC de alto rendimiento implementan un sistema que convierte dichas instrucciones complejas en varias instrucciones simples del tipo RISC, llamadas generalmente microinstrucciones.

En conclusión los microprocesadores van orientados al manejo y procesamiento de grandes cantidades de información, los cuales necesitan de una serie de componentes externos para poder realizar de forma óptima su operación, en cambio los microcontroladores son computadores, los cuales en su arquitectura de construcción interna ya cuentan con todos los elementos que se requieren para desarrollarse en aplicación específicas.

Cuando estas aplicaciones requieran precisión y alto rendimiento el microprocesador es el dispositivo correcto, aunque resulta un tanto costoso y delicado por los elementos auxiliares que necesita para operar, si la aplicación va orientada a tratamiento de señales, manejo de puertos de entrada y salida, el dispositivo correcto es el microcontrolador, aunque se pierda, potencia de manejo de datos, velocidad de operaciones aritméticas, y rendimiento en general.

Tabla 1: Comparativa entre microprocesadores y microcontroladores

	Microprocesador	Microcontrolador
CPU	Posee mucha más potencia de cálculo, realiza sus funciones con datos, algoritmos o programas establecidos	Es una de sus funciones principales la cual se encarga de dirigir sus operaciones
Memoria RAM y ROM	Son dispositivos externos que lo complementan para su óptimo funcionamiento.	Las memorias están incluidas en un circuito integrado
Velocidad de operación	Rápida	Lenta en comparación con la de un microprocesador
Tamaño	La configuración mínima básica de un Microprocesador está constituida por un Micro de 40 Pines, Una memoria RAM de 28 Pines, una memoria ROM de 28 Pines y un decodificador de direcciones de 18 Pines, lo cual lo convierte en un circuito bastante engorroso.	Incluye todos estos elementos en un solo Circuito Integrado por lo que implica una gran ventaja en varios factores, como por ejemplo, la disminución en el tamaño del circuito impreso por la reducción de los circuitos externos.

Costos	Para el Microprocesador, el costo es muy alto en la actualidad.	El costo para un sistema basado en Microcontrolador es mucho menor.
Interferencias	Son más susceptibles a la interferencia electromagnética debido a su tamaño y a su cableado externo que lo hace más propenso al ruido.	El alto nivel de integración reduce los niveles de interferencia electromagnética
Tiempo de Desarrollo	El tiempo de desarrollo de un microprocesador es lento.	Por el contrario, el de un microcontrolador es rápido.

Plataformas De Microcontroladores En Placas De Hardware Libre

Placas Pingüino (Microchip)

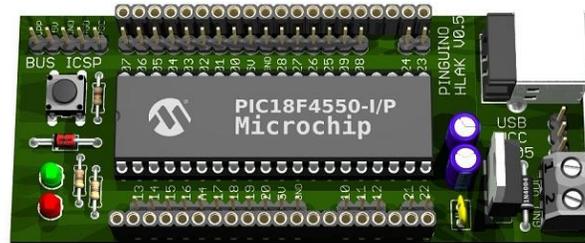


Imagen 2.16: Placa Pingüino

Pingüino (Microchip) es una plataforma de hardware y software "open source" para la experimentación con microcontroladores, basada en un microcontrolador PIC18F2550 y cuenta con su propio entorno de Desarrollo Integrado el cual permite programar algoritmos de programas en lenguajes de alto nivel.

Un detalle que diferencia de la placa Arduino, el Pingüino no necesita una Interfaz UART a USB adicional para comunicarse con la PC, debido a que el microcontrolador PIC18F2550 tiene un módulo USB integrado, lo cual le permite comunicarse directamente con la PC y dejando además libre el puerto UART del microcontrolador para las aplicaciones.

Queda a opción del programador utilizar la plataforma con un bootloader precargado para manejar el programa principal en la memoria flash del microcontrolador, o bien trabajarla directamente en la memoria ROM, en cualquier caso las prestaciones de esta plataforma son muy buenas, aunque las prestaciones de hardware no vengán especificadas y quedan más a criterio del diseño final del programador.

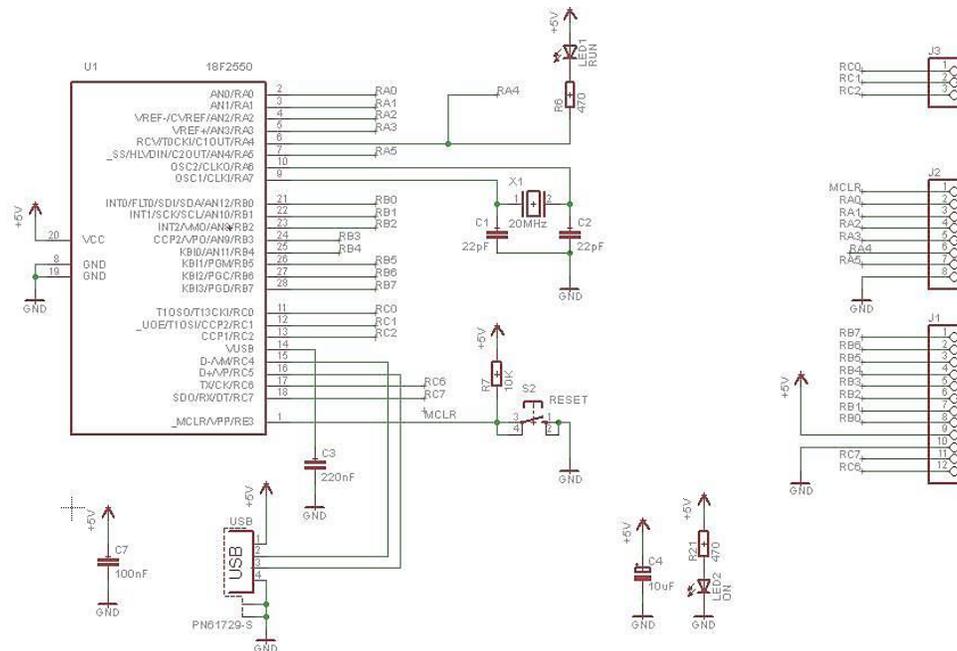


Figura 2.17 Estructura interna de PIC 18f2550

Sin embargo esta iniciativa de proyecto a pesar que trato en sus inicios solamente con el PIC 18F2550, actualmente existen otras tarjetas de pingüino basadas en microcontroladores de mayor potencia, en arquitecturas de hasta 32 bits, lo cual la hace sumamente superior a las de Arduino, El chipKIT basado en PIC32 permite velocidades de operación de 80 MHz, y ofrece hasta 512 KB de memoria Flash de programa, con un máximo de 128 KB de RAM. Las placas cuentan con una amplia conectividad de periféricos que incluyen Ethernet, CAN, USB, SPI, e interfaces UART.

Esta iniciativa nace Junto a Digilent, Microchip con la idea de poder competir con la comercialización de Arduino, poniendo a disposición de las personas interesadas un microcontrolador sumamente potente y accesible relativamente a lo que se refiere a costos.

Placas Arduino (ATMEL AVR)

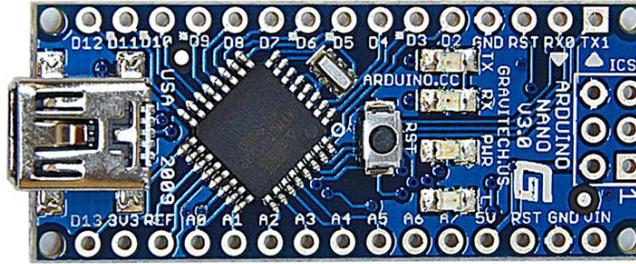


Imagen 2.18. Tarjeta Arduino NANO

Es otra plataforma de hardware de código abierto más popular en la actualidad, basada en una placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing. Diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Arduino nos proporciona un software consistente en un entorno de desarrollo (IDE) que implementa el lenguaje de programación de arduino y el bootloader ejecutado en la placa. La principal característica del software de programación y del lenguaje de programación es su sencillez y facilidad de uso.

Estructura de Arduino (ATmega 328/P)

CPU (Microcontrolador)

Es una plataforma basada en un microcontrolador ATmega 328P el cual es un microcontrolador de la empresa fabricante de chips Atmel® picoPower® es un microcontrolador tipo CMOS de baja potencia, con un bus de datos 8 bits, basado en la arquitectura AVR® RISC mejorada.

Ejecutando potentes instrucciones en un solo ciclo de reloj, el ATmega328 / P, alcanza rendimientos cercanos a 1 MIPS (Millones de Instrucciones por Segundo) por MHz. Este microcontrolador es normalmente acompañado por un reloj externo de 20 MHz, lo cual implicaría una velocidad de 20 MIPS.

Entradas: son aquellos pines de nuestra placa Arduino que están designados para realizar las lecturas. En la placa Arduino nano son los pines digitales (del 0 al 13) y los analógicos (del A0 al A7)

Salidas: son los pines asignados para enviar las señales En este caso los pines de salida son sólo los digitales (0 a 13).

Otros pines: Los pines GND (tierra), 5V que proporciona 5 Voltios, 3.3V que proporciona 3.3 Voltios, los pines REF de referencia de voltaje, TX (transmisión) y RX (lectura) también usados para comunicación serial, RESET para resetear, Vin para alimentar la placa y los pines ICSP para comunicación SPI.

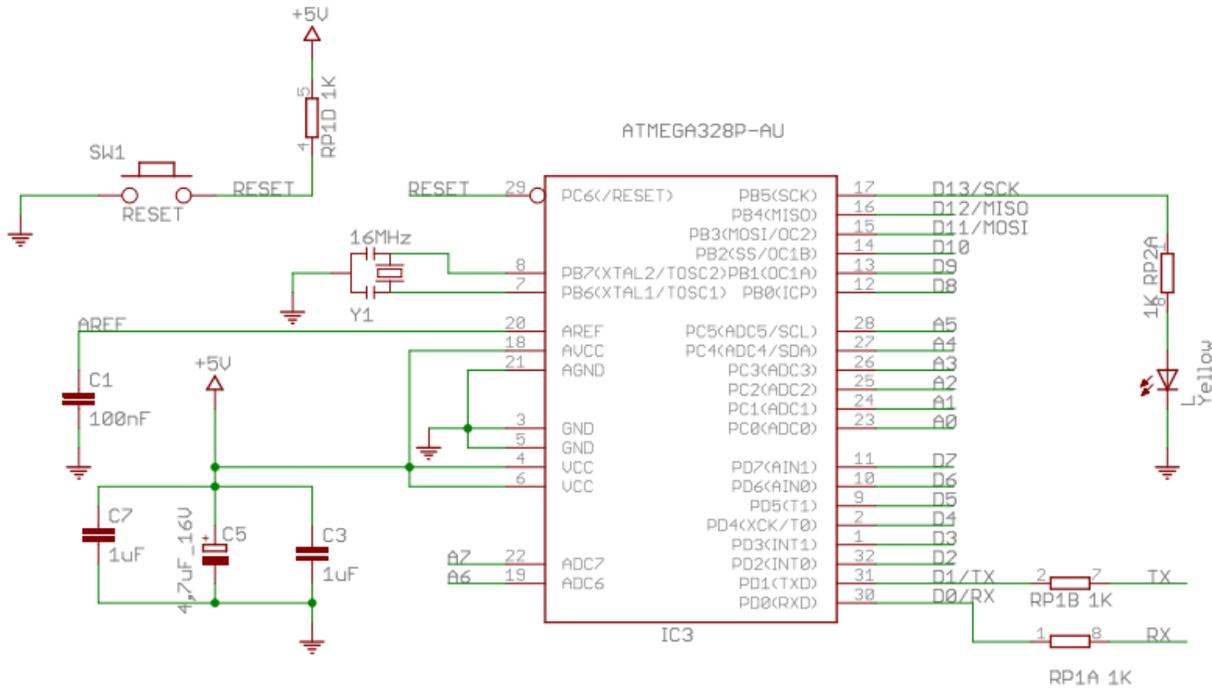


Figura 2.19 Diseño de Circuito con ATmega 328P en Arduino.

Alimentación: Para poder alimentar nuestro arduino lo hacemos por medio del pin Vin pero lo más normal es alimentarlo por el Jack de alimentación usando una tensión de 7 a 12 Voltios. También podemos alimentarlo por el puerto USB.

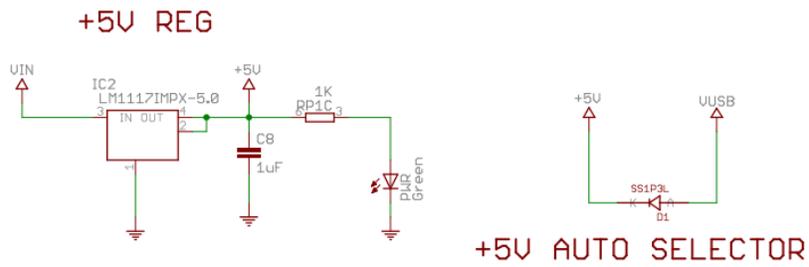


Imagen 2.20 Circuito regulador de voltaje Arduino NANO

Comunicación: La forma más común para establecer la comunicación a través de nuestro arduino es por medio de un puerto USB para cargar los programas o enviar/recibir datos. Arduino posee otras formas de comunicarse como insertando un shield por medio de los pines ICSP (comunicación ISP) los pines 10 a 13 también utilizados para comunicación ISP , además de los pines TX/RX o cualquiera de los digitales ya que son capaces de configurarse como pines de entrada o salida y recibir o enviar pulsos digitales.

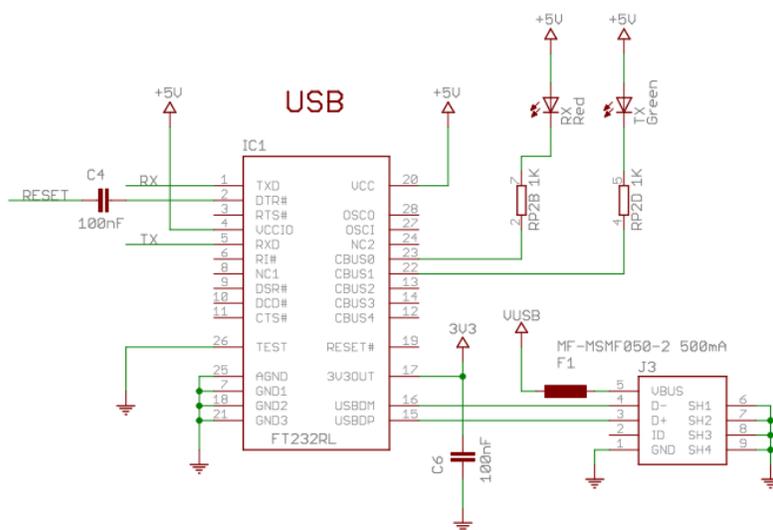


Imagen 2.21 Procesador de Comunicación UART/USB

Shields. Se llama así a las placas que se insertan sobre Arduino a modo de escudo o protección ampliando sus posibilidades de uso. En el mercado existen infinidad de shields para cada tipo de Arduino. Algunas de las más comunes son las de Ethernet, Wi-Fi, Ultrasonidos, Pantallas LCD, relés, matrices Leds, GPS.

Criterios Para El Diseño De Sistemas De Automatización

En el mercado de la automatización se ha venido desarrollando una serie de avances en los dispositivos de control de sistemas y aplicaciones, de lo que comúnmente llamamos PLC, se ha desarrollado el tema, hasta los pocos conocidos PAC (Controlador Programable de Automatización), sistemas avanzados, que en otros términos son controladores con la potencia de una computadora y la versatilidad de control de periféricos de un PLC, estos sistemas nacen debido a las aplicaciones en sistemas críticos y avanzados, que requieren altas prestaciones de procesamiento y sistemas de comunicación potentes, como en plantas petrolíferas, como en plantas nucleares, minería, etc.

Sin embargo a pesar de los grandes avances como lo son estos nuevos sistemas, también los sistemas de control sencillos y básicos se mantienen con un soporte de hardware de parte de los fabricantes de PLC muy buenos y de muy buen rendimiento, que particularmente son soportados con sistemas de hardware basados en microcontroladores de 8, 16 y 32 bits, micros PLC como LOGO! De Siemens, Zelio de Shneider Electric, ZEN de OMRON.



Imagen 2.22 Micro PLC de Allen Bradley - PICO Controller.

Es decir que los microcontroladores no solo han sido aplicados en sistemas embebidos sino también utilizados en sistemas industriales reprogramables como lo son los micros PLC, sistemas que no requieren gran potencia de procesamiento para el control de aplicaciones sencillas y básicas.

Entorno Industrial para sistemas de Automatización PLC

Compatibilidad Electromagnética del Micro PLC

Hablamos de compatibilidad electromagnética en nuestro PLC o cualquier dispositivo electrónico a los mecanismos que nos ayudan a eliminar, disminuir y prevenir los efectos de acoplamiento entre nuestro equipo y su entorno electromagnético. Desde su diseño basándose en normas y regulaciones internacionales asegurando la confiabilidad y seguridad de todos los tipos de sistemas en el lugar donde sean instalados y bajo un ambiente electromagnético.

Quiere decir que un equipo es electromagnéticamente compatible cuando este funciona en un ambiente electromagnético de forma satisfactoria y sin producir interferencias o perturbaciones electromagnéticas que afecten la operación normal de cualquier aparato o dispositivo que se encuentra en ese ambiente.

La compatibilidad electromagnética debe ser tenida en cuenta desde el principio de cualquier diseño electrónico, que más adelante serán explicados algunos los cuales son económicamente posibles de tratar en aplicación al diseño de los circuitos de nuestro PLC, tomados del libro de Compatibilidad Electromagnética y Seguridad funcional en sistemas electrónicos, pero términos generales podemos destacar los siguientes:

Segregación de Circuitos

Los componentes electrónicos deben estar agrupados en el circuito impreso, siguiendo los criterios correctos, evitando la mezcla de circuitos ruidosos con circuitos sensibles. También hay que hacer una buena clasificación según niveles de voltaje y tipos de señales con los que operan cada circuito. Aquí también hay que definir donde van a quedar ubicados cada uno de los circuitos funcionales.

Diseño de Sistemas de Alimentación.

El sistema de alimentación es una parte importante de todo el circuito, cuyo buen funcionamiento puede afectar a todos los circuitos integrados que deben ser alimentados. Este es mucho más que un simple regulador y unos condensadores, dado que los planos, las vías, el tipo de condensador de desacoplo o las bobinas en serie como los reguladores PI, con algunos circuitos ruidosos como los microcontroladores pueden variar el comportamiento global de nuestro PLC.

Estructura de Masas.

Una vez definidos los dos apartados anteriores es el momento de seguir con la estrategia con las masas. Hay que decir cuántos planos de masa vamos a utilizar, la forma de interconectarlos entre si y de que naturaleza serán los planos de alimentación asociados a cada plano de masa.

Estructura de alimentación.

Los planos de alimentación están directamente relacionados con el tipo de alimentación que necesitan los circuitos de 5VDC, 12VDC o 24VDC, y los planos de masa suelen estar relacionados con la naturaleza del circuito (digital, analógico, alta velocidad, entradas y salidas).

Entradas, salidas y protección.

Las entradas y salidas deben estar bien diseñadas ya que a través de ellas el ruido generado en nuestros circuitos puede alcanzar el cableado y producir radiaciones al entorno. También a través de las entradas es posible recibir interferencias que serán transmitidas después a nuestros circuitos, alterando su funcionamiento. Hay que intentar ver los caminos que seguirán las descargas electroestáticas o ESD, para proveer una buena protección a las entradas.

Diseño del oscilador de reloj del microcontrolador.

Son los circuitos con osciladores que producen emisiones de elevada energía al entorno. No debemos confundir las interferencias producidas por el oscilador de reloj del microcontrolador con las producidas por este mismo a esa misma frecuencia de reloj. Estos producen intensas interferencias a la frecuencia de trabajo, en unas ocasiones esta frecuencia será la misma que la frecuencia del oscilador y en una frecuencia distinta generada por un PLL (Phase Locked Loop), circuito multiplicador de frecuencia.

Desacoplo de los circuitos integrados.

Comúnmente son los capacitores los que realizan esta función, los cuales proveen energía adicional a los pines de alimentación de los circuitos integrados. Nuestro objetivo primordial debe estar centrado

en conseguir la mínima inductancia posible en la conexión de los condensadores de desacoplo. Cuanta menos inductancia tengan sus terminales más efectivos serán los condensadores y menos problemas tendremos en nuestros circuitos.

Todo lo anterior es útil e importante sin embargo, no existe una solución única para crear diseños de prototipos o productos que pasen las pruebas de compatibilidad electromagnética, por el contrario, debemos de aplicar una serie de técnicas para conseguirlo, como las mencionadas anteriormente.

Normalización de los PLC - IEC 61131

Esta norma se refiere a los autómatas programables (AP o PLCs) y a sus periféricos como:

- Los equipos de programación y depuración (PADT's)
- Los equipos de ensayo (TE's)
- Los interfaces hombre-máquina (MMI's)

Esta norma no trata del sistema automatizado, del cual el autómata programable es un componente básico.

- PADT: Programming And Debugging Tool
- TE: Test Equipment
- MMI: Man-Machine Interface

La finalidad de esta Norma IEC-61131

1. Definir e identificar las características principales que se refieren a la selección y aplicación de los PLC's y sus periféricos.

2. Especificar los requisitos mínimos para las características funcionales, las condiciones de servicio, los aspectos constructivos, la seguridad general y los ensayos aplicables a los PLC's y sus periféricos.
3. Definir los lenguajes de programación de uso más corriente, las reglas sintácticas y semánticas, el juego de instrucciones fundamental, los ensayos y los medios de ampliación y adaptación de los equipos.
4. Dar a los usuarios una información de carácter general y unas directrices de aplicación. • Definir las comunicaciones entre los PLC's y otros sistemas.

Condiciones de servicio normales.

1. Condiciones del entorno físico: temperatura, humedad, contaminación, inmunidad a la corrosión, altitud.
2. Condiciones de servicio y requisitos eléctricos: alimentaciones, ruido eléctrico, sobretensiones, etc.
3. Condiciones de servicio y requisitos mecánicos: Vibraciones, choque, caída libre.
4. Condiciones de servicio especiales. Polvo, humo, partículas radiactivas, vapores, sales, insectos, pequeños animales, etc.
5. Requisitos para el transporte y almacenaje: Temperatura, presión atmosférica, humedad relativa.

Requisitos Eléctricos

- Alimentación de corriente alterna (c.a.) y continua (c.c.).
- E/S digitales.
- E/S analógicas

- Interfaces de comunicación.
- Procesador(es) principal(es) y memoria(s) del sistema AP.
- Estaciones de entrada/salida remota (RIOS).
- Periféricos: PADT, TE, MMI.
- Inmunidad al ruido y ruido emitido.
- Propiedades dieléctricas.
- Auto diagnóstico y diagnóstico.

Requisitos Mecánicos

- Protecciones contra el riesgo de choque eléctrico.
- Requisitos de distancias en el aire y líneas de fuga.
- Requisitos de inflamabilidad para materiales aislantes.
- Envolvente.
- Requisitos mecánicos de los materiales de conexión.
- Disposiciones para la tierra de protección.
- Tierra funcional.
- Cables y conectores de interconexión.

Normativas para diseño de PLC y equipos de automatización.

Muy importante al momento de diseñar un dispositivo electrónico para entornos industriales son las normas que tratan la EMC, en resumen

cada una, según la organización CENELEC Europea, que regula los estándares y normativas europeas, son:

EN 50 155 Temperatura de operación.

EN 55 022 Equipos de tecnología de la información. EMI

EN 55 024 Equipos de tecnología de la información. INMUNIDAD

EN 61000-3-2 Corriente armónica en la red de C.A. $\leq 16A$

EN 61000-3-3 Fluctuaciones de tensión y Ficker en la red de C.A. $\leq 16A$

EN 61000-3-15 EMC Generadores dispersos, conexión red eléctrica VAC

EN 61000-4-2 Descargas electrostáticas. (ESD)

EN 61000-4-3 Campos electromagnéticos de alta frecuencia radiados.

EN 61000-4-4 Transitorios eléctricos rápidos en ráfagas. (BURST)
(EFT)

EN 61000-4-5 Impulsos de alta energía u ondas de choque. (Surges)

EN 61000-4-6 Campos electromagnéticos de alta frecuencia conducidos.

EN 61000-4-8 Campos magnéticos a frecuencia industrial.

EN 61000-4-9 Campos magnéticos pulsados.

EN 61000-4-10 Campos magnéticos amortiguados.

EN 61000-4-11 Fallos, fluctuaciones, cortes y micro-cortes en la alimentación a. c.

EN 61000-4-12 Ondas amortiguadas en la alimentación

EN 61000-6-1 CEM, Genérica Inmunidad: Residencial e inducción Ligera

EN 61000-6-3 CEM, Genérica Emisiones: Residencial e inducción Ligera.

EN 61000-6-2 CEM, Genérica Inmunidad: Ambiente Industrial.

EN 61000-6-4 CEM, Genérica Emisiones: Ambiente Industrial. (Uso restringido) (A)

Luego de tomar de esta serie de normas los elementos o términos que aplican al diseño del prototipo, surge la cuestión, de cómo garantizamos, que nuestro equipo es electromagnéticamente compatible, es decir, ¿qué pruebas esenciales fueron aplicadas a nuestro diseño para certificarlo con una EMC aceptable? Estos pueden ser:

- Seguridad General (Diseño general del prototipo)
- Protección de los envolventes (Grado de protección y diseño de la carcasa o chasis)
- Seguridad eléctrica (Comportamiento a estímulos: cortocircuitos, distorsiones de señal digitales o analógicas, buses de comunicación, etc.)
- Emisiones Electromagnéticas conducidas.
- Emisiones Electromagnéticas radiadas.
- Inmunidad o susceptibilidad electromagnética conducida.
- Inmunidad o susceptibilidad electromagnética radiada.
- Pruebas de Radio Frecuencia R&TT, mínimos esenciales.

En términos legales el ente certificador de equipos que garantiza que el producto se ha evaluado y cumple los requisitos de seguridad, sanidad y protección del medio ambiente exigidos por la UE electrónicos domésticos e industriales es el CE junto con la RoHS que define la Restricción de ciertas Sustancias Peligrosas en aparatos eléctricos y electrónicos pueden certificar nuestro equipo para exportación internacional.

La documentación e informes para obtener la marcación CE son las siguientes:

- Informes del laboratorio de ensayos, con los resultados de las directivas y normas que le son de aplicación.
- Expediente técnico constructivo
- Análisis de riesgos
- Manuales de utilización, instalación, mantenimiento,
- Declaración de conformidad.
- Etiqueta



Imagen 2.23 Etiqueta CE de certificación

Estructura Básica de un PLC con microcontroladores

Un PLC basado en micro controladores en su estructura de diseño electrónico no difiere en mucho de los basados en microprocesadores, en principio se manejan las mismas cualidades de diseño para control de periféricos, los cuales se detallaran a continuación.

Fuentes de Alimentación DC/DC

El primer término a tomar en cuenta en el diseño del PLC es, que tipo de voltaje utilizaremos para alimentar todos los sistemas de circuitos que componen en conjunto el PLC, de estos pueden existir diferente

tipos los cuales pueden ser AC y DC, por ejemplo los niveles estandarizados son:

- 12/24VDC
- 24VAC
- 110/240VAC

Del tipo de alimentación dependerán los circuitos de acondicionamiento de la fuente de alimentación. Si segmentamos por costos de diseño y construcción los PLC con alimentación a AC sea en bajo voltaje 12/24V o 120/240V el costo es mucho mayor, debido a los elementos de transformación y filtrado, dado que estos PLC son más propensos de transmitir ruido eléctrico en sus alimentaciones, los cuales trastornan el funcionamiento del PLC.

Sin embargo, aunque la alimentación sea AC o DC, siempre existirán dentro del PLC circuitos reguladores integrados o de conmutación los cuales acondicionan los niveles de voltaje para:

- Periféricos de Entrada y Salida
- Alimentación del CPU (Microcontrolador o Microprocesador)
- Procesadores de Comunicación
- Buses modulares de Bastidor

En definición general un sistema de regulación de voltaje es aquel que convierte o modifica un diferencial de voltaje a otro más bajo, con la característica de mantener dicha tensión constante aun que exista variación en la carga, ya sea por técnicas de regulación lineal o conmutada.

Normalmente en los PLC luego de la etapa de regulación principal existen sistemas de acondicionamiento a niveles todavía más bajos por ejemplo a 12V, 5V, 3.3V. Estos debidos a que en electrónica digital

solo se manejan fuentes alimentación DC. Dependerá de los circuitos integrados, los microcontroladores y demás sistemas electrónicos que requieran dichos niveles de voltaje. Estos a su vez pueden o no pueden tener una masa común con la fuente principal de regulación, dependerán de los criterios de aislamientos de circuitos quienes determinaran esta parte.

Para estas regulaciones utilizamos reguladores DC/DC las cuales pueden ser lineales debido a que los circuitos de alimentación son constantes y no provocan grandes disipaciones de potencia. Estos son muy comunes en los sistemas de acondicionamiento de voltajes en los PLC.

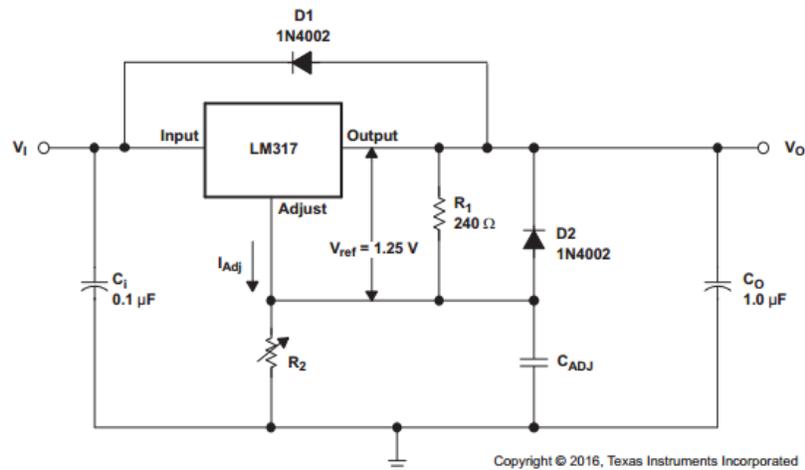


Figura 2.24: Circuito Típico de un regulador lineal LM317.

Diseño de Entradas de Señales Digitales y Analógicas

A considerar en el diseño de un PLC son los tipos de señales que este recibirá de sensores, pulsadores etc. Que le permitirán interpretar el estado del sistema a controlar.

Para estas pueden existir múltiples configuraciones, estas dependerán del tipo de voltaje a recibir a la entrada, estas pueden ser en AC o DC.

Definición de entradas a AC

Definimos las entradas de un PLC, con señales de voltaje en valores que rondan los 120/240 VAC; este tipo de diseño, está orientado a aplicaciones domóticas o aplicaciones industriales, en las cuales los niveles de tensión de las señales deben ser mayores debido a caídas ocasionadas por la longitud del cable, y donde no es exigente el acondicionamiento o aislamiento de las señales de AC a DC. Normalmente este tipo de PLC es del tipo modular compacto para contar con entradas analógicas.

Definición de Entradas a DC

Es el tipo de entrada más común que se utiliza en el diseño de PLCs. Esto debido a los niveles de tensión que manejan sensores o normas de diseño de control de sistemas automatizados. Por ejemplo para este tipo de entradas con un voltaje de 24VDC, podemos configurar contadores rápidos, para en conexión con encoders, una conexión que no sería posible de forma directa en un PLC con entradas AC.

Los niveles de voltaje DC más comunes para estos sistemas son: 12VDC y 24VDC. Mediante técnicas electrónicas, dichas entradas digitales pueden tener la capacidad de poder muestrear señales analógicas, esto dependerá del microcontrolador que se seleccione en el diseño.

Ambas entradas tanto AC, como DC, son debidamente aisladas y filtradas (Filtros RC), para separar los potenciales de señales externas con los niveles de voltaje que maneja el microcontrolador

Acondicionamiento de Entradas Digitales

Es común que la intensidad de una entrada digital dependerá del voltaje con el que se esté alimentando el PLC. Sin embargo esta intensidad no es la misma con la que trabaja el microcontrolador, es decir, la entrada

no está directamente conectada con el microcontrolador, debido a que las señales que manejan con microcontroladores oscilan entre los 5 VDC y 3.3 VDC. Obviamente si esto fuera así, el microcontrolador se destruiría debido a las diferencias de potencial.

Para que las señales que ocurren en el sistema automatizado y se envían al microcontrolador puedan ser eléctricamente seguras a una intensidad de voltaje correcto, que no nos ocasione problemas, debemos de contar de elementos que aislen dichos sistemas, la etapa de entradas, con la entrada en pines del microcontrolador. En la actualidad el aislamiento galvánico es la técnica más utilizada.

Diseño con aislamiento galvánico

El aislamiento galvánico es en principio, el método para aislar secciones funcionales de los sistemas eléctricos y así, evitar el flujo de corrientes entre sí; es decir, no se permite ninguna vía de conducción directa.

Sin embargo, la energía, corrientes o información entre ambas secciones, se puede intercambiar por distintos métodos, tales como: la inducción o las ondas electromagnéticas, la capacitancia, medios ópticos, medios acústicos o mecánicos.

Por ejemplo, en la inducción electromagnética, mediante transformador, los arrollamientos primario y secundario de un transformador no están conectados entre sí.

En conclusión, el aislamiento galvánico se utiliza cuando dos o más circuitos eléctricos deben comunicarse, por diferentes motivos como pueden ser a diferentes potenciales. Es un método eficaz de romper los lazos de tierra mediante la prevención de corrientes no deseadas que fluyen entre dos unidades que comparten un conductor de tierra.

Elementos de Aislamiento de Potencial - Optoacopladores.

Este preámbulo nos ayuda a comprender mejor el principio de operación del optoacoplador, el cual es un componente electrónico que no se suele conocer mucho, pero que juega un papel fundamental cuando estamos diseñando el método de captura de las señales digitales para nuestro PLC, principalmente cuando se trata de aislar eléctricamente dos circuitos.

Los Opto-aisladores transmiten información mediante ondas de luz. El emisor (fuente de luz) y un receptor (dispositivo fotosensible) no están conectados eléctricamente, normalmente se mantienen en su lugar dentro de una matriz de plástico aislante, opaco.

Los opto-acopladores tienen su campo de aplicación en el diseño de PLCs, en equipamiento electrónico de señal y transmisión de potencia de línea, que pueden ser sometidos a subidas de tensión inducida, ya sea por un rayo, descargas electrostáticas, las transmisiones de radiofrecuencia, impulsos de conmutación (picos) y las perturbaciones en el suministro de energía.

Los optoacoplador se utilizan dentro de un sistema para separar un bloque de otro conectado a la red eléctrica o de un bloque de alta tensión a otro tipo de tensión, para la protección de seguridad y el equipo. La función principal de un opto-aislador es bloquear las altas tensiones y transitorios de tensión, de manera que el aumento de una parte del sistema (circuitos de entradas o salidas digitales) no perturbe o destruya las etapas del microcontrolador.

Filtro Pasa Bajas en entradas DC - Filtros RC

En general los filtros son circuitos realizados con componentes pasivos para trabajar con la frecuencia de la señal.

De estos filtros podemos distinguir varios tipos de filtro:

- Filtros pasa bajas
- Filtros pasa altas
- Filtros pasa banda

De los cuales en el diseño de autómatas el más utilizado es el filtro pasa bajas, puesto que en ambientes industriales es muy común recibir por inducción o conducción señales parasitas a alta frecuencia que son indeseables al momento de leer una entrada digital o analógica.

Es muy común debido a costos y facilidad de aplicación utilizar filtros RC en las entradas, el mismo principio de filtrado se utilizan para las salidas como las redes Snubbers para amortiguar las conmutaciones de los relays, en el caso de las entradas analógicas el mismo principio bajo el termino de filtro antialiasing.

Aislar galvánicamente un sistema no es suficiente dado que las fluctuaciones a alta frecuencia, tienen características de resonancia por ejemplo al momento de pulsar un botón.

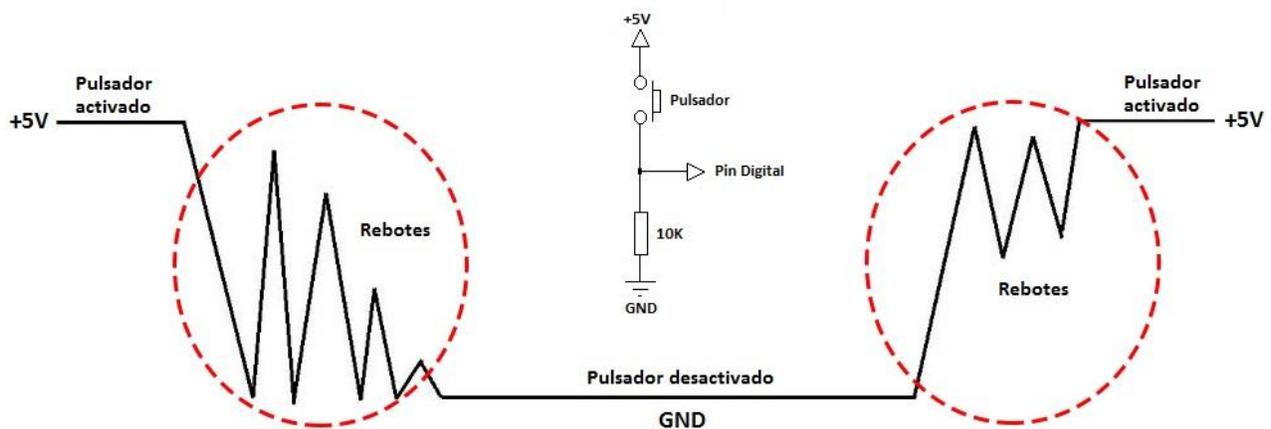


Imagen 2.25 Comportamiento de la señal de salida sin amortiguamiento.

Todos estos nos permiten filtrar frecuencias indeseables, tales frecuencias lo único que nos ocasionan, son perturbaciones y comportamientos imprevistos de los sistemas de control central, en este caso en los PLC microcontroladores o microprocesadores.

Diseño de Entradas Analógicas.

El PLC fue originalmente diseñado para el control de estados lógicos y es un equipo de tecnología digital. Por lo tanto la única manera de trabajar que tiene con valores analógicos es que éstos se representen internamente por medio de números binarios. La principal tarea de una entrada analógica es la de convertir un valor análogo en formato binario, por medio de un conversor A/D. Las entradas analógicas convierten una señal analógica en un número binario. Por ejemplo, una entrada analógica con un conversor de 8 bits puede dividir un rango de 4 a 20 mA en 256 valores.

Procesamiento digital de señales Analógicas.

Para el tratamiento de datos de las señales analógicas, existen los PDS o Procesadores Digitales de Señales.

El término DSP se aplica a cualquier chip que trabaje con señales representadas de forma digital. En la práctica, el término se refiere a microprocesadores específicamente diseñados para realizar procesamiento digital de señales analógicas. Los DSP utilizan arquitecturas especiales para acelerar los cálculos matemáticos intensos implicados en la mayoría de sistemas de procesamiento de señal en tiempo real.

En los sistemas de automatización puntualmente, la señales recibidas son enviadas directamente por un microcontrolador dedicado a este proceso, por ello y toda la gama de componentes que la componen,

quienes garantizan el procesamiento limpio de la señal, se eleva el costo de un módulo analógico respecto a uno digital.

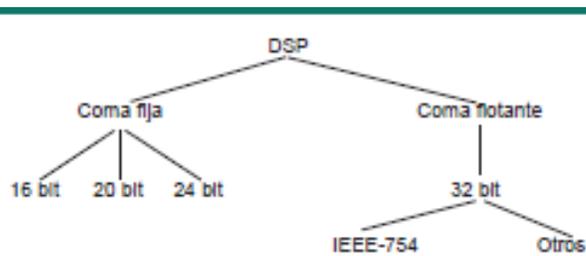


Imagen 2.26 Tipos de PDS.

Una de las características fundamentales de los PDS es el tipo de formato aritmético utilizado por el microprocesador. En la imagen 3.6 muestra los formatos de coma fija y coma flotante, respectivamente.

Ahora entenderemos cada uno, empecemos con el formato IEEE-754 de coma flotante, la "S" indica MSB es el signo, donde un 1 indica que se trata de un numero negativo. La letra "e" indica el exponente, formado por 8 bits y la "m", de 23 bits, la mantisa del número.

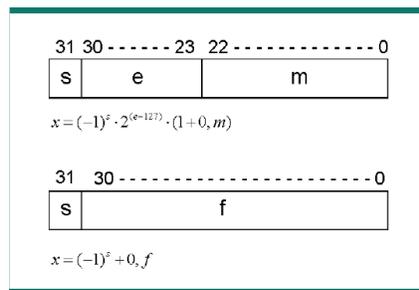


Imagen 2.27 Formatos Aritméticos de PDS.

Para un mismo tamaño en número de bits, el formato en coma fija proporciona una mejor resolución que el formato en coma flotante. Sin embargo, es este último quien posee un margen dinámico superior.

La aritmética de coma flotante es más flexible que la de coma fija. Con coma flotante, los diseñadores de sistemas tienen acceso a un

margen dinámico más amplio. En consecuencia, los DSP de coma flotante son generalmente más fáciles de programar que los de coma fija, pero son usualmente más caros. El mayor coste es resultado del requisito de una mayor complejidad circuital que se traduce en un mayor tamaño del microprocesador. Sin embargo, un mayor margen dinámico facilita su programación pues el programador no debe preocuparse por el margen dinámico ni por la precisión. Por el contrario, en los DSP de coma fija el programador a menudo debe escalar las señales en sus programas para asegurar una adecuada precisión numérica con el limitado margen dinámico del DSP de coma fija.

Ejemplo de ello sería comparar el procesamiento de señales analógicas en un Controlador LOGO! 0BA7 o incluso con el nuevo modelo 0BA8, con un S7 1200 serie 1211.

Particularmente en un LOGO! De Siemens no existe un PDS de coma flotante, sino de coma fija. El procesamiento de señales en un LOGO! Es realizado mediante una resolución de 10 bits, equivalente a 9.766 mV, con un nivel de cuantificación de 1024 valores posibles. Esto es así debido a que en logo trabaja con valores normalizados de 0 a 1000 mediante un ajuste de hardware y de software al momento de utilizar un amplificador analógico con corrección de ganancia y offset.

En cambio en un PLC completo se toman todos los posibles valores mediante un PDS de coma flotante y una resolución de hasta 13 bits para dinamizar el margen.

Resolución y Lectura de señales analógicas.

Para la lectura de estos datos es necesario entender los conceptos básicos de resolución y niveles de cuantificación.

De forma general, las fórmulas que se siguen para dicha tarea son las siguientes:

$$V_{in} = Valor\ ADC * Resolución$$

Donde:

$$Valor\ ADC = (Valor\ digital\ de\ lectura, \text{segun nivel max. de cuantificación})$$

$$Resolución = \frac{V_{ref}(+)}{2^n - 1}$$

Filtrado de entradas Analógicas - Filtros Antialiasing

Para cualquier tipo de señal analógica, el microprocesador realiza una serie de lecturas en intervalos definidos, en base a la frecuencia del mismo.

La naturaleza de estas señales en un PDS puede ser:

- Analógicas, $x(t)$: amplitud y tiempo continuos.
- Muestreadas, $x_s[n]$: tiempo discreto, amplitud continua.
- Cuantizadas, $x_Q(t)$: tiempo continuo, amplitud discreta.
- Digitales, $x_Q[n]$: tiempo y amplitud discretos.

De forma gráfica pueden ser:

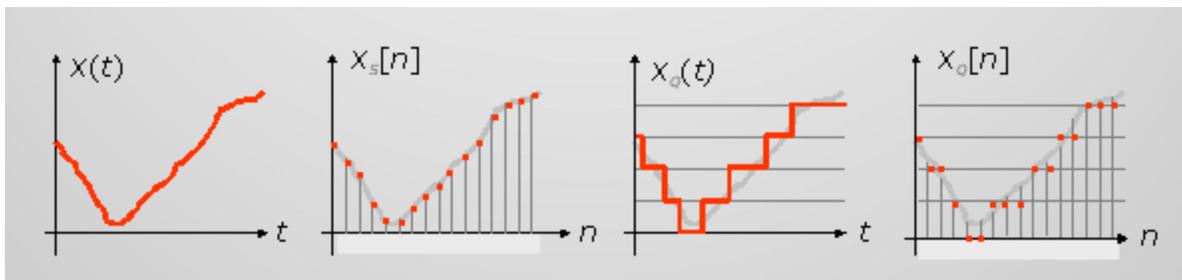


Figura 3.7: Graficas de una señal PDS

Esto nos ayuda a poder comprender lo que sucede al momento de tener una señal proveniente de un sensor, por ejemplo, si tenemos un lazo de voltaje de 0 a 10 voltios, la naturaleza de esta señal estaría determinada en amplitud y tiempos continuos. Esta señal al momento de ser recibida por una etapa de acondicionamiento ya sea por amplificadores u otro integrado, para este propósito, en el microcontrolador se convertiría en un tipo de señal muestreada, del tipo $xS[n]$, de tiempo discreto (determinado por la frecuencia de muestreo) y amplitud continua (la intensidad varia +/- según su naturaleza).

Un muestreo toma señales de valores en una frecuencia específica, de forma gráfica:

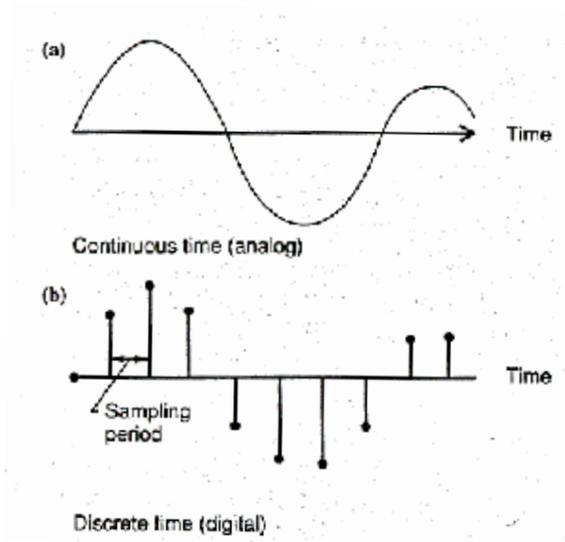


Figura 2.28: Toma de valores de una frecuencia específica

En este proceso hay que tomar en cuenta que pueden existir problemas en la representación de estas señales, ya sea a múltiples razones las más básicas en un PLC distorsiones armónicas transportadas en el lazo

de corriente o bien un mal diseño de acondicionamiento de la señal a muestrear.

De forma general, puede existir pérdida en el muestreo de una señal, dependiendo de la frecuencia de muestreo en relación con la frecuencia de la señal a muestrear.

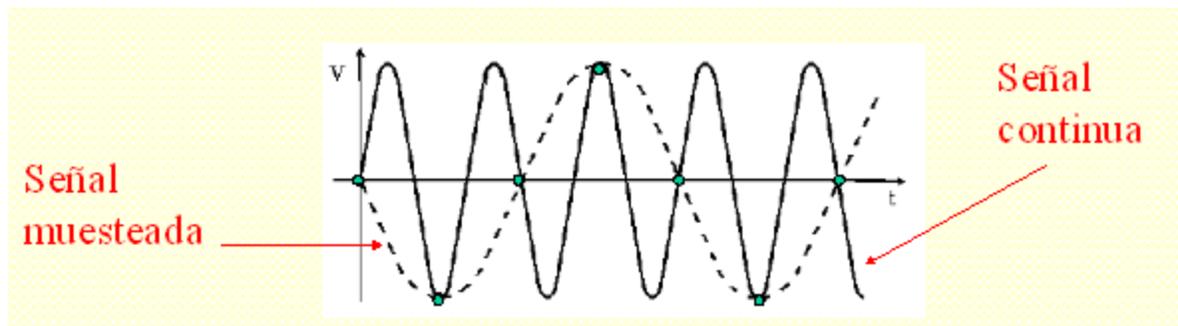


Imagen 2.29 Graficas de problemas de Aliasing.

A estos problemas de pérdidas en el muestreo de una señal, son conocidas como: "Errores de aliasing" o en otras palabras es cuando ocurre un enmascaramiento de la señal muestreada.

De acuerdo con el teorema de muestreo de Nyquist, la frecuencia de muestreo debe ser de al menos dos veces la frecuencia del máximo componente espectral de la señal de interés. En otras palabras, la frecuencia máxima de la señal de entrada debe ser menor que o igual a la mitad de la frecuencia de muestreo.

Sin embargo, ¿es posible garantizar que la señal cumple con dichas condiciones? Incluso si la señal medida posee un límite de frecuencia máxima, señales parásitas (tales como frecuencia de la red de alimentación o estaciones de radio locales) podrían contener frecuencias mayores a la frecuencia de Nyquist.

Dichas señales podrían verse reflejadas como componentes en el rango de interés y provocar resultados erróneos, si los analizamos con el teorema de Furrier.

Para asegurar que el contenido de frecuencia de la señal de entrada está limitado, un filtro pasa bajas (un filtro que permite el paso de frecuencias bajas pero atenúa las frecuencias altas) es añadido antes del muestreador y el ADC.

Este filtro se conoce como anti-aliasing debido a que, al atenuar las frecuencias altas (mayores que la frecuencia de Nyquist), evita que las componentes del aliasing sean muestreadas. Dado que en dicho punto (antes del muestreador y el ADC) la señal es aún analógica, el filtro anti-aliasing debe ser a su vez un filtro analógico.

Un filtro anti-aliasing ideal permitiría el paso de todas las frecuencias apropiadas (por debajo de f_1) y suprimiría todas las frecuencias no deseadas (por arriba de f_1). Sin embargo, no es posible implementar dicho filtro físicamente.

En la práctica, los filtros se comportan como se muestra en la ilustración (b) abajo, permitiendo el paso de todas las frecuencias menores que f_1 , y bloqueando todas las frecuencias mayores que f_2 . La región entre f_1 y f_2 se conoce como banda de transición, en la cual se da una atenuación gradual de las frecuencias de entrada.

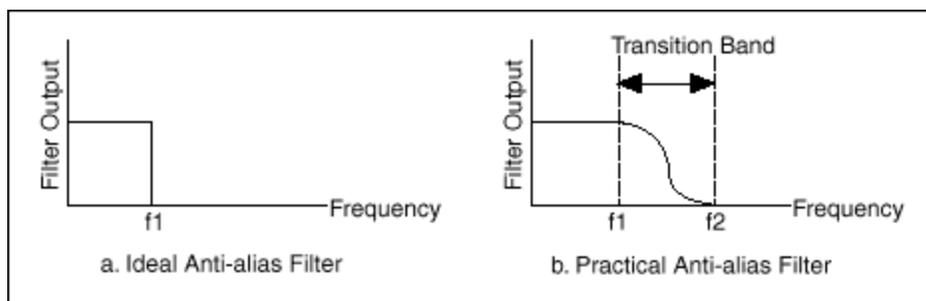


Imagen 2.30 Comportamiento de un filtro pasa bajas

A pesar de que se procura pasar únicamente las señales con frecuencias menores que f_1 , las frecuencias en la banda de transición podrían llegar a causar aliasing, motivo por el cual, en la práctica, la frecuencia de muestreo debe ser mayor a dos veces la mayor frecuencia en la banda de transición.

Por ejemplo, una señal de audio puede contener componentes hasta los 20 kHz. El teorema de muestreo de Nyquist afirma que la frecuencia de muestreo requerida para dicha señal sería de 40 kHz.

El filtro anti-aliasing tendría en ese caso una frecuencia de corte de 20 kHz, pero, dado que el filtro no es ideal, la frecuencia de muestreo utilizada podría ubicarse usualmente entre 44.1 kHz y 96 kHz, dando cabida a una banda de transición de al menos 2 kHz.

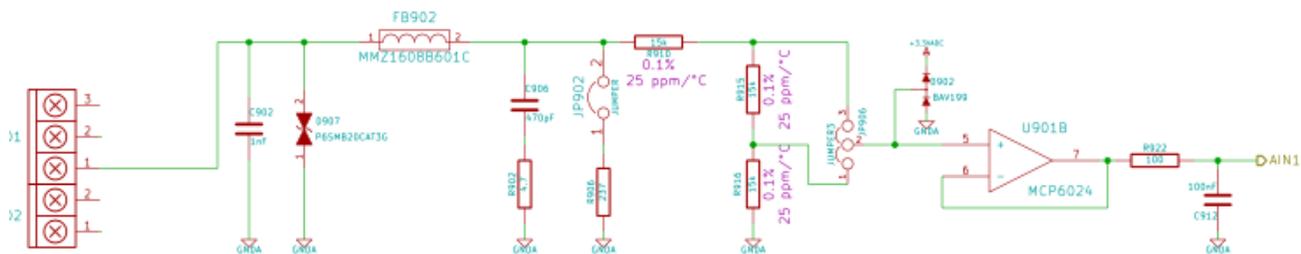


Imagen 2.31. Diseño de entrada analógica configurable.

En el circuito, si fuere un lazo de voltaje, la tensión entrante (0-10V) llega hasta el divisor resistivo de las 3 resistencias de 15K. Dicho circuito ofrece a su salida un tercio de la tensión de entrada. Con lo que la excursión será de (0-3.3V), cubriendo todo el rango útil del ADC. Como el operacional es Rail-to-Rail garantiza llegar a esos extremos sin alinealidades. En el modo de corriente, la misma circula principalmente por la resistencia de 237 ohm, generando una caída de 948mV (para 4mA) y 4.74V (para 20mA).

El divisor de las 3 resistencias de 15K cambia de configuración por los jumpers y da a su salida 2/3 de la de entrada. Con lo que se obtiene a la entrada del ADC: 632mV (para 4mA) y 3.16V (para 20mA).

Diseño de Salidas Digitales

Estas son las encargadas de comunicarse con los sistemas de control final o actuadores que efectúan acciones mecánicas o eléctricas. Estas permiten transmitir los resultados lógicos de las operaciones realizadas por el microcontrolador en función de las entradas en un ciclo de programa. Estas salidas por naturaleza de construcción electrónica pueden ser:

Salidas a Relé.

Por su función los PLC con salidas a relé son mayormente utilizados cuando la carga a controlar va directamente relacionada con sistema de conmutación, específicamente pueden ser, contactores, relés, solenoides, etc.

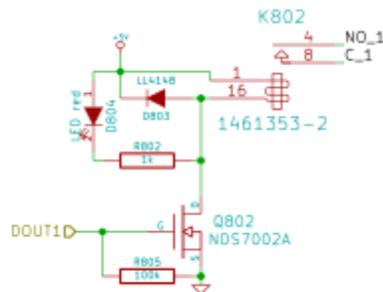


Imagen 2.32. Diseño de salidas a relé.

Este tipo de salida está restringida a ciertas frecuencias de conmutación, esto debido a que el relé por naturaleza un dispositivo electromecánico tiene un tiempo de retardo a la conmutación proporcional a la intensidad del campo magnético, un constante e

intermitente conmutación dañaría nuestro relé, estas, no permite modular salidas por ancho de pulso (PWM).

Salidas Transistorizadas (Open-Drain)

Su naturaleza de construcción está pensada para disponer de versatilidad al tipo de elemento final a controlar; esto significa que para manejar etapas de potencia podemos utilizar relés de acople para dicha tarea, si nuestro elemento final de control es un elemento de control de procesos continuos como un PWM o un tren de pulsos hacia un servomotor, es muy apropiado su uso.

Sin embargo estas salidas son muy delicadas, un buen diseño garantiza un buen comportamiento en disponibilidad de las mismas. Es común encontrarse con transistores MOSFET de alta capacidad en manejo de corriente (Aprox. 5A), aislados por Optoacopladores.

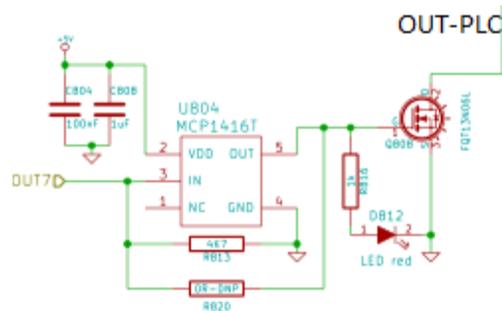


Imagen 2.33. Diseño de salidas Open Drain a transistor.

En este circuito de la imagen, se utiliza un MOSFET Driver para controlar el encendido de dichos MOSFET. A su vez, este driver sirve para limitar la corriente de conmutación del MOSFET a 1A, dado que la salida del driver llega hasta 5V, fijando la tensión de Gate a un máximo. Este tipo de salidas requieren que la carga se conecte entre una fuente externa y el Drain del MOSFET

Software Libre

En este apartado se pretende explicar por qué se hace uso de un software libre para el diseño de nuestro micro PLC, cuales son las ventajas y sus características. Para analizar este tema podemos hacernos la siguiente pregunta.

¿Qué es el Software Libre?

Cuando hablamos de Software libre nos estamos refiriendo a un software que respeta la libertad de los usuarios y la comunidad es decir que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

El software libre se ha desarrollado combinando las mejores virtudes de la investigación científica (cosmopolitismo, altruismo, transparencia, comunitarismo, meritocracia y objetividad) con las buenas prácticas de la eficiencia económica (libre competencia, innovación competitiva y reutilización de recursos) **(negocios, 2009, p. 4)**

El significado de software libre muchas veces se mal interpreta y se puede llegar a pensar que estos tipos de software son gratuitos, pero en realidad la palabra "Libre" va enfocada a la manipulación permitida que pueden hacer los usuarios a estos tipos de software.

Otra de las características de un software libre que influye mucho en nuestro proyecto de diseño del micro PLC es la facilidad que tiene el usuario para controlar el programa y todo lo que este realiza, de no haber hecho uso de un software libre nos hubiéramos encontrado con la dificultad de no tener el control sobre el programa.

Lo que provoca que el usuario se debe de acoplar a la condiciones del programa y a todas las condiciones que fueron previamente realizadas por un programador.

Para poder determinar que un software es libre debe de cumplir con cuatro libertades esenciales las cuales son:

Libertad 0: ES aquella que nos brinda la capacidad de poder ejecutar un programa como el usuario lo desee con cualquier propósito

Libertad 1: Por medio de esta libertad obtenemos la ventaja de poder estudiar cómo funciona el programa, y cambiarlo a nuestra conveniencia con el propósito que haga lo que el usuario pretende. El acceso al código fuente es una condición necesaria para ello.

Libertad 2: El propósito de esta libertad está basada en otorgar a las personas el derecho a poder redistribuir copias para ayudar a su prójimo

Libertad 3: Está enfocada en otorgar el permiso a los usuarios para poder distribuir copias de sus versiones modificadas a terceros

Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Software libre comercial

Al decir que un Software es libre no significa que no es comercial. Un programa libre debe de tener la característica de estar disponible para el uso comercial, la programación comercial y la distribución comercial.

La parte comercial se puede ver reflejada cuando algún usuario ha efectuado un pago para obtener copias de software libre, o puede haber

obtenido copias sin costo. Pero sin tener en cuenta cómo obtuvo sus copias, siempre tiene la libertad de copiar y modificar el software, incluso de vender copias.

La libertad de ejecutar el programa como se desee

Esta ventaja de un software libre nos proporciona una característica que se pretende alcanzar con nuestro proyecto. Al cumplir con esta ventaja nos brinda la opción de usarlo en cualquier tipo de sistema de computación, para cualquier tipo de trabajo y finalidad sin que exista obligación alguna de comunicarlo al programador ni a ninguna otra entidad específica

Como usuario somos libres de ejecutar el programa lo cual nos permite alcanzar nuestros propósitos, incentivando a las personas a realizar la fabricación de los micro PLC permitiendo también a las demás personas hacer uso libre de ejecutarlo para lo que necesite; así mismo no se tiene el derecho de imponerle nuestros propios objetivos a la otra persona.

La libertad de ejecutar el programa como se desee significa que al usuario no se le prohíbe o no se le impide ejecutarlo. Esto no tiene nada que ver con el tipo de funcionalidades que el programa posea, ni con su capacidad técnica de funcionar en un entorno dado, ni con el hecho de que el programa sea o no sea útil con relación a una operación computacional determinada.

La libertad de estudiar el código fuente y modificarlo

Para que las libertades 1 y 3 (realizar cambios y publicar las versiones modificadas), se debe tener acceso al código fuente del programa. Por consiguiente, el acceso al código fuente es una condición necesaria para el software libre

La libertad 1 nos permite la facilidad que el usuario pueda utilizar su versión modificada en lugar de la original.

Es importante aclarar que si una modificación constituye o no una mejora, es un asunto subjetivo. Si su derecho a modificar un programa se limita, básicamente, a modificaciones que puedan aportar una mejora, el programa no es considerado como libre.

La libertad de redistribuir copias si así lo desea: requisitos básicos

La libertad para distribuir (libertades 2 y 3) significa que las personas tiene la libertad para redistribuir copias después de haber hecho alguna modificación o incluso sin la necesidad de hacer modificaciones.

Con la implementación del software libre a utilizar para nuestro diseño de nuestro micro PLC esto es una ventaja ya que uno de nuestros propósitos es brindar a las personas una herramienta que se pueda desarrollar en un entorno industrial y a su vez desarrollarse en el área didáctica, dejando como opción realizar alguna mejora al proyecto y con la característica de poder reproducir estos micro PLCs.

Una de las ventajas del por qué se decidió utilizar un software libre es por la razón que para poder ejecutar el programa no es necesario realizar compra de licencia lo cual beneficia a muchos estudiantes para poder practicar e incrementar sus conocimientos de programación.

La libertad 3 incluye la libertad de publicar sus versiones modificadas como software libre. Una licencia libre también puede autorizar otras formas de publicación; en otras palabras, no tiene que ser una licencia con copyleft. No obstante, una licencia que requiera que las versiones modificadas no sean libres, no se puede considerar libre.

La libertad de redistribuir copias debe incluir las formas binarias o ejecutables del programa, así como el código fuente, tanto para las versiones modificadas como para las que no lo estén. (Distribuir programas en forma de ejecutables es necesario para que los sistemas operativos libres se puedan instalar fácilmente). Resulta aceptable si no existe un modo de producir un formato binario o ejecutable para un programa específico, dado que algunos lenguajes no incorporan esa característica, pero debe tener la libertad de redistribuir dichos formatos si encontrara o programara una forma de hacerlo.

Copyleft

El concepto de copyleft nos proporciona información de algunas reglas que las personas deben cumplir a la hora de distribuir el diferente software libre. Por medio del copyleft tenemos el respaldo de poder realizar las modificaciones al software libres pero aclarando que las personas que quieran hacer modificaciones al programa no pueden agregar ciertas restricciones que impidan que otras personas puedan ejecutar, ya que de esa manera se estaría perdiendo el concepto de software libre.

En el proyecto GNU usamos el copyleft para proteger legalmente las cuatro libertades para todos. Creemos que existen razones importantes por las que es mejor usar el copyleft. De todos modos, el software libre sin copyleft también es ético.

Reglas acerca del empaquetamiento y la distribución

Por medio de esta regla damos a conocer la forma correcta de realizar el empaquetamiento de una versión modificada, esta regla nos indica que una versión es aceptada si en ella no se limita la libertad para poder realizar otras versiones modificadas así como también la libertad que permite usar versiones modificadas en privado. Esta regla permite

que una licencia le obligue a cambiar el nombre de la versión modificada, eliminar el logotipo o identificar sus modificaciones como suyas, siempre y cuando todas estas modificaciones efectuadas no dificulte su publicación para otros usuarios.

Las reglas admisibles

Son aquellas reglas que se enfocan en cumplir la libre distribución para diferentes usuarios sin hacer restricción de personas, de tal manera podemos expresar que si una persona ha distribuido una versión modificada y uno de los programadores anteriores le solicita una copia, usted deba enviársela (tenga en cuenta que tal regla le sigue permitiendo optar por distribuir o no distribuir su versión).

Las reglas que obligan a suministrar el código fuente a los usuarios de las versiones publicadas también son admisibles.

Consideraciones legales

En esta parte se explica cuáles son las consideraciones que se deben de cumplir para que un software sea legal. Para que las libertades se cumplan deben ser permanentes e irrevocables siempre que usted no cometa ningún error. El software deja de ser libre cuando algún programador tiene el poder de revocar la licencia o de añadir restricciones a las condiciones de uso en forma retroactiva sin que haya habido ninguna acción de parte del usuario que lo justifique

Una licencia libre no puede exigir la conformidad con la licencia de un programa que no es libre. Así, por ejemplo, si una licencia requiere que se cumpla con las licencias de «todos los programas que se usan», en el caso de un usuario que ejecuta programas que no son libres este requisito implicaría cumplir con las licencias de esos programas privativos, lo cual hace que la licencia no sea libre.

Es aceptable que una licencia especifique la jurisdicción de competencia o la sede para la resolución de conflictos, o ambas cosas.

Licencias basadas en contrato.

En la actualidad existen algunas licencias que están sujetas a algunos contratos lo que permite que puedan imponer a las licencias un rango de mayores restricciones, esto puede generar que estas licencias sean inaceptablemente restrictivas lo que las convierte en licencias que no sean libres.

Esto significa que existen muchas maneras posibles de que tal licencia sea inaceptablemente restrictiva y que no sea libre.

Nos resulta imposible enumerar todas las formas en las que eso puede suceder. Si una licencia basada en un contrato restringe al usuario de un modo que no se puede hacer con las licencias basadas en el copyright, y que no está mencionado aquí como legítimo, tendremos que analizar el caso, y probablemente concluyamos que no es libre.

Cuando hable del software libre, hay que emplear los términos adecuados.

Cuando nos referimos al software libre es de especificar que no estamos hablando de un software que es regalado o que es de forma gratuito, ya que muchas personas relacionan el significado con el precio y no con las libertades que un software libre otorga para poder publicarlo o modificarlo siempre y cuando no se coloquen restricciones al software.

Es importante que el programador que ha hecho uso de algún software libre y haya realizado alguna modificación a este, proporcione un manual que sirva como orientación para otras personas que puedan utilizar este mismo software. Es de aclarar que dicho manual también

tiene que ser de forma libre ya que es una parte del que complementa al software libre

Con el diseño de fabricación de nuestro micro PLC se creara un manual de usuario con respecto al uso del software libre utilizado para la programación, brindando a las personas una orientación del manejo del software y ciertas funciones que pueden ejecutar.

¿Por qué Proporcionar un software libre para programación del PLC?

El objetivo de hacer uso de un software es para que las personas tengan acceso al software sin la necesidad de comprar licencias y también brindamos la oportunidad para que las personas puedan realizar mejoras al software y a su vez darlas a conocer a otros.

Muchas personas no están de acuerdo con proporcionar un software libre ya que consideran una pérdida económica cuando los usuarios copian programas por su cuenta. Pero en realidad eso no genera un efecto directo sobre el propietario, la única forma en la que un propietario puede decir que ha tenido pérdidas es que si la persona copio su programa en lugar de hacer la copia le hubiese comprado una a él.

Costos de adquisición.

Cuando hablamos de la economía en relación al software libre, los programadores propietarios de software se basan en que las necesidades de las personas los llevan a fabricar diferentes tipos de software aún más si las personas estarían pagando por dichos softwares

La mayoría de las personas solo necesitan que exista una mayor producción de software no importándoles si estos software tienen propietarios o son libres, el problema se da cuando un propietario

restringe su software o algunas funciones con el propósito que las personas paguen para adquirirlo.

En realidad las personas necesitan información que esté verdaderamente a disposición de todos, que tengan la capacidad de leer, corregir, adaptar, y mejorar, no solamente ejecutar el software.

Usted merece poder cooperar abierta y libremente con otras personas que usan software. Merece poder aprender cómo funciona el software, y utilizarlo para enseñar a sus alumnos. Merece poder contratar al programador que prefiera para que corrija el programa cuando falle.

Software Libre de Programación de PLC con microcontroladores

En la actualidad existen muchos proyectos de diseño de software que han sido lanzados con el objetivo de reinterpretar la programación de los microcontroladores.

El objetivo de estos es compilar una serie de datos hexadecimales programados en lenguaje nemotécnico, en unos y ceros o mejor conocido como lenguaje de máquina, mediante el uso de librerías y funciones permitidas por la arquitectura RISC del microcontrolador.

Existe una gama de lenguajes de programación que permiten compilar estas instrucciones, por ejemplo:

- Lenguaje bajo nivel, Ensamblador
- Lenguajes de Alto Nivel, Texto estructurado (Estilo C/C++)
- Lenguajes Gráficos

Industrialmente hablando todos estos lenguajes ya se utilizan en los PLC, realizando una homología, podemos decir:

- Leguaje ensamblador = LI o Lista de Instrucciones
- Texto estructurado (C++)= ST o Texto Estructurado
- Lenguajes Gráficos (Labview) = FBD Bloque de funciones, LAD Escalera

Programación Escalera (KOP) en Microcontroladores.

Como ya habíamos mencionado, desde el inicio de los sistemas basados en microcontroladores, el lenguaje de programación que más tenía presencia era el ensamblador, dadas las características del manejo versátil de registros y operaciones específicas con los mismos, pero ¿hasta qué punto la versatilidad era un término válido en este lenguaje de programación?

Siendo específicos la estructuración de un programa en ensamblador era complejo, el control de flujo de los datos debía ser crítica, por lo tanto implicaba tiempos de programación extensos, etc. Sin embargo, con el paso del tiempo en el desarrollo de los lenguajes de programación como pascal, C, C++, entre otros lenguajes de alto nivel, vinieron a definir el concepto de estructuración de un programa, con la eliminación de sistemas o funciones peligrosos (debido a que estas funciones limpiaban la pila (microprocesadores) o registros (microcontroladores) arbitrariamente sin definir un condición de transición de datos segura) en el control de los algoritmos como lo era en sus nemotécnicos JMP (jump), Brake, entre otros.

Cuando la industria de la electricidad y la electrónica, avanzó en el diseño de sistemas y procesos, controlados por microprocesadores, PLC, se buscó que el método o la técnica de diseño de los algoritmos fuera similar, o bien igual, a la de los sistemas de interlocks a relés, o bien para los electrónicos lista de instrucciones; de allí que surgen lenguajes particulares de los autómatas programables, considerados en

el estándar IEC 61131 los cuales define los cinco lenguajes de programación, los IL o STL, SFC, FBD, ST y LAD.

Este último el más predominante entre los sistemas de programación, debido a su familiarización a los sistemas de control basados en relés. Razón por la que es común ver su uso en la enseñanza de programación de Automatas Programables, dada la previa enseñanza de los sistemas a relés en las instituciones.

Para la programación de los sistemas basados en microcontroladores, en lenguaje de escalera podemos describir los siguientes con sus alcances y limitaciones más destacables:

Tabla 2: Comparación de funciones de softwares y sus plataformas compatibles

Software	Funciones	Microcontroladores Plataformas Compatibles
LDmicro	Tamaño de programa según espacio de microcontrolador. Compilación de archivo .hex Simulación de programación. Configuración libre de pines I/O. Múltiple gama de microcontroladores posibles a programar (Microchip y AVR)	PIC16F628 (A) PIC16F88 PIC16F819 PIC16F877 (A) PIC16F876 (A) PIC16F887 PIC16F886 ATmega128

	No posee función downloader	ATmega64 ATmega162 ATmega32 ATmega16 ATmega8
Waltech Ladder Maker	Tamaño de programa según espacio de microcontrolador. Pines I/O predeterminados por programa. Cuenta con función downloader. NO es posible la simulación del programa. Solo se programan plataformas Arduino	Arduino UNO Arduino NANO Arduino MEGA
SoapBox Snap	Tamaño de programa según espacio de microcontrolador. Configuración libre de pines I/O Posee simulación de circuitos en TR Cuenta con función Downloader Utiliza Phidgets extras para control de puertos COM y simulación.	Arduino UNO Arduino NANO Arduino MEGA

	Solo programa plataformas Arduino.	
--	------------------------------------	--

Software de programación de Micro PLC: Waltech Ladder Maker

Es un software de código abierto diseñado para la programación en escalera de las plataformas con microcontroladores AVR, en placas Arduino Mega, Nano y UNO; diseñado por los ingenieros: Karl Walter y Mark Walter, el cual su objetivo era utilizar estos microcontroladores, como sistemas de autómatas programables en un lenguaje industrial como KOP o escalera, en el diseño de sus proyectos de aplicaciones propias a la industria.

Su código puede ser manipulado y recompilado en PYTHON plataforma de diseño de aplicaciones con lenguaje de alto nivel. Particularmente en el proyecto debido al tiempo, competencias y capacidades del equipo no se ampliaron las posibilidades de modificaciones o mejoras al software para su uso, como por ejemplo configurar los puertos de entradas y salidas, ampliar la gama de instrucciones disponibles, o incluso incorporar otro lenguaje de programación, etc. Que su código sea abierto abre las puertas para obtener la base de un proyecto de mejora del software.

Sin embargo el software que se distribuye actualmente, contiene herramientas básicas, sencillas en uso y familiarización, que permiten realizar programas sencillos, básicos, y que contando con un buen hardware de interfaz puede lograrse una operación eficiente del microcontrolador como un PLC.

Nota importante sobre el software, es que no se permiten realizar simulaciones del programa, ni monitoreo en tiempo real.

El software realiza 3 operaciones al momento que se activa la función de compilación:

1. Generar el código en C de la programación en KOP, legible para una programación en IDE de Arduino.
2. Generar código máquina que se cargara al microcontrolador, para realizar las funciones necesarias.
3. Realizar la función de Download de descarga de código al microcontrolador.

Estructura de Software

Barra de Menú



Se tiene una barra de menú simple, con Archivo, Edición, Hardware y Ayuda.

Importante a destacar en el software es el fichero de Hardware, el cual al desplegarse, muestra todas las plataformas posibles a ser utilizadas para su programación, solo soporta las plataforma de Arduino UNO, NANO y MEGA.

Además cuenta con una opción llamada Waltech, esta compila el programa editado, y genera únicamente el código, sin invocar a ningún puerto de COM, donde tengamos nuestro dispositivo y efectué una descarga al mismo.

Además cuenta con una opción de test de puerto COM, en este caso el dispositivo conectado al puerto USB, de COM 1 al COM 5

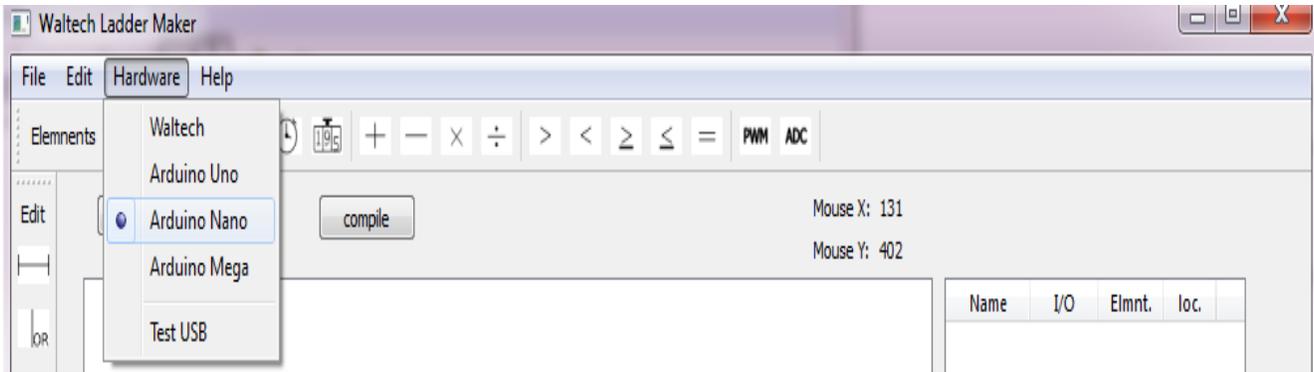


Imagen 2.34 Encabezado de Interfaz de Software Waltech Ladder Maker

Interfaz de Línea de Comandos

Algo muy importante a destacar y tomar en cuenta del software es de contar con una pantalla de compilación en tiempo real el cual permite:

1. Generar el código de programación en lenguaje C.
2. Registrar las acciones, modificaciones en la programación.

```

C:\Users\OSCARA-1\AppData\Local\Temp\Rar$EXa0.240\program\main.exe
while (<sample_count < OVERSAMPLES)<asm volatile <"nop";>>wait for comple
tion
ADCSRA &= (1<<ADSC); // stop the ADC
return (<ADCtotal/OVERSAMPLES>; //max osamples = 63 otherwise will overflow to
tal register with 10 bit adc results
>
int main()
<
//set up ADC
ADCSRA |= (1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0); // sets adc clock prescaler
to 128 //checked
ADMUX |= (1<<REFS0); // sets ref volts to Vcc
ADCSRA |= (1<<ADIF); // enable ADC conversion complete interrupt
ADCSRA |= (1<<ADSCIE); // set to auto trigger (free running by default)
DDRD |= (1<<5);
DDRD |= (1<<6);
DDRD |= (1<<7);
DDRB |= (1<<0);
DDRB |= (1<<3);
DDRB |= (1<<4);
DDRB |= (1<<5);

PORTC |= (1<<4);
PORTC |= (1<<5);
PORTD |= (1<<2);
PORTD |= (1<<3);
PORTD |= (1<<4);

//set up loop timer:
TIMSK0 |= (1<<TOIF0); // overflow capture enable
ICNT0 = 101; // start at this
TCCR0B |= (1<<CS10)|(1<<CS12); // timer started with /1024 prescaler
sei();
uint8_t y = 1;
while (1)
<
if (timerOF == 1)
<
timerOF=0; //reset timer flag
ICNT0 = 101; // start at this
//inputs:
//Start of Ladder:
//outputs:
>
>
>
saving C and Compiling
checked platform win32
current dir: C:\Users\OSCARA-1\AppData\Local\Temp\Rar$EXa0.240\program

```

Imagen 2.35 Interfaz de línea de comandos.

Instrucciones disponibles en software

Dado que es un lenguaje grafico de programación las instrucciones posibles vienen distribuidas en una barra de herramientas.

Barra de Herramientas



Dentro de las instrucciones que se pueden manejar con este software son las siguientes:

**TABLA 3: Descripción de las instrucciones del software
Waltech Ladder Maker**

Botón/Objeto Grafico	Descripción
	Contacto Abierto
	Contacto Cerrado
	Bobina
	Detector de Flanco Negativo
	Temporizadores BCD (UINT) Tipo: On Delay.
	Contadores (UINT) Tipo: Ascendente
	Operación SUMA (INT)

	Operación Resta (INT)
	Operación Multiplicación (INT)
	Operación División (INT)
	Comparación de Datos Mayor que (INT)
	Comparación de Datos Menor que (INT)
	Comparación de Datos Mayor o Igual que (INT)
	Comparación de Datos Menor o Igual que (INT)
	Comparación Igual que (INT)
	Salida PWM
	<p>Entrada ADC (Configuración de Entradas Analógicas)</p> <p>Voltaje de Ref. Inferior: 0.0 V</p> <p>Voltaje de Ref. Superior: 5.0 V</p> <p>Resolución Digital: 10 bits</p> <p>Resolución Analógica: 4.883mV</p> <p>Nivel de cuantificación: 1024 Valores</p>

Tabla 4: Barra de edición en programación escalera

Botón/Objeto Grafico	Descripción
	Insertar Segmento
	Insertar ramal paralelo en OR
	Aumentar espacio en segmento
	Reducir espacio extra en segmento
	Eliminar elemento

Observaciones del Software

A continuación una serie de aspectos a tomar en cuenta al momento de diseñar un programa en escalera, para evitar inconvenientes en compilación o inconsistencias de comportamiento:

1. No es posible crear funciones, bloques de funciones, bloques de datos, debido al espacio de memoria remanente y de trabajo.
2. Ni enlaces de comunicación a medios de redes industriales
3. La resolución de las señales analógicas están determinadas por una resolución de 10 bit, hasta 1024 valores, en un rango de 5VDC.
4. No es posible utilizar bloques que transferencia de datos, pj. MOV.

5. El tamaño del programa está restringido a la capacidad de memoria del microcontrolador, para este caso de un Arduino NANO, la EEPROM es de 1KB.
6. Resulta muy difícil determinar un orden lógico y de estructura de un programa, en el MicroPLC, dado que no es posible la programación estructurada.
7. No son posibles los ramales paralelos en ramales paralelos. Inhibición de un ramal OR con otro contacto en paralelo de la función.
8. Se dispone con un contador ascendente de 16 bits, mono canal, de 65,535 valores de tipo UINT, sin embargo no es un bloque de función con entradas de control Reset, etc.
9. Es posible utilizar un timer con capacidad de retardo a la conexión de 655.34 segundos la base de la cuenta es fija, no es posible cambiar minutos y horas.
10. Respecto a las operaciones aritméticas el comportamiento de una división por cero es indeterminada y arroja un valor de cero como resultado.

La operación resta permite un conteo de $-32767/+32767$ es decir que para todas las operaciones aritméticas se admiten datos de tipo int (enteros)

Solo se cuenta con un evaluador de flanco negativo y solo es posible realizar comentarios a la función

Los ADC son los bloques que nos permiten configurar los puertos de entrada analógica directamente en el micro controlador en niveles de 0 a 5v y en el PLC de 4-20mA la técnica de acondicionamiento fue la de un seguidor de tensión mediante un amplificador operacional. En el

programa la cantidad de puertos configurables son 4 con una resolución de 10 bits es decir de hasta 1024 diferentes valores.

CAPITULO 3

METODOLOGIA DE LA INVESTIGACION

METODOLOGIA DE LA INVESTIGACIÓN

La metodología de la investigación nos permite hacer uso de una serie de métodos y técnicas de rigor científico que se aplican de forma ordenada y sistemática durante el proceso de investigación para alcanzar un resultado teóricamente válido. De esta manera la metodología es la base que nos guía a poder desarrollar un estudio de investigación cumpliendo una serie de pasos para poder cumplir con todos los procedimientos y requisitos que debe de tener una investigación.

La manera de cómo ejecutamos la metodología de la investigación es siguiendo una serie de pasos que nos permitieron analizar de mejor manera la problemática y determinar las mejores ideas que se podrían adquirir para dar una solución efectiva con fundamentos teóricos y con la capacidad de implementarse y poder presentar un análisis de resultados de toda la investigación.

PASOS A SEGUIR PARA DESARROLLAR LA METODOLOGIA DE LA INVESTIGACION

- Concebir la idea a investigar
- Plantear el problema de investigación
- Elaborar el marco teórico
- Definir el tipo de investigación a desarrollar
- Seleccionar el diseño apropiado de investigación
- Selección de la muestra
- Recolección de datos
- Analizar los datos
- Presentar los resultados

Con el propósito de poder alcanzar los objetivos propuestos en el proyecto hemos determinado desarrollar el tipo de metodología que más se involucre con nuestro proyecto y que nos oriente por medio del conocimiento a comprobar la veracidad de la teoría y dar a conocer los beneficios de la implementación de estas teorías.

La metodología que nosotros elegimos para poder desarrollar nuestra tesis es el tipo de metodología **"Científica aplicada tecnológica"**

Este tipo de metodología se basa en al proceso que busca convertir el conocimiento puro, es decir teórico, en conocimiento práctico y útil para la vida de la civilización humana.

"cuando los conocimientos adquiridos en la ciencia aplicada, se trasladan al campo de lo concreto con el fin de configurar un proceso o de fabricar un artefacto entonces entramos en el campo de la tecnología'' (Cegarra, 2014, p. 14)

En nuestra investigación lo que buscamos es obtener toda la información teórica que esté relacionada con la fabricación de un PLC por medio de la implementación y desarrollo de un microcontrolador utilizando un software libre con el fin de poder crear un prototipo que pueda beneficiar a un grupo de estudiantes de carreras de mecatrónica y electrónica que están empezando a desarrollar su lógica para poder programar un PLC.

Objetivo de la metodología: Obtener una mayor consolidación del conocimiento humano, a través de la puesta en práctica de éste, proceso que se traduce en el crecimiento del saber científico y humano, al comprobar la veracidad de los conocimientos, y además beneficiarse utilitariamente de éste.

Por medio del desarrollo e implementación del prototipo a fabricar basado en un microcontrolador de software libre se trata de aportar un conocimiento de conceptos y funcionabilidad de los PLC con la capacidad de comprobar dicha afirmación a través de la práctica que los estudiantes pueden tener previamente para poder familiarizarse con la lógica de programación y los lenguajes industriales que manejan los PLC industriales.

La metodología científica aplicada tecnológica está orientada a la producción de conocimientos y métodos que vengán a mejorar o hacer mucho más eficiente el sector productivo de bienes o servicios, buscando imprimir en la vida del humano promedio bienestar, a través de un medio productivo mucho más eficiente. Igualmente, puede tener una fase teórica e investigativa, así como otra experimental, que lleve incluso a la elaboración de prototipos.

Este tipo de metodología nos ayuda a evaluar en qué manera nosotros podemos hacer más eficiente la enseñanza de la institución donde se pretende implementar el prototipo y reducir el tiempo de aprendizaje de los alumnos que están por iniciar las materias de programación de PLC

Otra metodología que nos va a servir de guía para poder realizar nuestra investigación es la metodología de tipo **"participativa"**

Es un estudio que surge a partir de un problema que se origina en la misma comunidad, con el objeto de que en la búsqueda de la solución se mejore el nivel de vida de las personas involucradas.

En el caso de nuestro estudio de investigación el problema que se está tratando de resolver es que los alumnos que están por empezar a recibir materias de programación de PLC ya posean un conocimiento básico del

funcionamiento y lógica por medio de nuestro prototipo reduciendo así el uso de controladores industriales que requieren licencias en los estudiantes que empiezan a conocer sobre la programación dejando así los PLC para los estudiantes que estén por cursar las materias de programación intermedia y avanzada para poder evitar errores y daños a estos equipos.

Dentro de la investigación participativa se pueden encontrar:

Estudio de casos: Es el estudio de sucesos que se hacen en uno o pocos grupos naturales.

En el caso de nuestra investigación el estudio se realiza a los estudiantes de mecatrónica y electrónica que en su curricular aparezca la materia de programación de PLC y también nos enfocamos en los docentes que imparten esta materia para poder determinar en qué medida este grupo de personas se vería beneficiada por medio de la implementación del prototipo.

Estudio Etnográfico: Es una investigación en la cual el investigador se inserta, camuflado en una comunidad, grupo o institución, con el objeto de observar, con una pauta previamente elaborada.

Lo que se pretende en este punto es realizar la investigación en los laboratorios donde se imparten las materias de programación de PLC de ITCA-FEPADE para visualizar en qué medida beneficiaría la implementación de nuestro prototipo en los estudiantes y docentes que se encuentran en el laboratorio, esto aplicaría no solo para mecatrónica sino también para la carrera de eléctrica y electrónica esto para tener una mejor muestra y con esto demostrar el beneficio multidisciplinario del proyecto.

TABLA 5: Encuesta Proyecto: PLC – Software y Hardware Libre

Pregunta	Si	Un Poco	No
1. Conoce que es un PLC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Tiene conocimientos de programación de un PLC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Conoce que es un Arduino	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Tiene conocimientos de programación de Arduino	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Conoce el lenguaje de programación Ladder en un PLC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. Durante el desarrollo del módulo de Lógica y Programación recibió clases de Programación KOP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Durante el desarrollo del módulo de Lógica y Programación recibió clases de programación de Arduino	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Le gustaría que durante el módulo de Lógica y programación se diera una introducción a PLC, Lenguajes de Programación KOP, etc.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. Conoce de la aplicación de hardware y software libre para fabricar un PLC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. Conoce de la capacidad del uso microcontroladores para construir un PLC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. Conoce de la aplicación de Arduino para fabricar PLC de hardware libre	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12. Conoce de la aplicación de Software Libre para programar un PLC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13. Conoce de la aplicación de Software libre para poder crear programas con lenguaje KOP para programación de PLC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14. Conoce de proyectos de aplicación de Software y Hardware Libre para fabricación de PLC en el país	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15. Considera necesario el impulsar el desarrollo de investigaciones en ITCA FEPADE sobre este tema	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Para dar a conocer ciertos beneficios y características de nuestro prototipo nos apoyaremos también de la **"investigación acción"**

Se centra en generar cambios en una realidad estudiada y no coloca énfasis en lo teórico. Trata de unir la investigación con la práctica a través de la aplicación, y se orienta en la toma de decisiones y es de carácter ideográfico.

Esta investigación la vamos a implementar para dar a conocer la capacidad de control que se puede llegar a tener por medio de nuestro prototipo incluso alcanzando funciones y habilidades parecidas a las de un PLC, así como ciertas normativas de construcción y funcionabilidad industrial que pueda ser competente de forma didáctica y se puedan obtener resultados positivos con su implementación en los estudiantes de mecatrónica y electrónica de ITCA-FEPADE.

En nuestra investigación también utilizaremos el tipo de **enfoque cualitativo**

"Se fundamenta más en un proceso inductivo (explorar y describir y luego generar perspectivas teóricas) va de lo particular a lo general" (Sampieri, Collado, & Baptista, 2014, p. 49)

En un típico estudio cualitativo, el investigador entrevista a una persona, analiza los datos que obtuvo y saca algunas conclusiones; posteriormente, entrevista a otra persona, analiza esta nueva información y revisa sus resultados y conclusiones; del mismo modo, efectiva y analiza más entrevistas para comprender lo que busca. Es decir, procede caso por caso, dato por dato, hasta llegar a una perspectiva más general.

Para el caso de nuestro trabajo de investigación hacemos uso del enfoque cualitativo para poder realizar una serie de encuestas a maestros y alumnos de carreras de electrónica y mecatrónica de ITCA FEPADE que en su curricula aparezca la materia de programación de PLC.

Con el objetivo de poder determinar el impacto que puede llegar a tener nuestro prototipo y el grado de aceptación por parte de las personas involucradas en el tema

El enfoque cualitativo nos permitirá reunir una serie de datos, perspectivas y diferentes opiniones o puntos de vistas que los estudiantes y maestros pueden tener hacer de la fabricación de nuestro prototipo incluso se puede llegar a tener conclusiones que pueden aportar nuevas ideas para mejorar o facilitar aún más la fabricación y operación de nuestro prototipo.

CAPITULO IV

ALCANCES -ANALISIS DE RESULTADOS

Analizando toda la información recopilada en el proceso de investigación podemos determinar una serie de resultados que se esperan obtener con la realización de nuestro proyecto entre los cuales están:

1-Poder incentivar a las instituciones educativas sobre el seguimiento, desarrollo e implementación del uso de microcontroladores para ejecutar diferentes tipos de controles que sean capaces de realizar una aplicación ya sea para el área didáctica de sus estudiantes o para entornos industriales.

2-Brindar una herramienta eficiente a los estudiantes que les permita realizar sus prácticas de laboratorio y aportar el conocimiento para poder desarrollar su lógica de programación y comprender el funcionamiento de los PLC

3-Facilitar el uso del software libre implementado para la programación del micro PLC, evitar una mala manipulación del software que pueda dañar el equipo, todo esto se pretende llevar acabo con la realización de un manual didáctico para orientar a las personas que deseen hacer uso del software libre.

Con la fabricación de nuestro Micro PLC nosotros pretendemos beneficiar al sector estudiantil directamente a los estudiantes que estén estudiando una carrera técnica en donde es necesario llevar la materia de programación de PLC, indirectamente también se busca beneficiar a los maestros que imparten estas materias por la razón que se les brinda otra herramienta que estaría a su disposición para comenzar a enseñar sobre la programación de PLC y de esa manera crear las bases del conocimiento en sus estudiantes que les servirá posteriormente al llevar otras materias relacionadas al tema logrando un rápido aprendizaje y que les permita a los estudiantes poder demostrar sus conocimientos por medio de la práctica.

El potencial que nosotros podemos analizar sobre nuestro proyecto es que tendrá la capacidad y libertad de otorgar a las personas, los diseños realizados para la fabricación del micro PLC, así como también los equipos y accesorios electrónicos necesarios para su fabricación, brindando a las personas la oportunidad de reproducir estos Micro PLC, Conociendo su estructura interna y con la capacidad de poder hacer una mejora ya sea en el hardware o software para desarrollar diferentes aplicaciones didácticas y agregar características que faciliten el control y beneficien a otras personas.

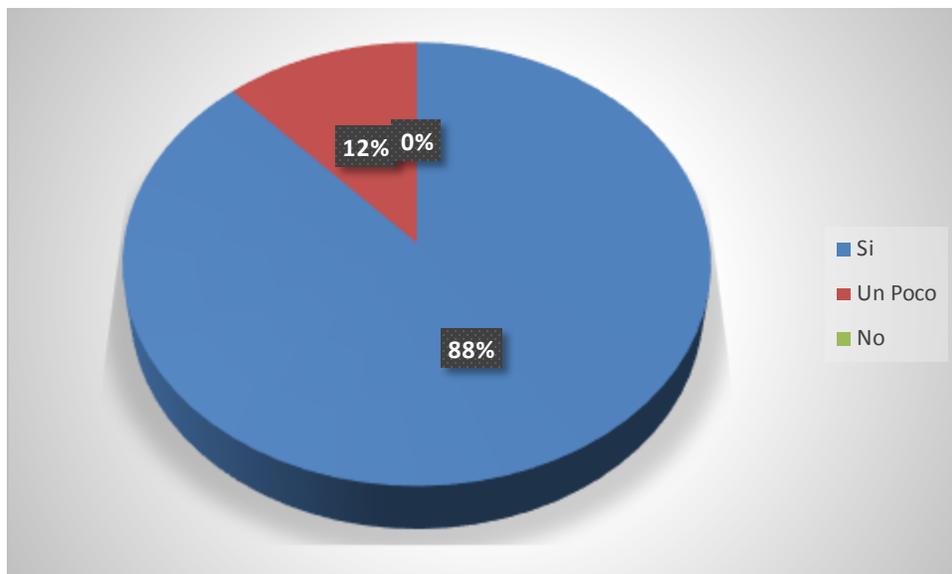
Resultados de aplicar la metodología participativa y el enfoque cualitativo

Dentro del desarrollo de nuestro trabajo de investigación decidimos aplicar esta metodología y recabar datos de un grupo de Técnico en Mecatrónica de Segundo año que actualmente está recibiendo el módulo de Programación de PLC, y asimismo vienen de recibir el módulo de lógica y programación que es donde se pretende implementar nuestro proyecto, se les proporciono una encuesta con 15 puntos importantes a tomar dentro de nuestro proyecto final (anexo el modelo de la encuesta junto con las encuestas contestadas)

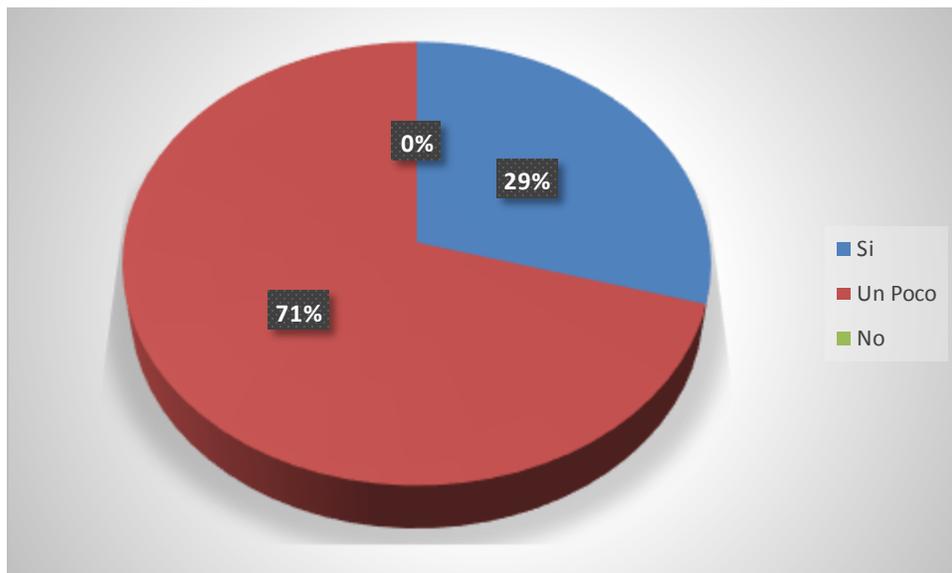
Se le explico al grupo de alumnos que se desea realizar un prototipo de PLC que se pudiese usar de forma didáctica y que este a su vez esté disponible para que ellos pueda realizar el ensamble del mismo con partes de fabricación local basando su programación en lenguaje KOP por medio de un software libre disponible para descarga.

A continuación los resultados:

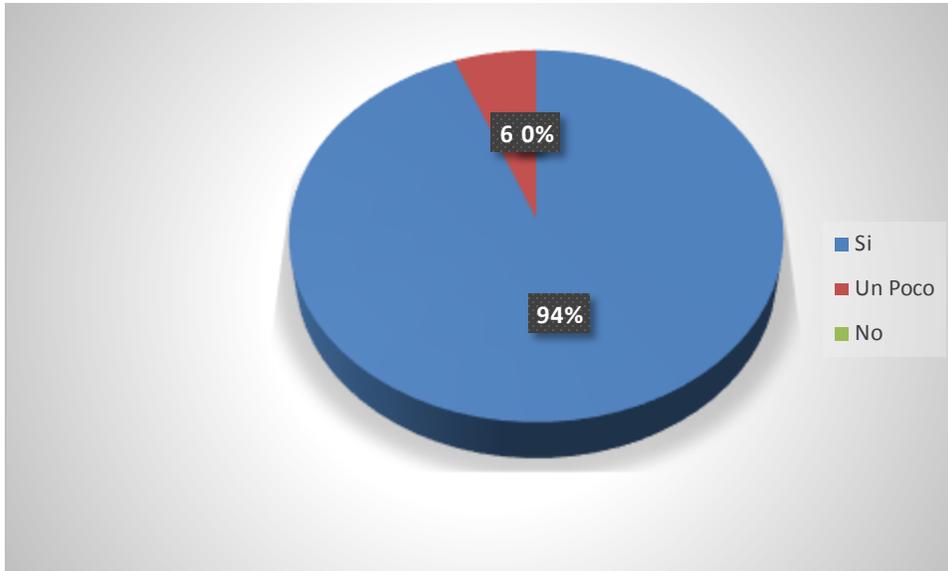
1. Conoce que es un PLC



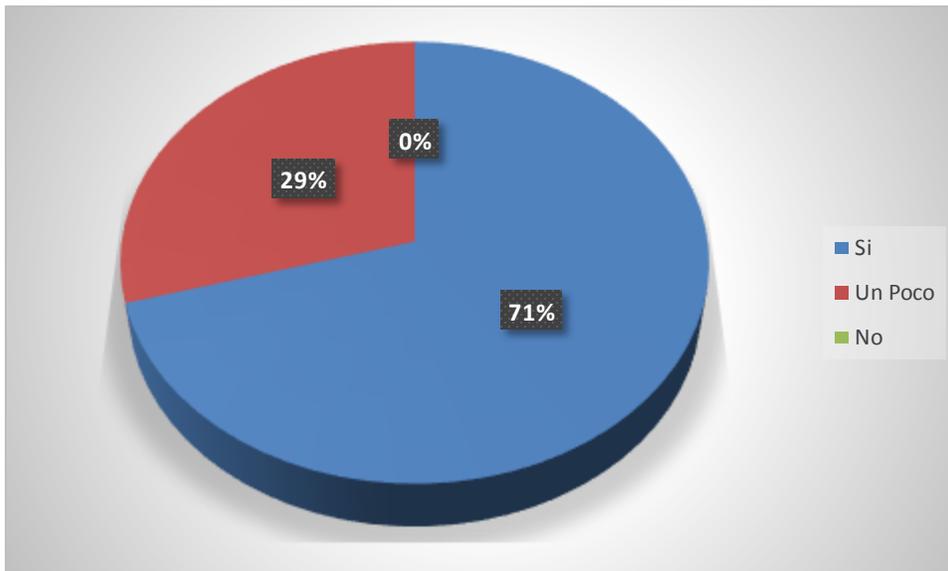
2. Tiene conocimientos de programación de un PLC



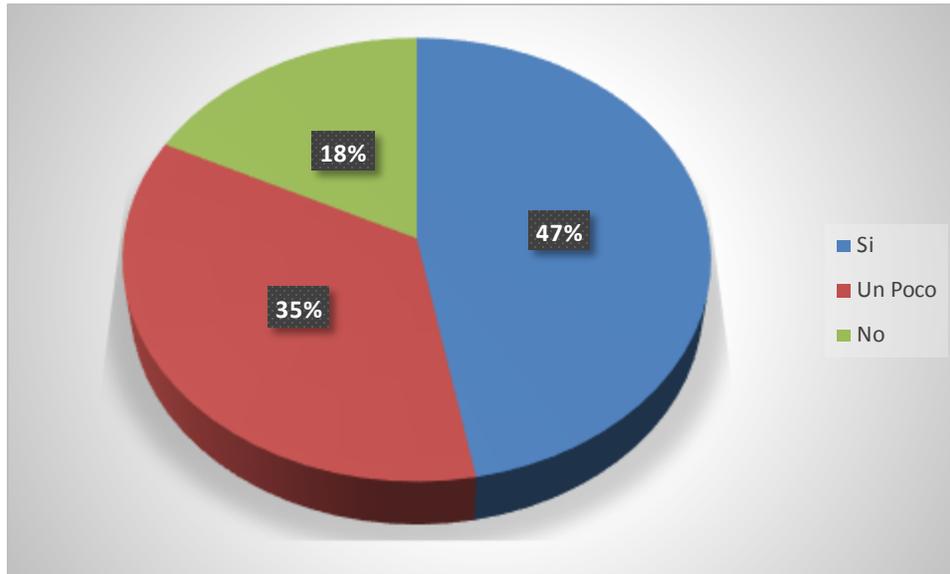
3. Conoce que es un Arduino



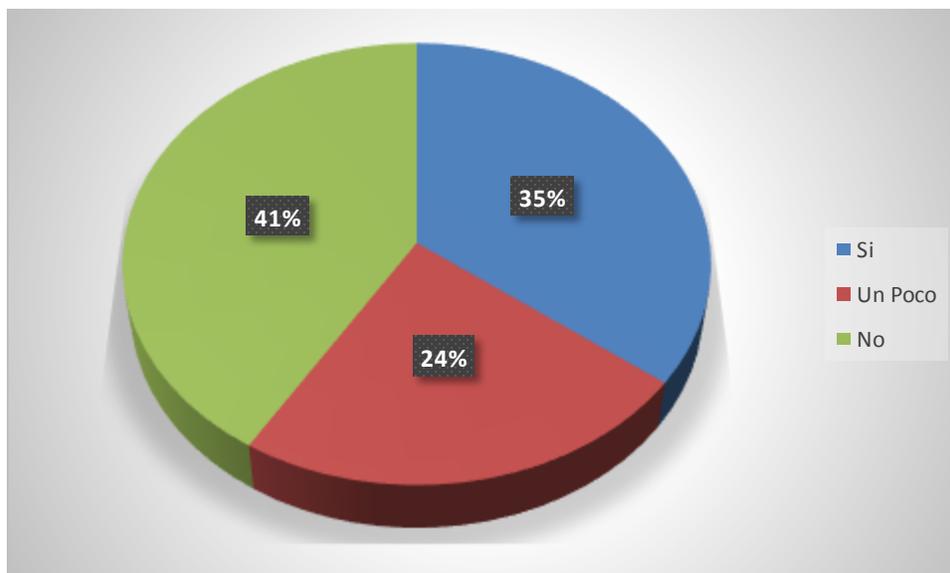
4. Tiene conocimientos de programación de Arduino



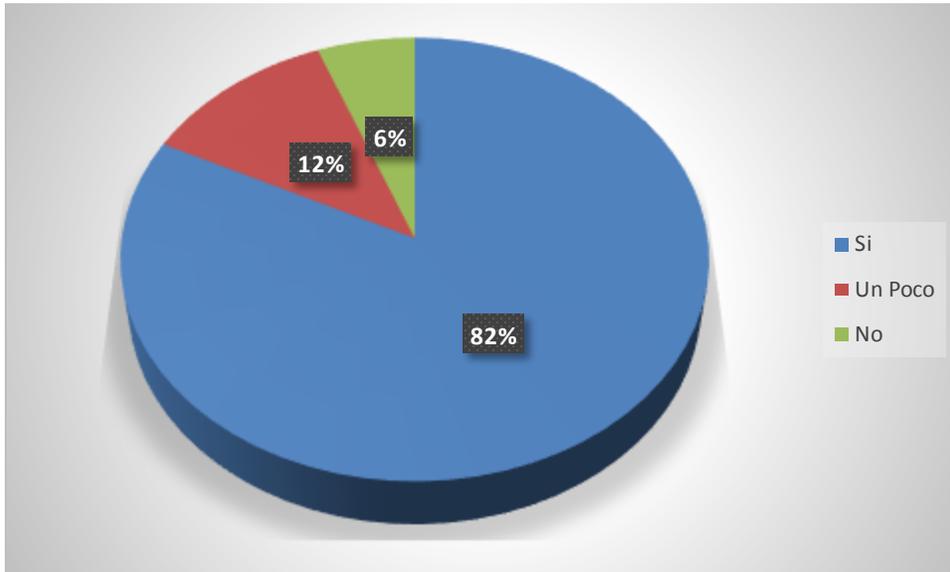
5. Conoce el lenguaje de programación Ladder en un PLC



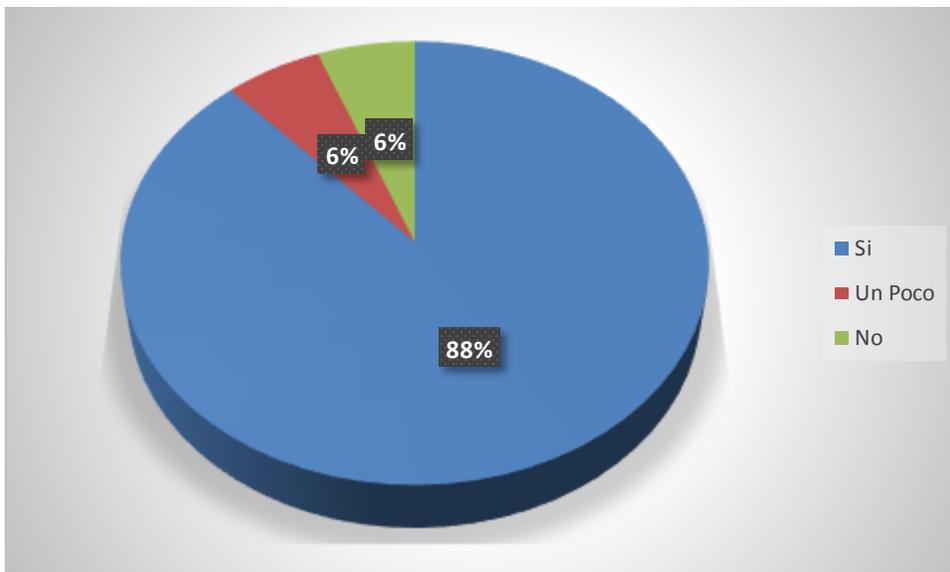
6. Durante el desarrollo del módulo de Lógica y Programación recibió clases de Programación KOP



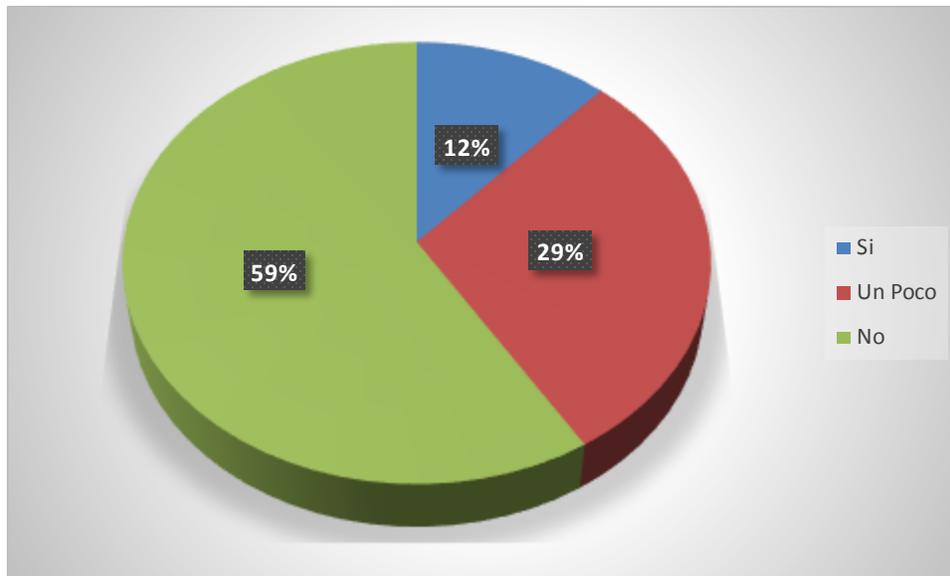
7. Durante el desarrollo del módulo de Lógica y Programación recibió clases de programación de Arduino



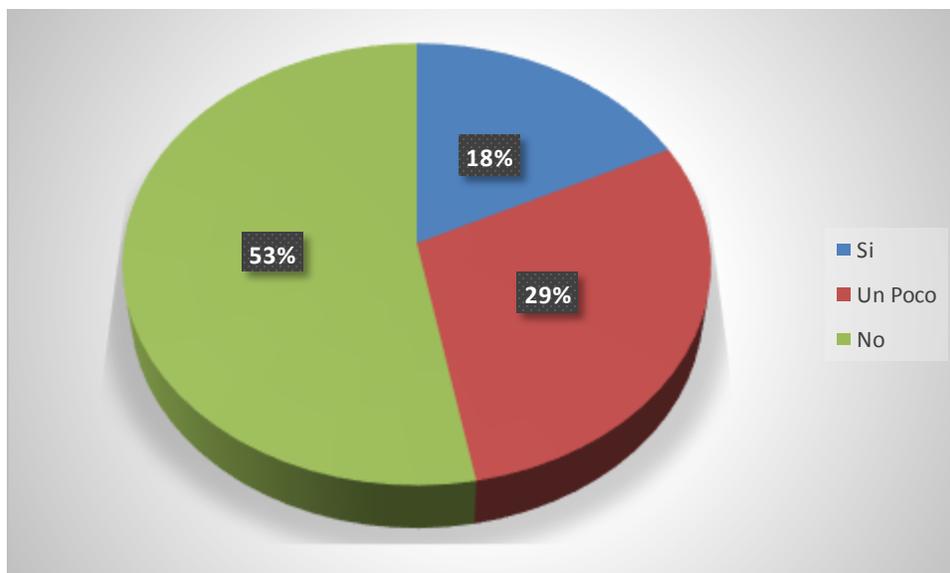
8. Le gustaría que durante el módulo de Lógica y programación se diera una introducción a PLC, Lenguajes de Programación KOP, etc.



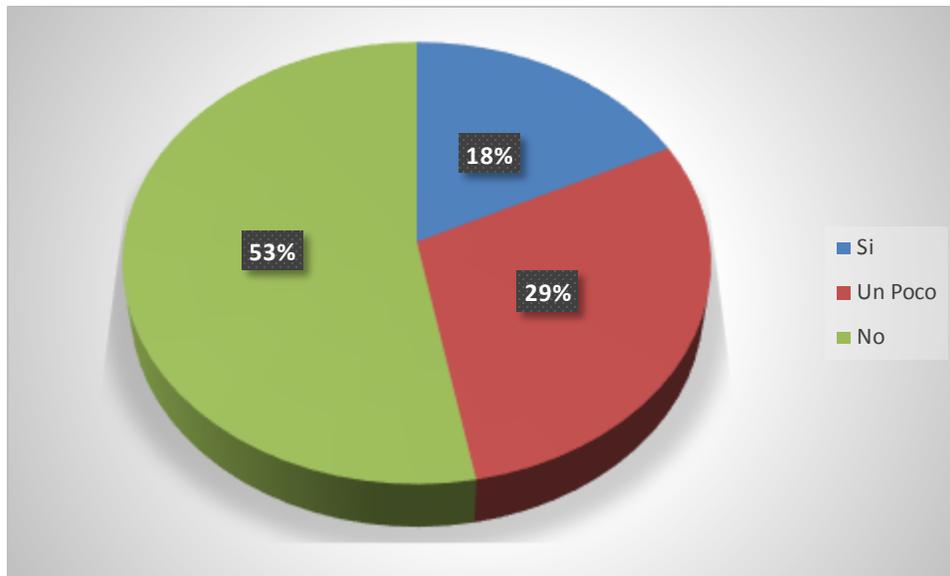
9. Conoce de la aplicación de hardware y software libre para fabricar un PLC



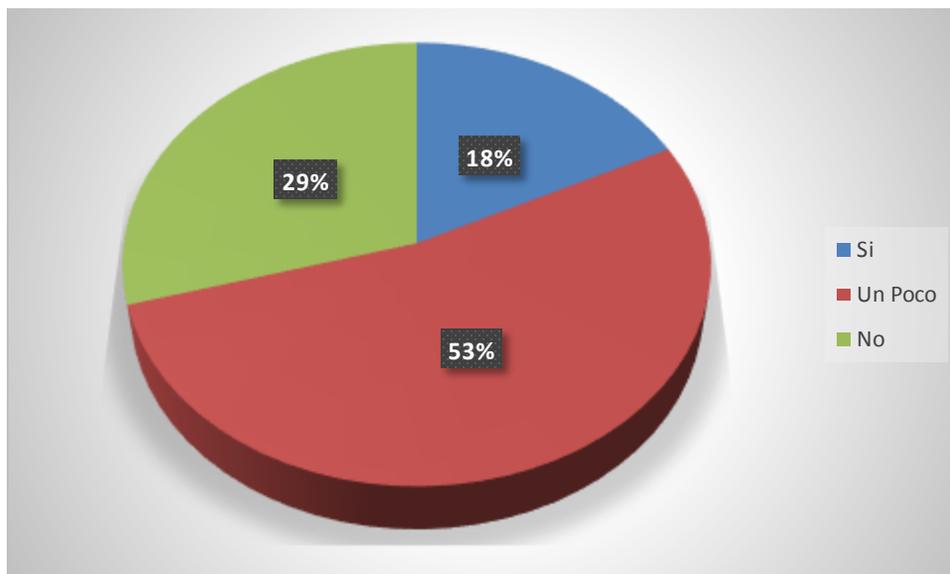
10. Conoce de la capacidad del uso microcontroladores para construir un PLC



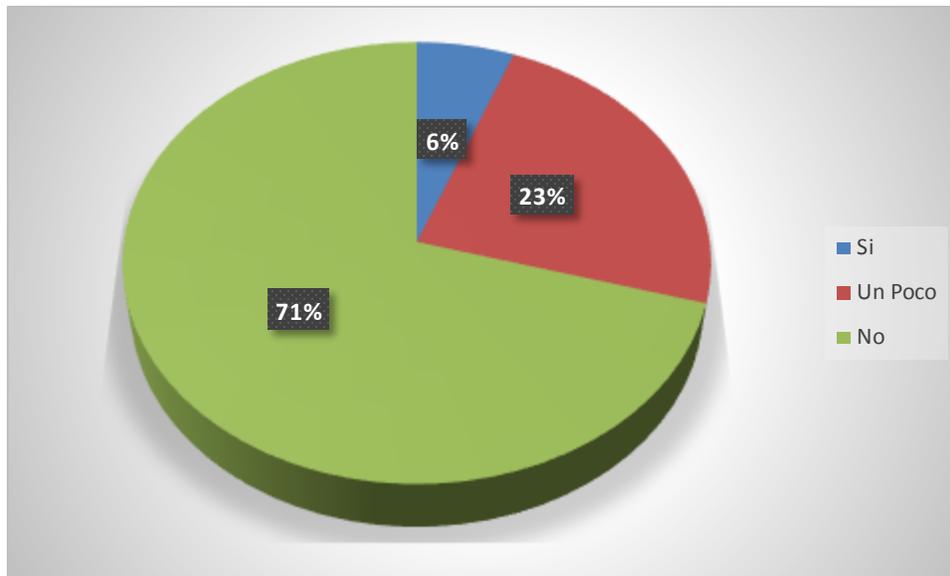
11. Conoce de la aplicación de Arduino para fabricar PLC de hardware libre



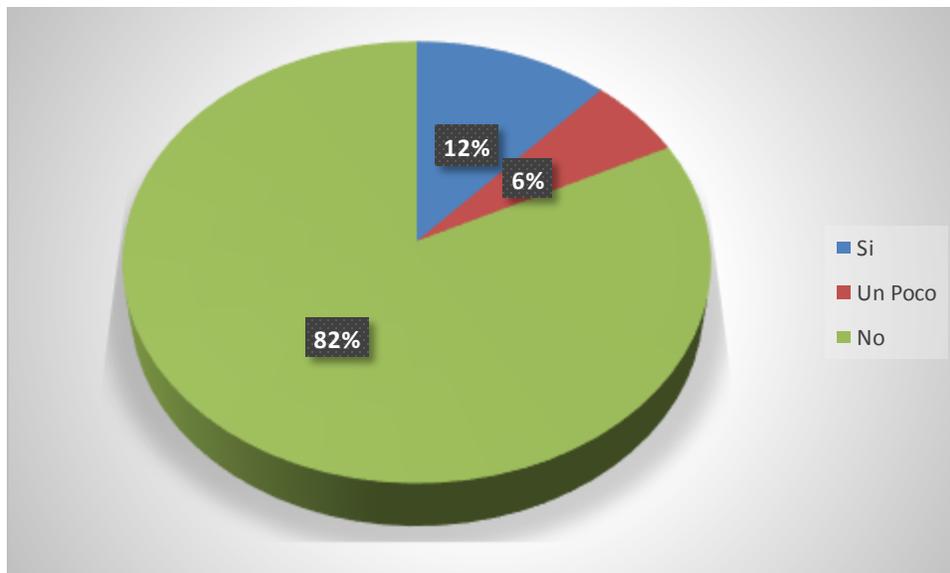
12. Conoce de la aplicación de Software Libre para programar un PLC



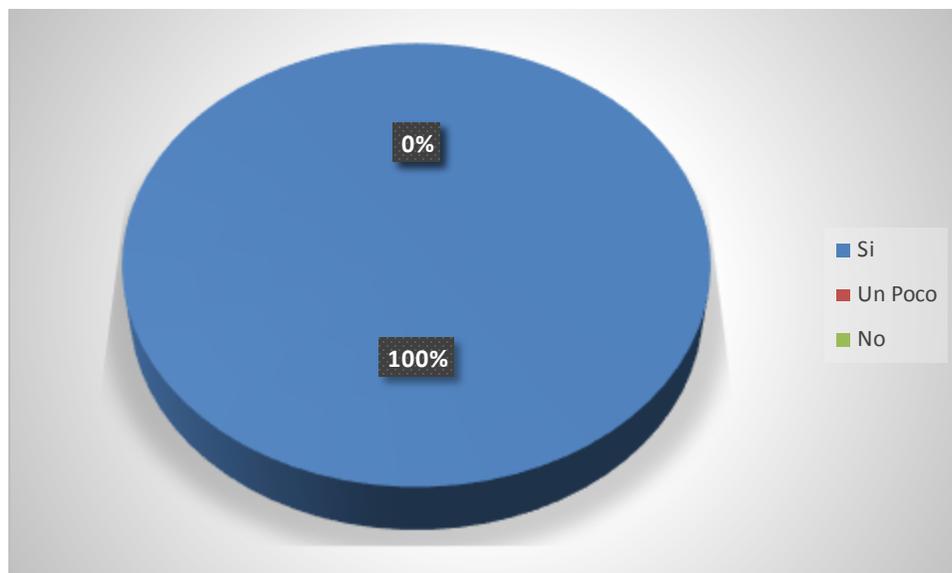
13. Conoce de la aplicación de Software libre para poder crear programas con lenguaje KOP para programación de PLC



14. Conoce de proyectos de aplicación de Software y Hardware Libre para fabricación de PLC en el país



15. Considera necesario el impulsar el desarrollo de investigaciones en ITCA FEPADE sobre este tema



RESULTADOS DE APLICAR LA METODOLOGÍA CIENTÍFICA APLICADA Y METODOLOGÍA ACCIÓN

Por medio de la implementación de estas metodologías se comprobó que todo el conocimiento teórico que se recopiló durante toda la información se utilizó específicamente para poder reflejarlo en la práctica por medio de la fabricación del micro PLC y dando a conocer el diseño de los diferentes circuitos que lo componen, de esa manera realizar una demostración didáctica sobre el funcionamiento del micro PLC

DISEÑO Y CALCULOS DE CIRCUITOS DE MICRO PLC

Todos los diseños de los circuitos que componen el dispositivo, ha sido analizados de manera que cada uno realice su función eléctricamente "estable" tomando en consideración para su diseño los cálculos respectivos según datos y gráficos de las hojas técnicas de los fabricantes, adjuntos en anexo de tesis.

Sin embargo cabe resaltar que el diseño de dichos circuitos no ha sido sometido a evaluación o test de laboratorio con patrones electromagnéticos, que nos ayuden a determinar su completa compatibilidad e inmunidad a ruido en un entorno industrial. A pesar de estos detalles, en todos los diseños se han considerado en la medida de lo posible, contener todos los criterios y elementos necesarios para intentar alcanzar cierta adaptabilidad a un entorno industrial como cualquier otro microPLC.

Para finalizar cabe resaltar que a cada presentación de los circuitos, se hace aclaración de donde se tomó el diseño básico y que elementos se añadieron según criterio, para su funcionamiento. Sin embargo los cálculos de los circuitos en todos los casos son agregados para futuras evaluaciones en el diseño del circuito.

CIRCUITO REGULADOR DE VOLTAJE 24VDC A 12VDC

Descripción

En principio el dispositivo se alimentara con una fuente de voltaje de 24VDC.

Sin embargo para poder segregar circuitos a niveles de potencial correctos, se tomaron a consideración 3 niveles de voltaje:

- 24.0 VDC para alimentación General y Control de Salidas a Relé (Fuente de Voltaje).
- 9.35 VDC (aprox.) para Alimentación de Circuito Arduino (LM317).
- 5.0 VDC para Microcontrolador (Circuito Interno Arduino NANO).

La primera regulación viene dada por una "fuente de voltaje externa" AC/DC alimentada comúnmente a 110/240VAC y salida a 24VDC.

La segunda regulación viene diseñada con un regulador LM317 donde se calcularon las resistencias R1 y R2 para alcanzar los 9 VDC, los cuales alimentaran la tarjeta del Arduino NANO.

La tercera regulación es realizada en el circuito de la tarjeta del Arduino NANO para control de entradas y salidas del microcontrolador.

Diseño Esquemático

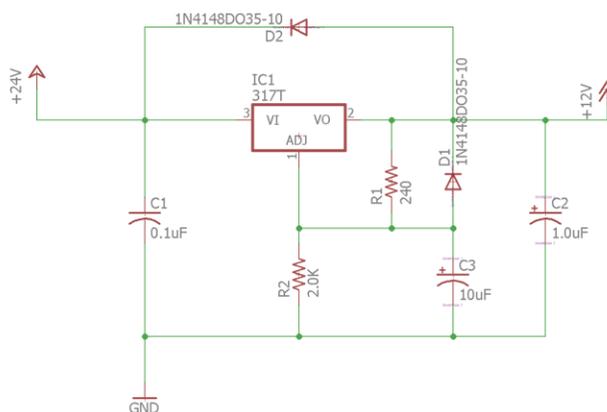


Imagen 4.1 Circuito LM317

- La R1 y la R2 son necesarias para determinar el voltaje de salida.
- C1 es utilizado dado que el regulador no está muy cerca de los condensadores del filtro de la fuente de alimentación.
- Según hoja técnica del LM317 los valores del condensador oscilan entre los 0.1uF y 1.0 uF cerámicos o de tantalio, proporcionan una característica de bypass como ajustadores de voltajes de entrada.
- El condensador C3 es utilizado para mejorar el rechazo de la ondulación, evitando la amplificación de la ondulación o rizados, dado que el voltaje de salida se ajustaría cada vez más alto.
- El condensador C2 mejora la respuesta transitoria de la salida.
- El D2 proporciona una trayectoria de descarga de baja impedancia para evitar que el condensador C3 se descargue en la salida del regulador.
- El D1 se recomienda si se utiliza el C2 proporciona una trayectoria de descarga de baja impedancia para evitar que el condensador se descargue en la salida del regulador.

CÁLCULOS DE DISEÑO

Según hoja técnica del LM317 el voltaje de salida del regulador es calculado con la siguiente ecuación:

$$V_o = V_{ref} \left(1 + \frac{R_2}{R_1} \right) + (I_{adj} * R_2)$$

Despejando R2 y considerando Iadj sumamente baja nos queda:

$$R_2 = \left(\frac{R_1}{V_{ref}} \right) (V_{out} - V_{ref})$$

A criterio sugerido por el fabricante el Vref= 1.25Vdc

Y si consideramos para un Vout=9.0Vdc, R1 fija de 240 Ohm, tenemos:

$$R2 = \left(\frac{240}{1.25}\right)(9 - 1.25) = (192)(7.75) = 1,488 \text{ Ohm} \cong \mathbf{1.5K \text{ Ohms}}$$

Entonces las resistencias de diseño serian:

R1=240 Ohms

R2=1.5K Ohms

ENTRADAS DIGITALES

Descripción

El circuito de entradas digitales, se componen en principio de tres sistemas:

- Sistema de aislamiento de potenciales (Aislamiento Galvanico)
- Circuito estabilizador antirebote.
- Compuerta buffer inversora (Schmitt Trigger)

En el Sistema por optoacoplador aislamos los dos potenciales, recibimos un voltaje de señal a 24VDC y controlamos dicha señal con un voltaje a 5VDC, en el extremo con el transistor del optoacoplador.

Diseño Esquemático

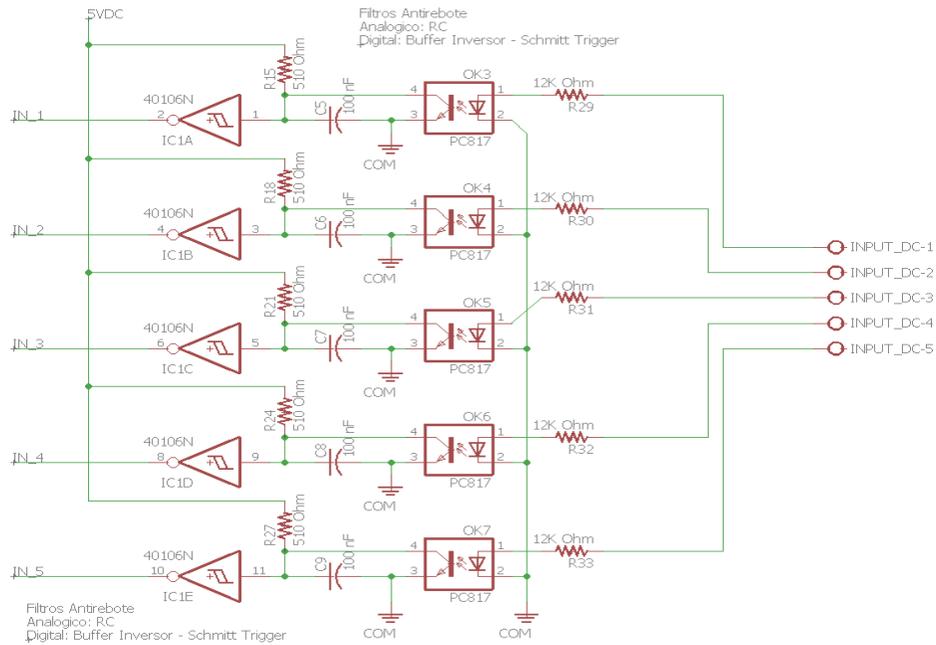


Imagen 4.2: Circuito de entradas digitales

Cálculos de Diseño

En el diseño utilizamos un optocoplador PC817, el cual utiliza un diodo infrarrojo para la saturación de la base del transistor, el cual su corriente Max en directa, según hoja técnica es $I_f=50\text{mA}$.

A 100% de su corriente máxima utilizamos una Resistencia dada por un voltaje de entrada de 24VDC.

$$R = \frac{24\text{VDC}}{50\text{mA}} = 480.0 \text{ Ohms}$$

Si trabajamos al 80% de la capacidad necesitamos 40mA, entonces:

$$R_{80\%} = \frac{24\text{VDC}}{40\text{mA}} = 600 \text{ Ohms} \cong 620.0 \text{ Ohms (Valor Comercial)}$$

Paras una $\text{CTR} = 100\% \text{ min} = 0.1\text{mA}$

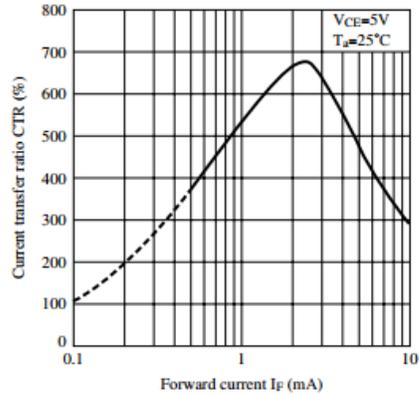


Figura 4.3 Ratio de Transferencia de Corriente vs. Corriente Directa

Seleccionamos el pico máximo de la curva para establecer el punto de máxima transferencia de potencia hacia la base del transistor.

Tomamos 2mA como carga al 680%, este valor nos servirá para establecer la resistencia del diodo, dado que dicha corriente, define la intensidad del as infrarrojo cuando la saturación del transistor por lo tanto a un voltaje nominal de 24vdc tenemos

$$R = 24\text{vdc} / 2\text{Ma} = 12\text{kohms}$$

$$\text{Disipando una potencia de } P = (24 - 1.2) (0.002)$$

$$P = 46\text{mW} - 60\%$$

Un voltaje Max permitido seria

$$R = \left(\frac{30\text{vdc} - 1.2}{2000} \right)^2$$

$$R = 0.0691 \quad R = 69.1\text{mW} \text{ equivalente a } 70\text{mW } 100\%$$

$$I = \left(\frac{30 - 1.2}{12\text{K}} \right)^2$$

$$I = 2.4\text{mA}$$

Se establece un 30% de Potencia como mínimo lo cual sería disipar 20mW en la entrada. El voltaje mínimo en la entrada sería:

$$V_{min} = \sqrt{P * R} = \sqrt{20\text{mW} * 12\text{Kohms}} = 15.5\text{vdc}$$

Es decir, tendríamos un rango de voltaje de 15.5 a 30vdc

El circuito finaliza con una salida de una compuerta tipo buffer inversora Schmitt Trigger la cual hace uso de una histéresis de voltaje que es la tendencia de conservar un nivel lógico hasta que no se produzca un cambio mismo del mismo. De esta manera se previene el ruido que podría tapar la señal original y que causaría cambios falsos de estado, si los niveles de referencia y entrada son parciales, con este sistema cubrimos las frecuencias más altas que no se logran filtrar con el circuito anti rebote.

Entradas Analógicas

Descripción

En el diseño de este circuito se consideró un lazo de corriente de (4-20mA) - (0-20mA) puesto que el lazo de corriente necesita ser convertido a voltaje, se considera utilizar entonces un seguidor de voltaje con una resistencia en paralelo a la entrada el cual nos permitirá condicionar la señal analógica con una alta impedancia de entrada, lo que significa que no nos consumiría gran parte de corriente de entrada.

Diseño Esquemático

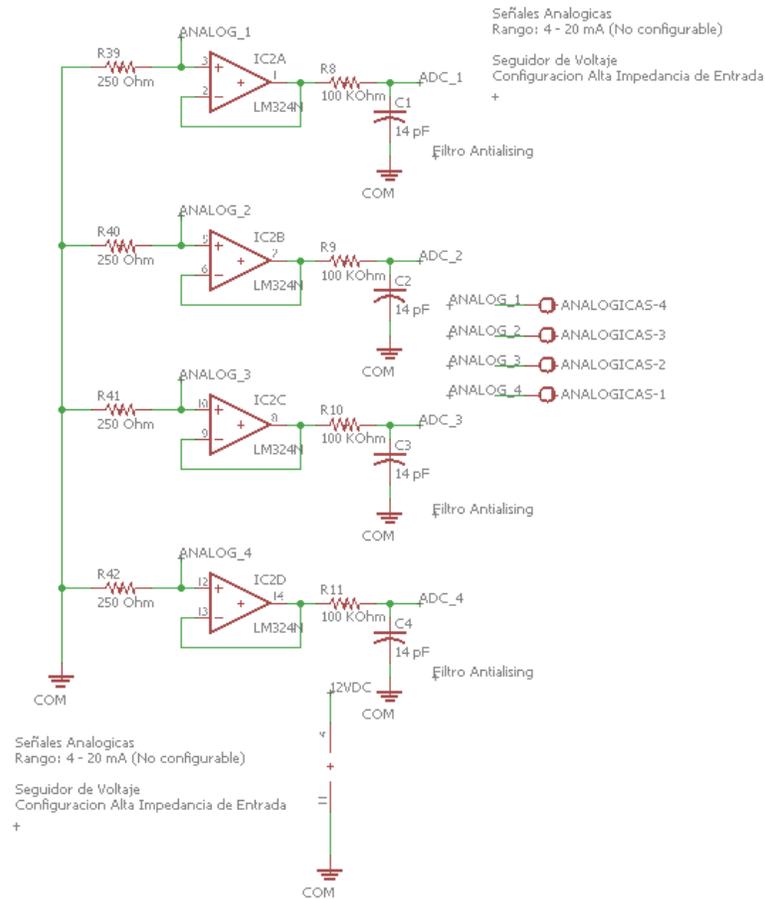


Imagen 4.4: circuito de entradas analógicas

Cálculos de Diseño

Con la resistencia en serie a la fuente de corriente convertimos a un diferencial de voltaje dicha corriente (Teorema de Norton) dado que el Arduino percibe lazos de voltaje de 0 a 5v calculamos la resistencia necesaria para la conversión a 20mA

$$R = \frac{5vdc}{20mA} = 250ohm$$

Con esta resistencia de conversión en 4mA tendríamos

$$V_{min} = 4mA * 250 \text{ ohm} = 1vdc$$

$$V_{max} = 20mA * 250 \text{ ohm} = 5vdc$$

Por lo tanto estamos en el rango correcto de 0 a 5 voltios admisibles en el Arduino

Muchas señales analógicas se convierten en portadoras de señales parasitas de alta frecuencia transmitidas normalmente por redes de alimentación AC o sistemas de radio frecuencia. Estas frecuencias normalmente se incorporan como componentes de lazo de corriente.

Se tomó consideración que para asegurarse que el contenido de la frecuencia de entrada está limitado, se añade un filtro pasa bajas en el diseño justo antes de la entrada analógica del micro controlador. su misión será atenuar las frecuencias altas evitando que el ruido sea muestreado

Para un lazo de corriente la frecuencia de corte del filtro pasa bajas viene sugerido por el fabricante del micro controlador en la hoja técnica de 113.68KHz según el teorema de Nysquist que establece que la frecuencia de corte será la mitad entonces

$$F_c = \frac{1}{2\pi * RC} = \frac{1}{2\pi(100K)(14pF)} = 56.84KH$$

Circuito Base Arduino NANO

Descripción

Para el diseño de salidas digitales se supone una señal de 5vdc provenientes del micro controlador tomando en cuenta que las salidas del arduino manejan un máximo de 40 mA max

Diseño Esquemático

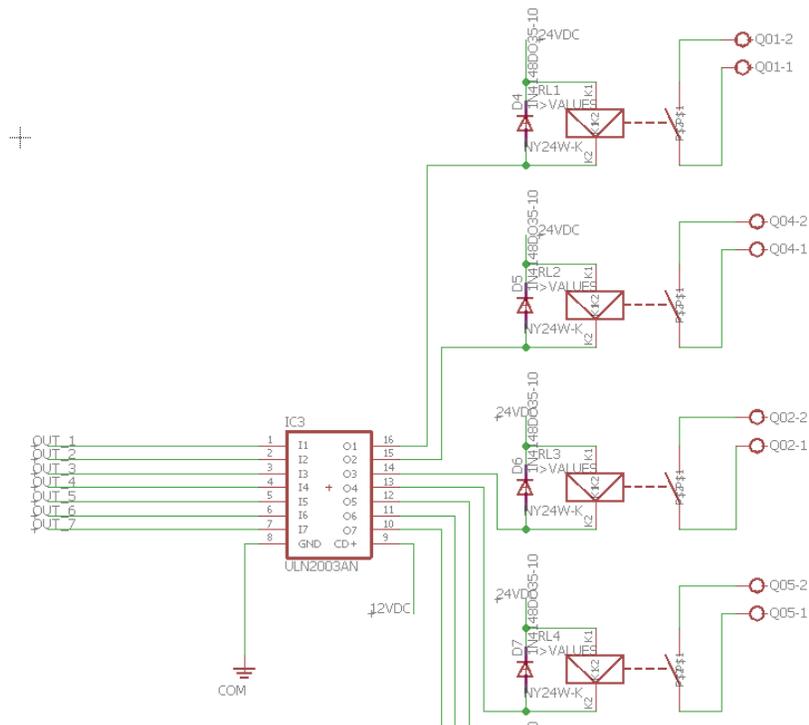


Imagen 4.5: Circuito base de arduino nano

Cálculos de Diseño

Si sabiendo que la impedancia de entrada es de 2.7Kohm entonces tenemos

$$I = \frac{5v}{2.7Kohm} = 1.85mA.$$

Como consumo máximo de las salidas digitales, estamos dentro del margen admisible de su capacidad.

PCB Regulator de Voltaje 24Vdc a 12Vdc

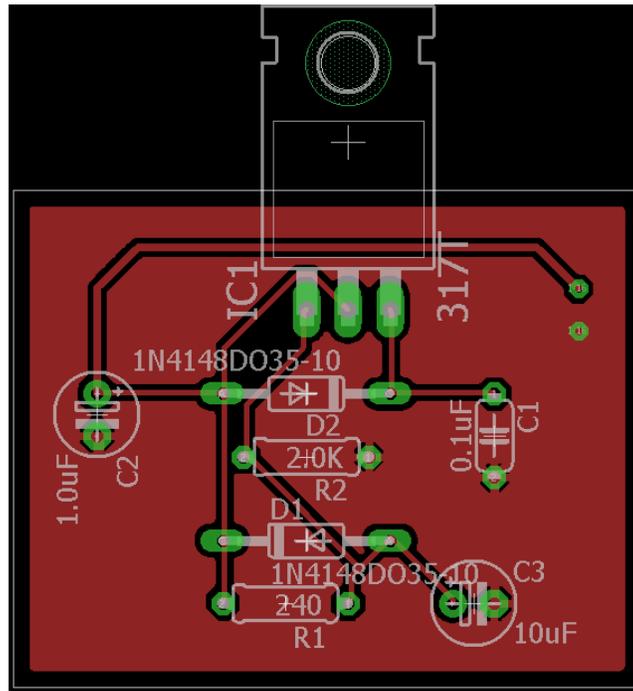


Imagen 4.6: Diseño de PCB's

Imagen 4.7 Esquema Circuito Base Arduino

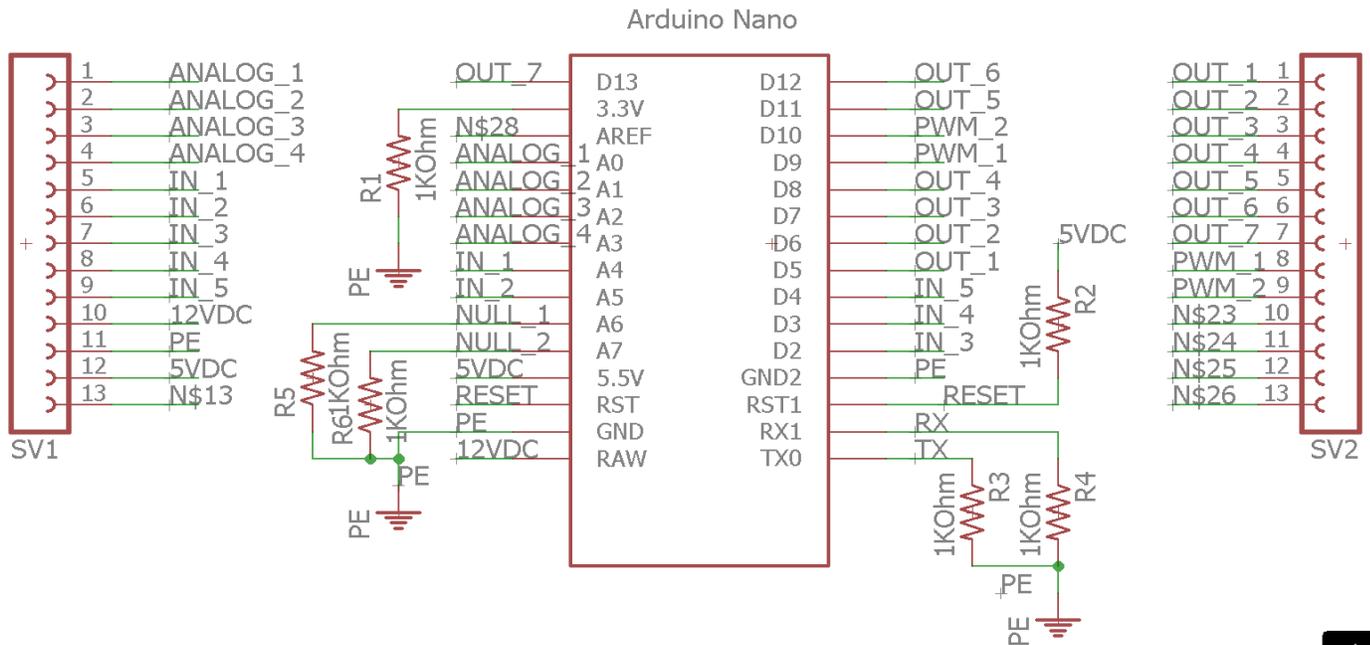


Imagen 4.8 TOP PCB Circuito Base Arduino

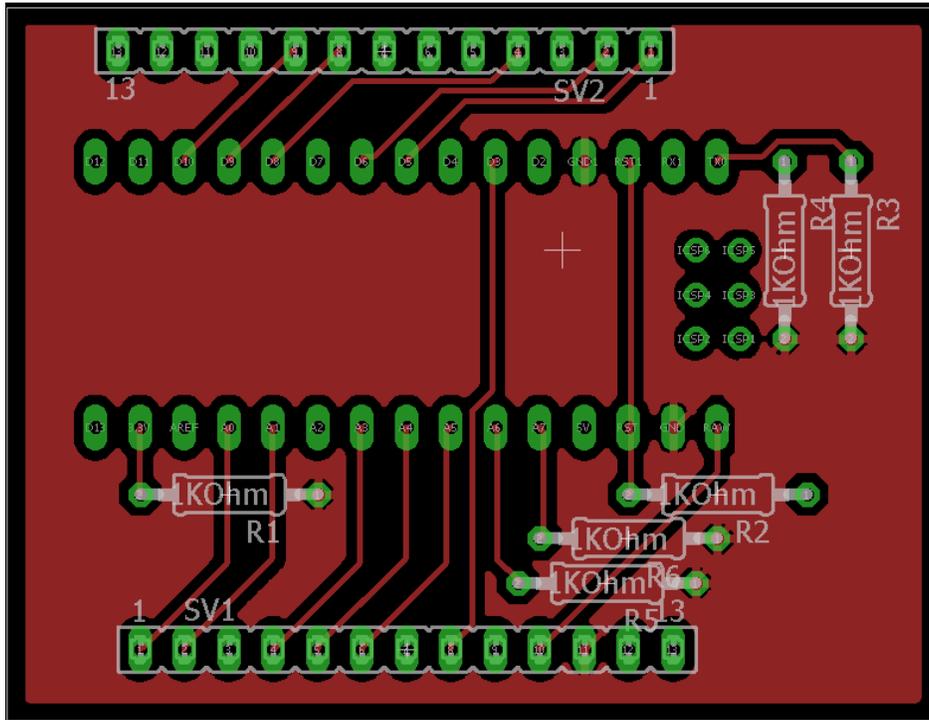


Imagen 4.9: Bottom PCB Circuito Base Arduino

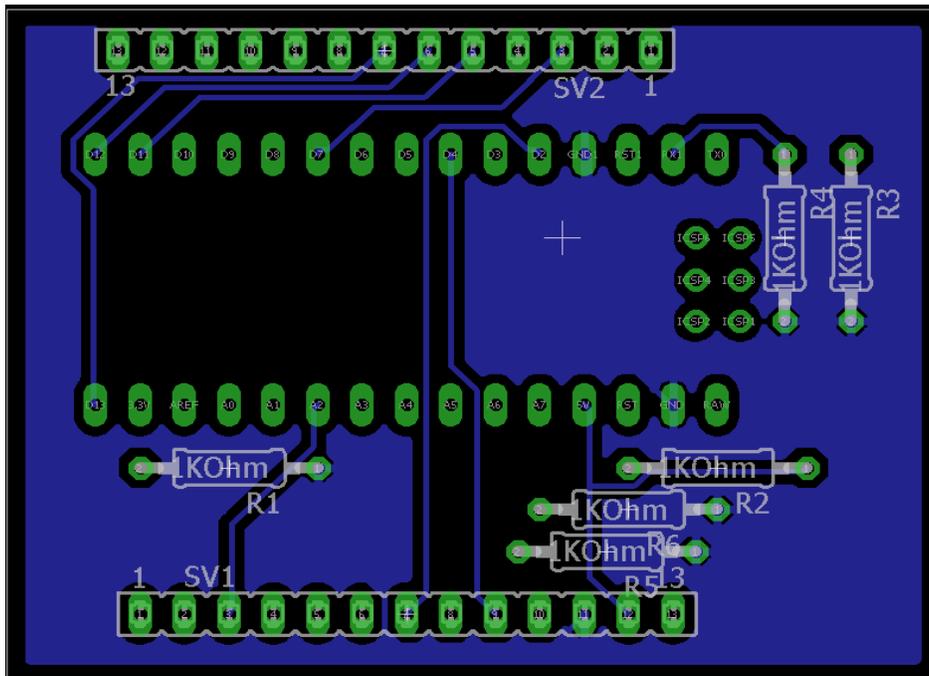


Imagen 4.10: PCB CIRCUITO INTERFAZ IN/OUT - TOP

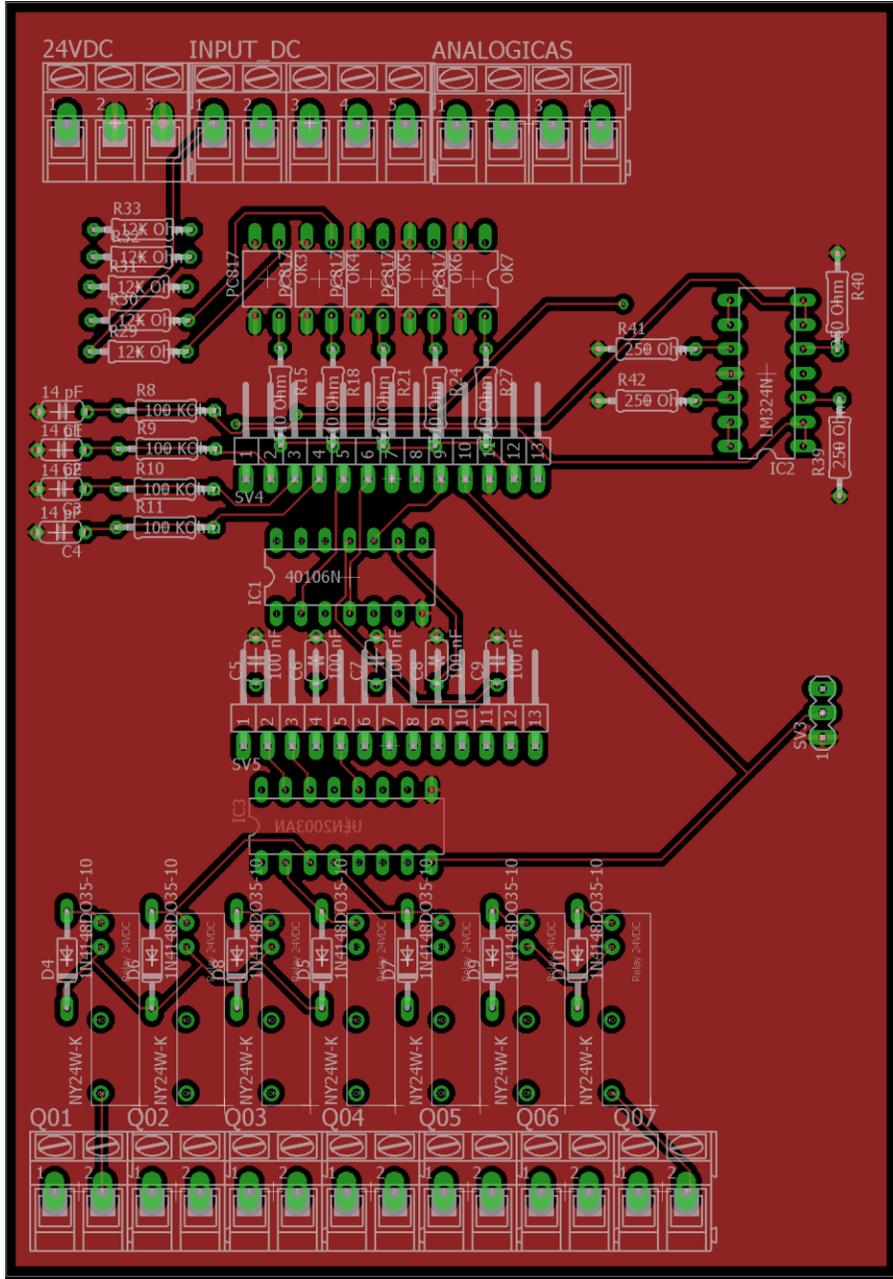
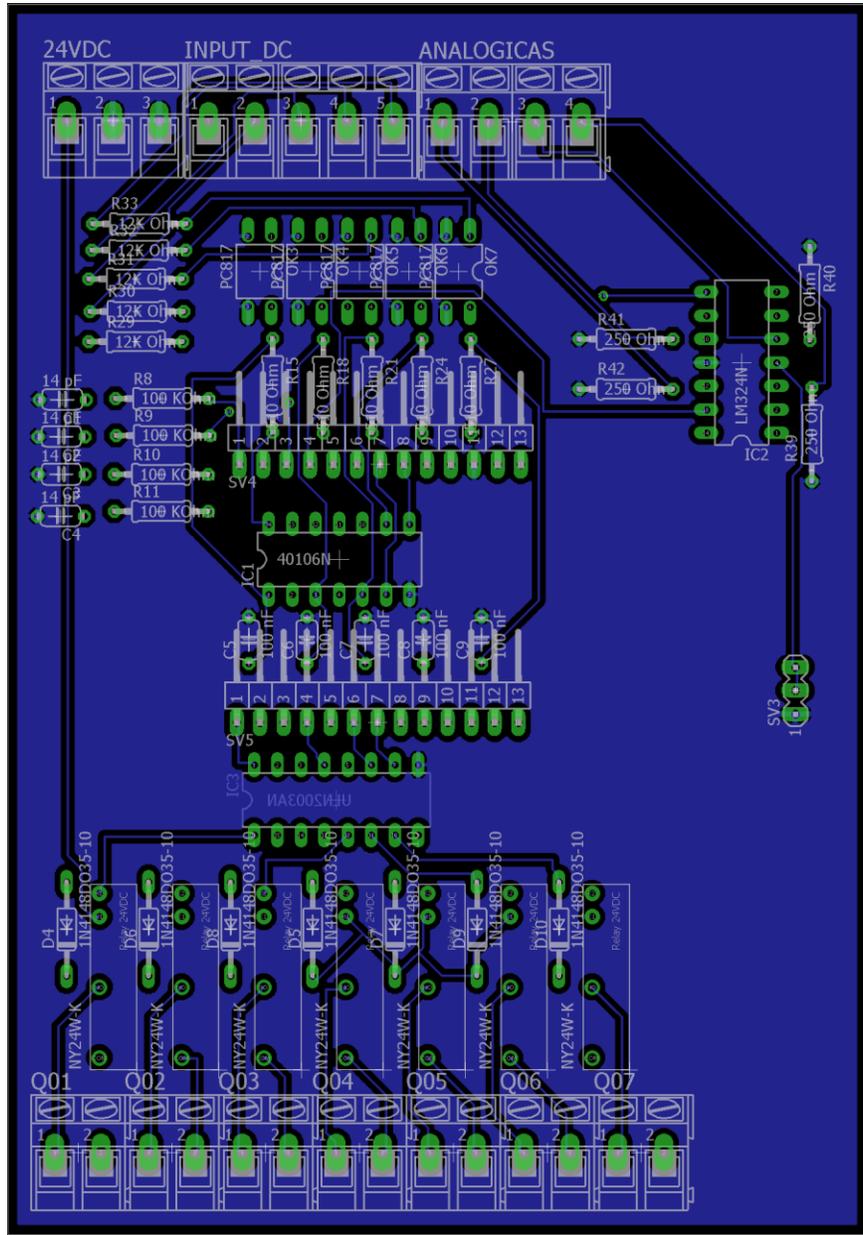


Imagen 4.11: PCB CIRCUITO INTERFAZ IN/OUT - BOTTOM



CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

En la realización de nuestro estudio de investigación se desarrollaron una serie de diferentes temas que están relacionados con nuestro proyecto, analizando detalladamente que aporte o ideas nos pueden proporcionar para fortalecer nuestro conocimiento teórico y cómo podemos demostrar estas teorías llevándolas a la práctica y resaltando los hallazgos más importantes e influyentes que se relacionan directamente con nuestro proyecto, lo cual nos lleva a plantear las siguientes afirmaciones con base en el análisis realizado durante todo el trabajo de investigación

Se analizó que en nuestro país no se ha desarrollado un a investigación específica sobre la implementación de autómatas programables basados en microcontroladores utilizando Hardware libre y software de código abierto

La interpretación del por qué los controladores lógicos surgieron se basa en una necesidad por parte de las empresas en querer automatizar sus procesos cada vez exigiendo un alto nivel de tecnología, eficiencia y adaptabilidad a diferentes cambios. Para poder cumplir con los estándares del cliente es necesario conocer cómo funciona internamente un PLC, cuáles son sus componentes, que capacidad de control posee y cuáles son sus características de funcionamiento y fabricación.

Se determinó que los microcontroladores y microprocesadores en la actualidad se están fabricando de tal manera que puedan tener la capacidad para desarrollar aplicaciones complejas siempre manteniendo su forma compacta pero con un gran potencial, accesible económicamente para las personas y brindando una interfaz didáctica con el usuario para facilitar su uso.

Las diversas plataformas de microcontroladores en placas de Hardware libre se han convertido en una herramienta de uso didáctico muy eficaces para transmitir el conocimiento científico a las personas implementado una comunicación común como lo es el puerto USB y presentando un tipo de programación que pueda ser desarrollada por diferentes personas sin tener un conocimiento avanzado en lógica de programación

Se especificaron una serie de normativas con las que debe de cumplir un diseño de PLC así como los requisitos eléctricos y mecánicos con los que debe de contar de tal manera que garantice al usuario un alto grado de funcionabilidad, alta eficiencia y detallar las características de fabricación y las condiciones óptimas en las que deben de operar correctamente estos controladores.

Se desarrolló un análisis basado en el uso de Softwares libre resaltando sus características principales las cuales permiten al usuario hacer modificaciones de programas pero sin aplicar alguna restricción para que otros no puedan cambiar o modificar dichos programas, aclarando ciertas dudas sobre el uso de estos softwares que por su nombre se piensa que todos son de forma gratuita pero en realidad su nombre se debe a las libertades y manipulación que tienen los usuarios sobre los programas y exponiendo las ventajas que se obtienen al utilizarlos.

BIBLIOGRAFIA

Cegarra Sánchez, J. (2004). Metodología de la Investigación Científica y Tecnológica. Madrid: Díaz de Santos.

Hernández Sampieri, R., Fernández Collado, C., & Pilar Baptista Lucio, M. (2014). Metodología de la investigación. México: McGraw-Hill.

Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P., García Espejo, M., & Limón Cano, S. (2010). Fundamentos de metodología de la investigación. Madrid [etc.]: McGraw-Hill.

Martín Alloza, J. (2000). Montaje de componentes y periféricos microinformáticos (UF0465). IC Editorial.

Morris Mano, M. (1998). Arquitectura de computadoras. México: Prentice Hall.

Hennessy, J., & Patterson, D. (2002). Arquitectura de computadores. Madrid: MacGraw-Hill.

Perspectiva - Procesadores digitales de señal (DSP) Arquitecturas y criterios de selección. Dpto. Ingeniería Electrónica. Universidad Politécnica de Cataluña. Jordi Salazar.

Guerrero, V., Martínez, L., & Yuste, R. (2014). Comunicaciones industriales. Barcelona: Marcombo.

Torrente Artero, O. (2013). Arduino. San Fernando de Henares, Madrid: RC Libros.

Álvarez Pulido, M. (2000). Controladores lógicos. Marcombo.

Banzi, M., Shiloh, M., & Gómez Celador, J. (2016). Introducción a Arduino. Madrid: Anaya Multimedia.

Banzi, M. (2012). Getting started with Arduino. Beijing: O'Reilly.

Enríquez Harper (1999) "fundamentos de control de motores eléctricos en la industria", Limusa 186 pág.

Martin Juan Carlos "El autómatas programable (automatismos industriales)" Editex S.A. Madrid 288 pág.

Cerda Luis Miguel (2014) "instalaciones eléctricas y automatismos" Paraninfo S.A. 374 pág.

Benchimol Daniel (2011) "Microcontroladores" Dalaga Buenos Aires 194 pág.

Escuela de negocios (2009) "la oportunidad del software libre, capacidades derechos e innovación" España 16 pág.

Salazar Pedro (2009) Robótica y automática en el aula: Trabajando con microcontroladores picaxe 80 pág.

SITIOGRAFÍA

- ✓ http://biblioteca.ucv.cl/site/servicios/documentos/como_escribir_tesis.pdf
- ✓ <http://www.proyecto-ciaa.com.ar>
- ✓ <http://digital.ni.com/public.nsf/allkb/7D2769CE61EC2F7786257D9A00560067>
- ✓ <https://www.gnu.org/philosophy/fsfs/rms-essays.pdf>
- ✓ <https://www.gnu.org>

GLOSARIO

Curricula: Es el plan de estudios o proyecto educativo general donde se detallan las materias que el alumno debe aprobar.

AWL: (Anweisungsliste) ó STL (Statement List) ó Lista de Instrucciones

SCL: (Structured Control Language) ó Lenguaje de Texto Estructurado

KOP : (Kontakts plan) ó LAD (Ladder Diagram) ó Esquema de contactos

FUP: (Funktions Plan) o FBD (Function Block Diagram) ó Diagrama de) ó lenguaje de conexión de bloques

DFD: Un diagrama de flujo de datos (DFD) es una representación gráfica del "flujo" de datos a través de un sistema de información

SPI: (Serial Peripheral Interface) es un síncrono de comunicación en serie especificación de interfaz usado para comunicación a corta distancia,

CEM: (Compatibilidad Electro Magnética): es la rama de la tecnología electrónica y de telecomunicaciones que estudia los mecanismos para eliminar, disminuir y prevenir los efectos de acoplamiento entre un equipo eléctrico o electrónico y su entorno electromagnético

IEC: (La Comisión Electrotécnica Internacional) organización de normalización en los campos: eléctrico, electrónico y tecnologías relacionadas.

IDE: (Entorno de desarrollo integrado) es una aplicación de software que proporciona servicios integrales para los programadores informáticos para el desarrollo de software

ANEXOS

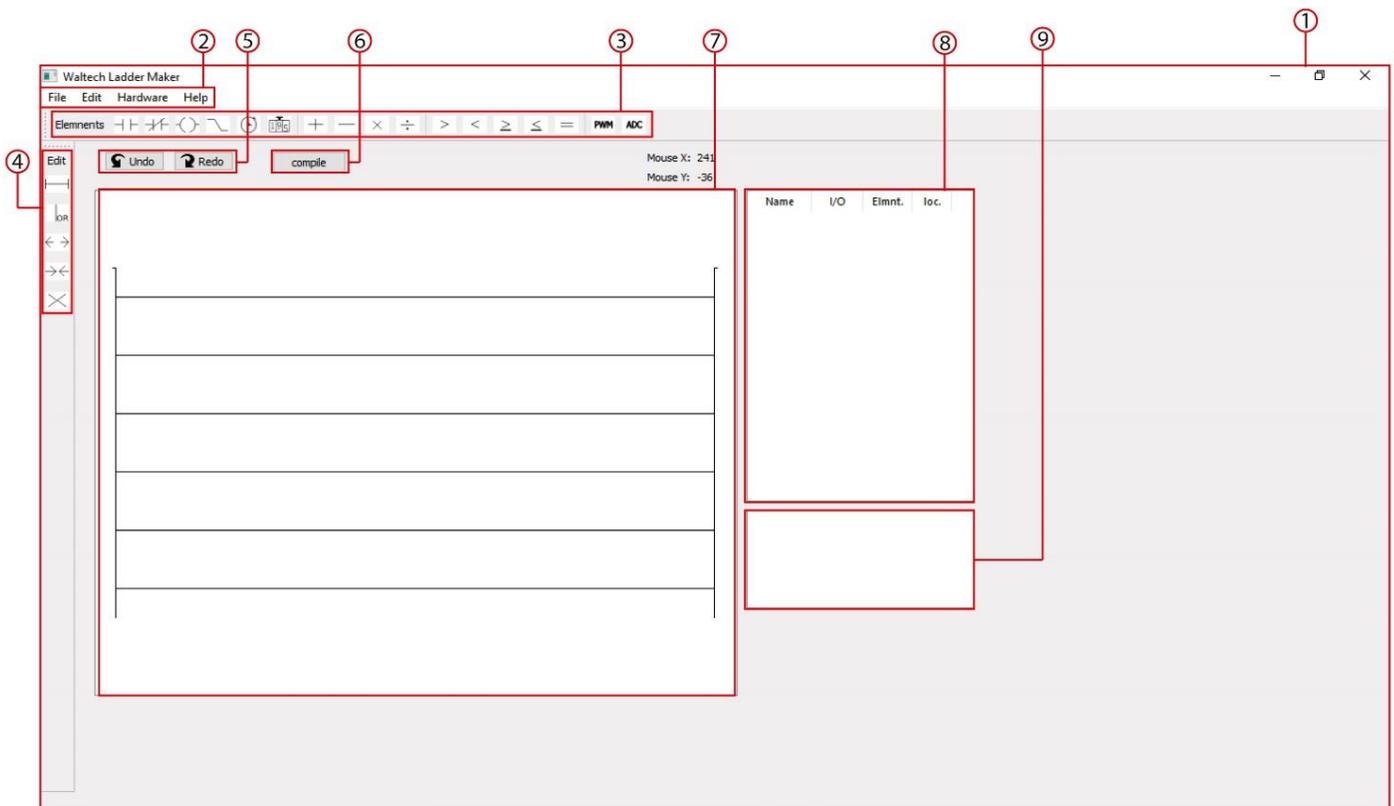
**Manual de Usuario -
Software LadderMaker
for Windows v.107a
(Released
11/08/2014)**

Introducción

Este manual contiene instrucciones detalladas y notas sobre el funcionamiento y uso de este manual. Por su beneficio, lea este manual con detenimiento antes de utilizar el software. Conserve este manual en un lugar al alcance para una rápida consulta.

Objetivo: Facilitar al usuario la familiarización e uso del sistema así como sus funciones y forma de trabajo.

Dentro del entorno del software podemos encontrar lo siguiente:



1. Ventana de programa
2. Barra de Menu
3. Barra de Elementos
4. Barra de Edición
5. Función Deshacer/Rehacer

6. Compilar y Descargar Archivo
7. Entorno de Trabajo
8. Tabla de Información de I/O
9. Información y Estado de Programa

Instrucciones Disponibles

Dado que es un lenguaje grafico de programación las instrucciones posibles vienen distribuidas en una barra de herramientas.

1. **Ventana del programa:** Entorno general de la interfaz donde el usuario puede realizar las rutinas del programa

2. **Barra de Menú:** Aquí se encuentran los principales menú del software:

- a) Archivo
- b) Edición
- c) Hardware (Tipo de Placa Arduino a utilizar)
- d) Ayuda

3. **Barra de Elementos**



Botón/Objeto Grafico	Descripción
	Contacto Abierto
	Contacto Cerrado
	Bobina
	Detector de Flanco Negativo
	Temporizadores BCD (UINT)

	Tipo: On Delay.
	Contadores (UINT) Tipo: Ascendente
	Operación SUMA (INT)
	Operación Resta (INT)
	Operación Multiplicación (INT)
	Operación División (INT)
	Comparación de Datos Mayor que (INT)
	Comparación de Datos Menor que (INT)
	Comparación de Datos Mayor o Igual que (INT)
	Comparación de Datos Menor o Igual que (INT)
	Comparación Igual que (INT)
	Salida PWM
	Entrada ADC (Configuración de Entradas Analógicas) Voltaje de Ref. Inferior: 0.0 V Voltaje de Ref. Superior: 5.0 V Resolución Digital: 10 bits

	Resolución Analógica: 4.883mV Nivel de cuantificación: 1024 Valores
--	--

4 Barra de Edición

Botón/Objeto Grafico	Descripción
	Insertar Segmento
	Insertar ramal paralelo en OR
	Aumentar espacio en segmento
	Reducir espacio extra en segmento
	Eliminar elemento

5 Función Deshacer/Rehacer

6 Compilar: Al tener completo el diagrama KOP a programar en el microplc, este se compila y se transmite al microcontrolador para que este pueda ejecutarlo

7 Entorno de Trabajo: Aquí es donde se realiza el diagrama del programa a ejecutar, en este se utilizan la barra de elementos y edición.

8 Tabla de Información de I/O: Aquí se muestra el detalle de las entradas y salidas que se están ejecutando en el programa

9 Información y Estado de Programa: Se muestra el estado en el que se encuentra el compilador, si el microcontrolador está

conectado, carga de programa, errores que puedan surgir durante el proceso de compilado y transmisión, etc

Datos Técnicos de micro PLC

Información general	
Tarjeta de Microcontrolador base	Arduino NANO
Ingeniería con	
- Programación	Walter Ladder Maker IDE Arduino
Lenguajes de Programación	
- KOP	SI, Waltech Ladder Maker
- Lista de Texto	SI, Atmel QTouch
- FUP	NO
- Texto Estructurado	SI, IDE Arduino
Tensión de Alimentación	
Valor Nominal (DC)	24.0 V
Rango Admisible, límite inferior (DC)	20.4 V
Rango admisible, límite superior (DC)	28.8 V
Intensidad de entrada	
Consumo Valor Nominal	19 mA (CPU)
Consumo Max.	500 mA
Memoria	
Memoria de trabajo	
- Integrado	2 KB
- Ampliable	No
Memoria de Carga	

- Integrado	32 KB, de los cuales 2KB reservado para Bootloader
- Ampliable	No
Respaldo	
- Existente	Si, Libre de Mantenimiento
- Sin Pila	SI
Tiempos de ejecución de la CPU	
Para operaciones a bits	6 uS
Para operaciones a palabras	260 uS
Para operaciones de coma flotante	N/A
CPU - Bloques	
N° de bloques (total)	1 OB Principal (Limitado Waltech Ladder Maker)
Bloque de Organización	
Cantidad Máxima	Limitada por la memoria de trabajo para código Memoria Flash 32KB.
Funciones Admisibles	Limitado por programa (Ver Manual)
Área de datos y su remanencia	
área de Datos Remanentes	No
área de Marcas y Datos Globales	16KB
Configuración de hardware	
N° de módulos por sistema máx.	Solo modulo CPU
Entradas digitales	
N° de Entradas Digitales	5 canales
Con funciones Tecnológicas	Ninguna

Canales Integrados	7 canales
Tensión de entrada	
- Valor Nominal (DC)	24VDC
- Para señal "0"	0.1VDC - 2.4 VDC
- Para señal "1"	2.4VDC - 5.5VDC
Retardo de entrada (a tensión nominal de entrada)	
- Parametrizable	No
- En transición "0" a "1", máx.	20 nS
- En transición "1" a "0", máx.	50 nS
Longitud de Cable	
Apantallado, máx.	500 m
No apantallado, máx.	300 m
Salidas Digitales	
Número de salidas	6 canales
Canales integrados	7 canales
Tipo de Salidas	Relé
Tensión máx.	28 VDC 120 VAC
Intensidad máx.	10 A
Entradas Analógicas	
N° de entradas analógicas	4 canales
Canales Integrados	4
Tipos de Entrada	
- Corriente	Si
- Voltaje	No
Rangos de entrada (valores nominales)	
- 4 - 20 mA	Si
- 0 - 4 mA	Si
Formación de valores analógicos	

Tiempo de integración y conversión/resolución por canal	
Resolución	8 bits, 1,024
Tiempo de conversión por canal	260 uS
Tipo de cableado	2 hilos, 3 hilos
Interfaz Programación PLC	
Tipo de Interfaz	Mini - B USB
Norma Física	USB 1.0
Funciones de Comunicación	
Comunicación en RS 232/485	No
Comunicación Ethernet	No
Servidor Web	No
N° de Conexiones	1 tipo USB
CEM (Compatibilidad Electromagnética)	
Tipo de Prueba e Inmunidad	Sin Pruebas Aplicadas
Condiciones Ambientales	
Temperatura de Servicio	
- Mínima	-10 °C
- Máxima	60 °C; Todas las salidas conectadas al mismo tiempo, en horizontal.
Humedad relativa	60%