

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



“Diseño y construcción de un (PLC) Control Lógico
Programable basado en la tecnología del
microcontrolador”

PRESENTADO POR:

RENATO BARRERA CUBIAS
JORGE ALBERTO CARTAGENA MÉNDEZ
CARLOS OSMIN POCASANGRE JIMÉNEZ

PARA OPTAR AL TÍTULO DE:
INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, FEBRERO DEL 2004

UNIVERSIDAD DE EL SALVADOR

RECTORA :
Dra. María Isabel Rodríguez

SECRETARIO GENERAL :
Licda. Lidia Margarita Muñoz Vela

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO :
Ing. Mario Roberto Nieto Lovo

SECRETARIO :
Ing. Oscar Eduardo Marroquín Hernández

ESCUELA DE INGENIERÍA ELÉCTRICA

DIRECTOR :
Ing. Luis Roberto Chévez Paz

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:
INGENIERO ELECTRICISTA

Título :
“Diseño y construcción de un (PLC) Control Lógico
Programable basado en la tecnología del
microcontrolador”

Presentado por :
Renato Barrera Cubias
Jorge Alberto Cartagena Méndez
Carlos Osmín Pocasangre Jiménez

Trabajo de Graduación aprobado por:

Docente Director:
Ing. Ricardo Ernesto Cortez

Docente Director:
Ing. Hugo Miguel Colato Rodríguez

San Salvador, febrero del 2004

Trabajo de Graduación Aprobado por:

Docentes Directores :

Ing. Ricardo Ernesto Cortez

Ing. Hugo Miguel Colato

DEDICATORIA

A DIOS

Dedico este Trabajo de Graduación en primera instancia a DIOS, EL TODOPODEROSO, sin quien; ninguna faena es realizable.

A MIS PAPAS

Como hijo, se lo dedico a mis papas: Jorge y María Luz, fuentes de todas mis virtudes y algunos defectos, de él heredé la fortaleza, perseverancia y la voluntad para terminar lo iniciado y otras, de ella, la paciencia, el amor por los demás, etc.

A MI FAMILIA

Como Esposo, se lo dedico a mi Esposa y mis Hijos: Mercedes Arely (mamá e hija) y a Renatito. Mi esposa, por permitirme graduarme de papá y a mis hijos por permitirme ejercer, llenándome aun más de fuerzas para enfrentar la vida, todos ellos mi orgullo y consuelo, mi alegría y aflicción, mi apoyo pleno.

A MIS HERMANOS

Como hermano, se lo dedico a: Jorge, Afrodita, Maria Elena, Mercedes, Víctor, Ernesto y Oscar Ulises, todos iguales y tan diferentes, separados pero muy unidos, mi auxilio terrenal fuera de mi hogar.

Sin olvidar a mi tía Isabel y mi prima Carmen, y a todos Ustedes: amigos, compañeros, vecinos.

GRACIAS
Sinceramente
Renato Barrera Cubias

DEDICATORIA

A DIOS TODO PODEROSO

Por ser mi salvador y mi guía, por darme sabiduría, salud, valor y fuerza para alcanzar mis metas. Por iluminarme mi pasos y por estar siempre conmigo. Gracias Señor Jesús por haberme ayudado a salir adelante que a pesar de todo, a ti sea la honra y la gloria, porque a ti lo debo todo.

A MIS PADRES: JORGE y ARGENTINA

Que siempre me han dado todo su apoyo, sacrificio y amor para hacer de mí un profesional.

A MI HERMANO: LUIS MANUEL

En memoria a mi hermano que me dio apoyo, que se preocupó porque fuera alguien en la vida y que siempre quiso ver concluida mi carrera. Gracias mi querido hermano por todo lo que me ayudaste y porque cada triunfo mío siempre significó para ti una alegría.

A MIS HERMANOS: ANA SOLEDAD y FELIX ANTONIO

Por haberme brindado su apoyo, comprensión y ánimo para realizar mis anhelos.

A MIS TÍOS: MIGUEO y ALBERTINA

Que siempre me brindaron su apoyo incondicional y me ayudaron cuando más lo necesitaba.

A MI QUERIDA ESPOSA: ESPERANZA

Por su comprensión, amor incondicional y por estar siempre a mi lado en los momentos más duros alentándome siempre a seguir adelante.

A TODA MI FAMILIA

Que siempre me ayudaron.

A TODAS AQUELLAS PERSONAS

Que de una u otra manera me ayudaron durante mi carrera.

JORGE ALBERTO CARTAGENA MÉNDEZ

DEDICATORIA

A dios Todopoderoso: Por concederme la fortaleza para alcanzar la meta propuesta y este triunfo.

A mis queridos padres: A mi mamá Elba Esperanza Pocasangre y a mi papá Osmín Pocasangre por su incondicional apoyo y paciencia.

A mi hermana: Con cariño por su apoyo y comprensión brindada para este trabajo.

A mis hijos: Con ternura y amor a Carlos Alejandro Pocasangre Flores y Hazel Dayana Pocasangre Flores por estar presentes y darme fuerzas para obtener este triunfo, a ellos se la dedico.

A mis queridos amigos y demás familia que me apoyaron en esta empresa,
GRACIAS.

Atentamente: **Carlos Osmín Pocasangre Jiménez**

AGRADECIMIENTOS

Agradecemos sinceramente DIOS y a todas las personas que nos ayudaron para que pudiésemos terminar el Trabajo de Graduación.

Agradecemos el apoyo incondicional de nuestras familias, la paciencia de nuestros asesores para con nosotros y la confianza que nos brindaron nuestros compañeros y amigos.

MENCIÓN HONORÍFICA para nuestros compañeros y amigos del Instituto Técnico Industrial, que sin merecerlo nos brindaron su apoyo, su paciencia y su confianza.

Agradecimientos especiales para las Especialidades de Electrónica y Electrotecnia por su apoyo incondicional a través del Prof. Jorge Romero y el Ing. Juan Antonio Ascencio.

Renato Barrera Cubias
Jorge Alberto Cartagena Méndez
Carlos Osmín Pocasangre Jiménez

ÍNDICE

I. INTRODUCCIÓN	3
II. OBJETIVOS	5
III. IDENTIFICACIÓN DEL PROBLEMA.....	7
IV. JUSTIFICACIÓN	8
V. ALCANCES Y LIMITACIONES.....	9
VI. ANTECEDENTES.....	11
VII. PROGRAMA DE ACTIVIDADES.....	13
1. CAPITULO I.....	16
1.1 DESCRIPCIÓN DEL DIAGRAMA ELÉCTRICO DEL PLC	17
1.1.1 CARACTERÍSTICAS DEL PIC16F877 APLICABLES AL PLC.....	18
1.1.2 CONVERTIDOR ANALÓGICO/DIGITAL (HARDWARE EXTERNO).....	18
1.1.3 I/O PARA SEÑALES TTL.....	20
1.1.4 ENTRADAS Y SALIDAS DIGITALES PARA FUNCIONES LÓGICAS.....	21
1.1.4.1 INTERFASE CMOS/TTL.....	21
1.1.4.2 INTERFASE TTL/CMOS.....	22
1.1.4.3 SALIDAS CON GRAN CAPACIDAD DE CORRIENTE.....	24
1.1.5 MANEJADOR PARA EL DISPLAY MATRICIAL	26
1.1.6 UNIDAD PROGRAMADORA.....	28
1.2 MÉTODO DE FABRICACIÓN DEL PROTOTIPO	30
1.3 PROGRAMACIÓN Y FUNCIONAMIENTO PARTICULAR DEL PROTOTIPO	36
1.3.1 DESCRIPCIÓN DEL LENGUAJE ENSAMBLADOR.....	36
1.3.1.1 ENSAMBLADOR MPASM	36
1.3.1.1.1 VISIÓN GENERAL DEL ENSAMBLADOR MPASM.....	36
1.3.1.1.2 LOS ARCHIVOS DE ENTRADA/SALIDA DEL ENSAMBLADOR.....	38
1.3.1.1.3 INSTALACIÓN DEL ENSAMBLADOR	40
1.3.1.2 ENSAMBLADOR BAJO LA PLATAFORMA LINUX (RED HAT 7.0)	41
1.3.2 DESCRIPCIÓN GENERAL DE LOS PROGRAMADORES	42
1.3.2.1 PROGRAMADOR BAJO LA PLATAFORMA MICROSOFT (IC-PROG 1.05A).....	42
1.3.2.2 PROGRAMADOR BAJO LA PLATAFORMA LINUX RED-HAT 7.0.....	45
1.3.3 DISEÑO DE MANEJADORES DE LOS COMPONENTES DEL PLC.....	47
1.3.3.1 MÓDULO CONTADOR	47
1.3.3.2 MÓDULO GENERACIÓN DE SEÑALES TEMPORIZADAS.....	49

1.3.3.2.1	MÓDULO TMR0.....	49
1.3.3.2.2	MÓDULO TMR1.....	50
1.3.3.2.3	USO DE LOS TEMPORIZADORES TMR0 Y TMR1	51
1.3.3.3	MÓDULO DE ENTRADAS ANALÓGICAS.....	53
1.3.3.4	MÓDULO DE ENTRADAS Y SALIDAS CMOS.....	59
1.3.3.5	PUERTO DE SALIDA DE POTENCIA (3AMP MÁXIMO)	60
1.3.3.6	FUNCIONES BÁSICAS: AND, OR, NOT, NAND, NOR, XOR	62
2.	CAPITULO II.....	65
2.1	DISEÑO DEL ALGORITMO PARA EL SOFTWARE DE CONFIGURACIÓN DEL PLC.....	66
2.1.1	FLUJOGRAMA GENERAL.....	66
2.1.2	FLUJOGRAMA DEL MÓDULO CONTADOR	67
2.1.3	FLUJOGRAMA DEL MÓDULO COMBINACIONAL	68
2.1.4	FLUJOGRAMA DEL MÓDULO TEMPORIZACIÓN.....	69
2.1.5	FLUJOGRAMA DEL MÓDULO CONVERTIDOR ANALÓGICO DIGITAL.....	70
2.1.6	FLUJOGRAMA DEL MÓDULO SECUENCIAL	71
2.2	IMPLEMENTACIÓN DEL SOFTWARE DE CONFIGURACIÓN DEL PLC.....	72
2.2.1	MANUAL DEL USUARIO DEL OCRPLC.....	72
2.2.1.1	LÓGICA SECUENCIAL.....	74
2.2.1.2	LÓGICA SECUENCIAL.....	75
2.2.1.3	LÓGICA SECUENCIAL CON RELOJ EXTERNO	78
2.2.1.4	MÓDULO CONTADOR	81
2.2.1.5	TEMPORIZADORES (ASTABLE / MONOESTABLE).....	82
2.2.1.6	INTERFAZ GRAFICA PARA EL MÓDULO ADC	84
2.2.2	AMBIENTE LINUX	86
2.2.2.1	MÓDULO CONTADOR UP – DOWN.....	86
2.2.2.2	MÓDULO TABLA DE VERDAD.....	87
2.2.2.3	MÓDULO TEMPORIZADOR.....	88
2.2.2.4	MÓDULO CONVERTIDOR ANALÓGICO DIGITAL (ADC)	89
2.2.2.5	MÓDULO DE GENERACIÓN DE SEÑALES SECUÉNCIALES.....	90
2.2.2.6	DIFERENTES MENSAJES O CUADROS DE DIALOGO EN EL PROGRAMA.....	91
2.3	EJEMPLOS DE IMPLEMENTACIÓN DEL PLC	92
2.3.1	ARRANQUE DE UN MOTOR TRIFÁSICO	92
2.3.2	ARRANQUE Y PARO SECUENCIAL DE CUATRO MOTORES TRIFÁSICOS	98
3.	CONCLUSIONES.....	105
4.	BIBLIOGRAFÍA Y FUENTES DE CONSULTA.....	108
5.	ANEXOS.....	110

I. INTRODUCCIÓN

El presente documento se desarrolla en diferente etapas, que para la comprensión del lector se denominan capítulos, La primera etapa (ver anexo 5), inicia con un estudio de factibilidad que presenta, desarrolla y analiza la idea de producir en el país PLCs (Programming Logic Control) de bajo costo, fácil de programar, de reproducir y de utilizar, dicho estudio de factibilidad comprende un estudio de mercado, estudio técnico y propuesta financiera, todos con la finalidad de trazar la ruta para la obtención de los resultados deseados: propuesta, aceptación y permanencia de un producto en el mercado, logrando así uno de los objetivos primordiales, demostrar a lo largo de las distintas etapas en este trabajo , y a través de diversos estudios, que el proyecto realmente es factible, y que será aceptado y adoptado por la Industria de automatización local, localizada en la mediana y microempresa, gracias a su capacidad de satisfacer sus necesidades debido a sus características propias de trabajo.

En la segunda etapa (capítulo I), se hace un estudio al microprocesador PIC16F877, que servirá de base para la construcción del PLC, además de ello se analizarán las características más sobresalientes que tienen relación con el mismo, tales como temporizaciones, captura de datos de entrada, manejo de datos de salida, conversión A/D y las operaciones booleanas que se pueden implementar en aplicaciones de control. Así mismo se explica como funciona el diagrama a bloques, con las siguientes características:

- a) Funciones básicas: AND, OR, NOT, NAND, NOR, XOR;
- b) Retardo de activación y desactivación;
- c) Relé de Temporización;
- d) Contador progresivo/regresivo;
- e) Entradas analógicas;
- f) Entradas y Salidas que cumplan la normativa CMOS¹;
- g) Salida de potencia de hasta 3Amp máximo.
- h) Técnica de programación “ICSP²”.

¹ CMOS: Semiconductor de Óxido Metálico Complementario.

² ICSP: programación del circuito en forma serial

Se estudia la aplicación del software utilitario (MPASM y GPASM) necesario para ensamblar los códigos fuentes (drivers del PLC) y posteriormente se revisa la programación del microcontrolador³ en los ambientes de trabajo LINUX y WINDOWS.

Se describe el uso de un bromografo y el programa EAGLE (Easily Applicable Graphical Layout Editor) en una técnica que facilita la fabricación de circuitos impresos dándole una apariencia profesional muy por encima del sistema artesanal tradicionalmente ocupado (Plumón, Cinta adhesiva, etc.)

Para finalizar, en la tercera etapa (capítulo II) se presenta el software utilizado para programar el **PLC** con diferentes funciones que realizará, dicho software se ha diseñado para trabajar en los sistemas operativos Microsoft Windows y Linux, quedando a opción del usuario elegir el de su mayor conveniencia.

El software de programación ha sido diseñado de una forma modular, de manera que se pueden configurar diferentes aplicaciones según sea el caso. Cada módulo, haciendo uso de un interfaz gráfica, presenta una ventana diferente en donde se puede seleccionar los parámetros de entrada y/o de salida, además de configurar el hardware con el que se interactúa, utilizando esta información se genera un archivo fuente con extensión .ASM con los parámetros seleccionados que posteriormente es compilado y transferido al **PLC**.

También, se presenta un manual de usuario, que explica didácticamente los procesos necesarios para llevar a cabo la programación del PLC.

³ No debe confundirse con la programación del PLC

II. OBJETIVOS

OBJETIVO GENERALES:

- ❑ Contribuir como Escuela de Ingeniería Eléctrica de la UES con el desarrollo de la industria de automatización de la industria local, determinando la viabilidad y factibilidad de construir un PLC de fácil programación, reproducción y bajo costo.
- ❑ Diseñar e implementar el Prototipo de un PLC (Control Lógico Programable) de fácil programación y reproducción de bajo costo para contribuir con el desarrollo de la industria de automatización en El Salvador.
- ❑ Diseñar y crear un software el cual permita programar funciones de operación para el PLC en los ambientes de trabajo Microsoft Windows 9X/2000/Me/XP y Linux Red Hat 7.0.

OBJETIVOS ESPECÍFICOS:

- ❑ Conocer el nivel de demanda que puede tener el producto para determinar el tamaño del mercado.
- ❑ Determinar la tecnología a usar para la construcción del prototipo en mención.
- ❑ Determinar el sistema operativo de mayor demanda en la Industria de automatización local, focalizada en la mediana y microempresa.
- ❑ Determinar las materias primas óptimas del producto, sus características y el posible abastecedor para la fase de producción en serie.
- ❑ Determinar los requerimientos necesarios de maquinaria, equipo y materia prima necesaria para el proceso de producción en serie.
- ❑ Describir la arquitectura y funcionamiento general del microprocesador PIC16F877 para ocuparlo en el diseño del PLC.

- ❑ Diseñar el diagrama a bloques del PLC que cumpla con las características eléctricas especiales.
- ❑ Construir un circuito eléctrico (Hardware) del prototipo del PLC.
- ❑ Desarrollar los drivers que controlan las diferentes características internas del PIC para incorporarlos como módulos completos en el PLC.
- ❑ Diseñar un software el cual permita programar funciones de operación para el **PLC** en los ambientes de trabajo Microsoft Windows 9X/2000/Me/XP y Linux Red Hat 7.0.
- ❑ Diseñar la interfaz gráfica con el lenguaje de programación Microsoft Visual Studio 6.0 que corre en el sistema operativo WINDOWS
- ❑ Utilizar las herramientas gráficas GTK+ 1.2V en el Ambiente LINUX que nos permita crear el Software de programación del PLC.
- ❑ Comparar las ventajas y desventajas que existen al programar el PLC con lenguaje ensamblador para microcontrolador y con la interfaz Gráfica.
- ❑ Comparar las ventajas y desventajas de utilizar el Software de programación para el PLC usando ambos sistemas operativos.
- ❑ Diseñar una Interfaz gráfica modular, que permita programar el PLC con diferentes aplicaciones tales como: convertidor ADC, Lógica Combinacional, Lógica Secuencial, contadores ascendente y/o descendente, temporizadores, etc.
- ❑ Crear un conjunto de manejadores (drivers), que controlan las diferentes etapas del hardware del PLC (puertos periféricos, etc.)

III. IDENTIFICACIÓN DEL PROBLEMA

Las personas sean ya jurídicas o naturales, buscan dos cosas: estar más cómodos y hacer uso más eficiente del tiempo, todo esto con un bajo costo. A la satisfacción de éstas necesidades responden muchos de los productos en el mercado, de los cuales se enfatizarán los dirigidos a las Empresas Productoras de los PLC

Para el caso, el problema de importar y aún el mismo costo de los PLC (que se usan en la Industria de Automatización local en la mediana y microempresa) hacen que encarezcan los procesos de producción, por lo cual no llega a todos los niveles de la industria, por ende la opción de no utilizarlo hacen que el tiempo se use ineficientemente en las líneas de producción y se tenga menos competitividad de estas empresas en los mercados locales y costos elevados de sus productos finales.

Por lo que se describe anteriormente conlleva a proponer a la industria productiva nacional un PLC de fácil programación, producción y bajo costo, que se pueda programar desde una computadora que opere con cualquiera de los dos sistemas operativos: Windows y Linux.

IV. JUSTIFICACIÓN

La utilización del PLC en el control industrial, optimiza el uso del tiempo disminuye el costo de producción y maximiza las ganancias, es por tal motivo, que dicha práctica se vuelve cada vez más necesaria sobre todo en la industria moderna; que se enfrenta a la globalización de los procesos , a la estandarización de los mismos y a la producción a bajo costo, sin embargo, la implementación del mismo requiere grandes inversiones de capital, por lo que las micro y medianas empresas que deseen instalar y actualizar su sistema de control, se encuentran en clara desventaja por ello, se propone la construcción local de PLCs de bajo costo destacando la ventaja de fácil configuración y utilización lo que conlleva un beneficio para dichas empresas y contribuye al desarrollo de la industria local.

V. ALCANCES Y LIMITACIONES

ALCANCES

- El estudio inicial de factibilidad del trabajo de graduación describe las siguientes etapas: Estudio de Mercado, Estudio Técnico y Estudio Financiero; por lo que con estas herramientas se tiene la capacidad de reconcluir la rentabilidad, producción y demanda del proyecto.
- El voltaje de operación del PLC será 110/220 VAC, a fin de utilizar una fuente para PC (Personal Computer) que los soporta.
- El PLC puede controlar indirectamente cargas que sobrepasan en gran porcentaje las capacidades intrínsecas del mismo, auxiliándose de Relés de Estado Sólido en conjunto con contactores.
- Se presenta el software de configuración mediante una interfaz gráfica para Windows y Linux que se utilizará para programar el PLC.
- El PLC realiza funciones combinacionales y secuenciales, manejando cargas de cualquier tipo ya sea inductivas o resistivas.
- Se programara en ambos ambientes: Windows y Linux, así como también el funcionamiento del PLC con dichos programas en aplicaciones específicas.
- Cuando se trabaja en modo Secuencial, si se puede interactuar entre los módulos que maneja el PLC, no así en el resto de módulos que trabajan independientemente.

LIMITACIONES

- La falta de conocimiento por parte de las empresas de la existencia y funcionamiento de este tipo de tecnologías, fue una barrera a la hora de realizar los estudios de mercado, ya que dentro del mismo se pretenderá realizar encuestas relacionadas con productos similares(PLC), por lo que las personas entrevistadas generalmente son personas especializadas empíricamente.
- Para la determinación de la muestra en el estudio de mercado, se limita a la zona industrial del departamento de San Salvador.
- Las tecnologías aplicadas se basarán en los microcontroladores de marca MICROCHIP y/o los CPLD's de la marca XILINX, por la facilidad de importación.
- El prototipo de PLC solo tendrá funciones básicas que pueden hacerlo parecer inferior a fabricantes con mayor posicionamiento en el mercado, por ejemplo: SIEMENS, HITACHI, ABB, OMRON, etc.
- El circuito electrónico de programación del Microcontrolador PIC limita a usar solamente el software IC-PROG, que es una aplicación de WINDOWS, y PIC-PROG 2.3C que corre en LINUX.
- El valor del umbral de transición para las aplicaciones de los ADCs es fijo, significa que no se puede modificar desde el software y está ajustado en 2.5 voltios con una histéresis de +/- 200 mV
- Cuando se trabaja modularmente, cada módulo es específico para una determinada tarea, (Contador, Temporizador, Combinacional, ADC), no permitiendo la interacción entre ellos.
- La distancia máxima a la que se pueden colocar los sensores que proveen señal a las entradas analógicas del PLC dentro del rango nominal⁴ que éste requiere es de 15 metros para ambientes sin ruidos y de 10 metros en ambiente ruidosos⁵, usando cable PAR – TRENZADO AWG 22.

⁴ Definimos **rango de voltaje nominal** a aquellos valores de voltaje que están dentro de cero hasta cinco voltios

⁵ Ambiente ruidoso se refiere a un espacio físico en el cual la conmutación eléctrica (generadores de chispa) es continua, además de la interferencia electromagnética presente.

VI. ANTECEDENTES

¿QUE ES AUTOMATISMO?⁶

Se define como automatismo, al desarrollo de un proceso o funcionamiento de un mecanismo por si solo.

Los PLC'S se introdujeron por primera vez en la industria en 1960 aproximadamente. La razón principal fue la necesidad de eliminar el gran costo que se producía al reemplazar el complejo sistema de control basado en relés y contactores.

El problema de los relés era que cuando los requerimientos de producción cambiaban también lo hacía el sistema de control. Esto comenzó a resultar bastante caro cuanto los cambios fueron frecuentes.

Los nuevos controladores deberían ser fácilmente programables por ingenieros de planta o personal de mantenimiento. El tiempo de vida debería de ser largo y los cambios en el programa tenían que realizarse de forma sencilla.

Bedford Associates propuso algo denominado Controlador Digital Modular (MODICON: Modular Digital Controller) a una gran fabricante de autos. Otras compañías propusieron esquemas basados en ordenador, uno de los cuales estaba basado en el PDP-8. El MODICON 084 fue el PLC del mundo en ser producido comercialmente.

A mediados de los años 70's las tecnologías dominantes de los PLC'S eran máquinas de estados secuenciales y CPU basados en desplazamiento de bit. Los AMD 2901 y 2903 fueron muy populares en el MODICON y los PLC's A-B. Por cada modelo de microprocesador había un modelo de PLC basado en el mismo. No obstante el 2903 fue el más utilizado.

Las habilidades de comunicación comenzaron a aparecer aproximadamente en 1973. El primer sistema fue el bus Modicon(Modbus). El PLC podía ahora dialogar con otros PLC's y en conjunto podían estar aislados de las máquinas que controlaban. También podían enviar y recibir señales analógicas, entrando con esto al mundo real.

⁶ Real Academia Española(RAE)

Desafortunadamente a falta de un estándar acompañado de un constante cambio ha hecho que la comunicación entre los PLC's sea un desorden de sistemas físicos y protocolos incompatibles entre si.

En los 80's se produjo un intento de estandarización de las comunicaciones con el protocolo MAP(Manufacturing Automation Protocolo) de la General motor's, también se paso a la programación simbólica a través de ordenadores personales en lugar de los clásicos terminales de programación.

Los primeros PLC's en El Salvador se comenzaron a utilizar en los años 80's en algunas de las empresas tales como: MOLSA, AVX, TEXAS INSTRUMENT, EMBOTELLADORA SALVADOREÑA⁷, etc., para realizar funciones de control de procesos de producción, calidad, etc..

Los 90's han mostrado una gradual reducción en el número de nuevos protocolos y modernización de las capas físicas. El único estándar internacional (IEC 1131-3)⁸ intenta unificar el sistema de programación de todos los PLCs.

Ahora se disponen de PLCs que pueden ser programados mediante diagramas de bloques, lista de instrucciones, C y texto estructurado al mismo tiempo.

⁷ Fuente: Ing. Nelson Rauda gerente de mantenimiento MOLSA
Ing. Angel Ernesto Orellana Gerente de SERVIELECTRIC

⁸ Estándar de la IEEE

VII. PROGRAMA DE ACTIVIDADES

PROGRAMA GENERAL DE ACTIVIDADES

Detalla las actividades y el tiempo programado en cada etapa para la evolución del proyecto.

CRONOGRAMA DE ACTIVIDADES

Tiempo ⁹	Marzo				Abril				Mayo				Junio				Julio				Agosto				Septiembre				Octubre				Noviembre				Diciembre			
Actividades	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Estudio de factibilidad							X																																	
Diseño y construcción de un prototipo																																								
Diseño y funcionamiento del software																	X																							
Demostración practica del uso del PLC																																								
Fecha probables de defensa.									▲												▲																			
Nota: X semanas de asueto.																																								

Tabla 1 Programa general de actividades

Son fechas probables para las defensas:

- ❑ Viernes 9 de Mayo del 2003 primera defensa
- ❑ Viernes 25 de julio del 2003 segunda defensa
- ❑ Viernes 21 de noviembre del 2003 tercera defensa
- ❑ Viernes 12 de diciembre del 2003 defensa publica

⁹ Unidad de tiempo: semana (Cada mes tiene 4 semanas)

PROGRAMA DE ACTIVIDADES DEL ESTUDIO DE FACTIBILIDAD

Tiempo	Marzo			Abril			
	10-14	17-21	24-28	31Mar - 4	7-11	14-18 ¹⁰	21-24
Actividades							
PERFIL O ANTEPROYECTO						XXXXXXXX	
ESTUDIO DE MERCADO						XXXXXXXX	
Estudio de mercado de consumo						XXXXXXXX	
Estudio de mercado competidor						XXXXXXXX	
Estudio de mercado de abastecimientos						XXXXXXXX	
ESTUDIO TÉCNICO						XXXXXXXX	
Investigar características de la competencia						XXXXXXXX	
Investigar tecnologías que se usarán						XXXXXXXX	
Determinar características de control y programación						XXXXXXXX	
ESTUDIO ECONÓMICO						XXXXXXXX	
Cotizar y evaluar Precios						XXXXXXXX	
Determinar gastos indirectos de fabricación						XXXXXXXX	

Tabla 2: Programa de actividades del estudio de factibilidad

¹⁰ NOTA: X semana de asueto.

PROGRAMA DE ACTIVIDADES PARA EL DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO

Tiempo	Mayo				Junio				Julio	
	1	2	3	4	1	2	3	4	1	2
Actividades										
Diseño del diagrama a bloques del prototipo del PLC										
Descripción eléctrica de cada bloque										
Diseño del diagrama eléctrico del prototipo del PLC										
Elaboración del prototipo										
Programación y funcionamiento particular del prototipo										
Fecha probable segunda defensa										

Tabla 3 Programa de actividades para el diseño y construcción de un prototipo

PROGRAMA DE ACTIVIDADES PARA EL DESARROLLO DEL SOFTWARE DEL PLC

Tiempo	Julio		Agosto				Septiembre				Octubre	
	3	4	1	2	3	4	1	2	3	4	1	2
Actividades												
Descripción del algoritmo de programación												
Diseño de la estructura del flujo grama			xxxxx									
Selección del lenguaje de programación			xxxxx									
Diseño del programa manejador del PLC												
Aplicación del software de programación												
Depuración del programa												
Tercera defensa												

Tabla 4 Programa de actividades para el desarrollo de la programación del PLC

PROGRAMA DE ACTIVIDADES DE LA DEMOSTRACIÓN DEL USO DEL PLC

Tiempo	Octubre		Noviembre				Diciembre			
	3	4	1	2	3	4	1	2	3	4
Actividades										
Selección de las aplicaciones a realizar										
Preparación de la demostración practica del PLC										
Demostración de la aplicación del PLC										

Tabla 5 Programa de Actividades de la demostración del Uso del PLC

1. CAPITULO I

“DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE PLC (CONTROL LÓGICO PROGRAMABLE) BASADO EN LA TECNOLOGÍA DEL MICROCONTROLADOR PIC16F877 DE LA FABRICA MICROCHIP PARA EL CONTROL DIRECTO DE CARGAS DC CON UN MÁXIMO DE TRES AMPERES”

1.1 DESCRIPCIÓN DEL DIAGRAMA ELÉCTRICO DEL PLC

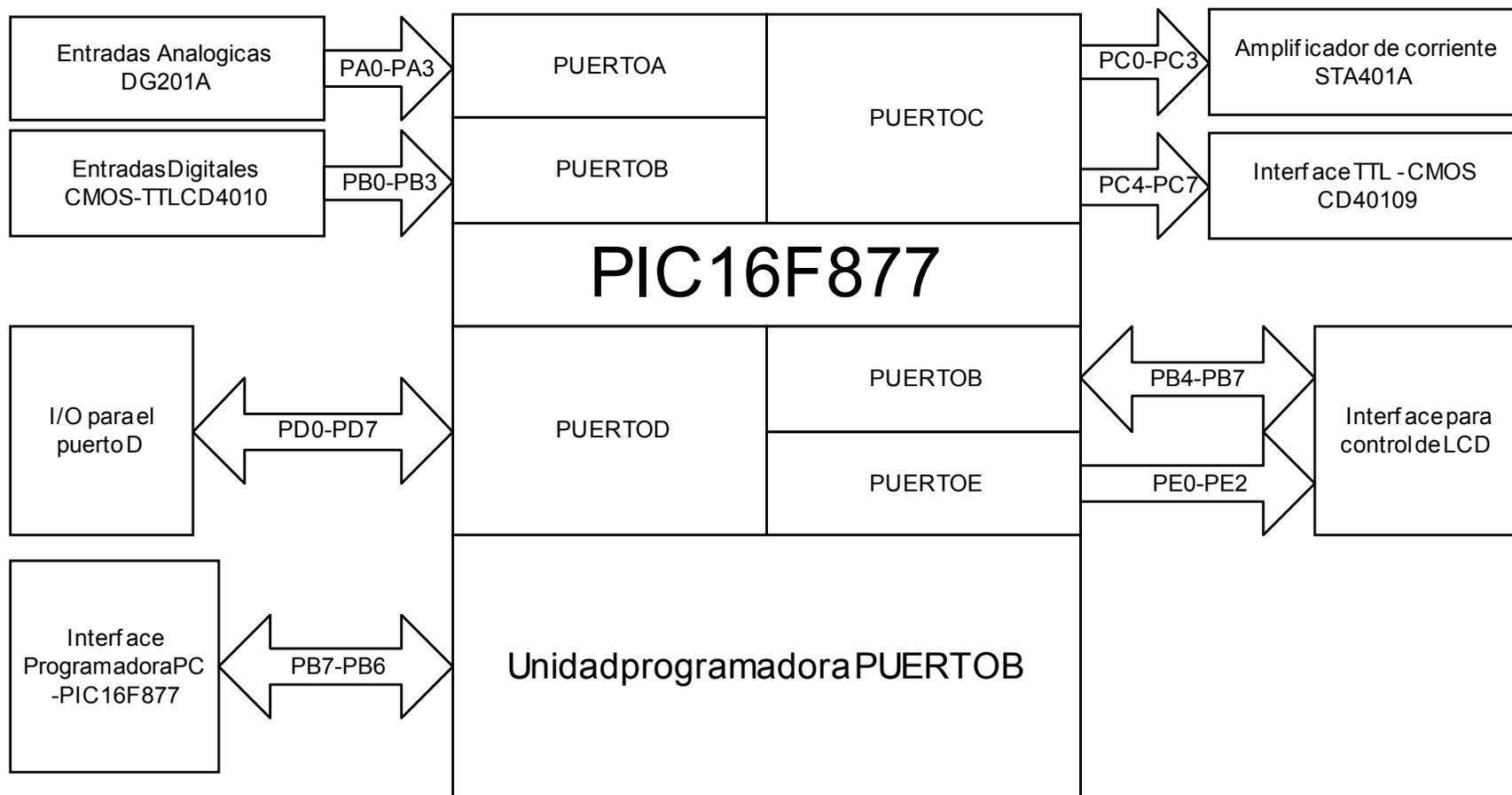


Figura 1.1 Diagrama a Bloques del PLC

1.1.1 CARACTERÍSTICAS DEL PIC16F877 APLICABLES AL PLC.

Como se describe en el Anexo A.3, el PIC16F877, posee las características idóneas para utilizarlo en el diseño de un PLC, éstas son:

- a) Tiene incorporado un módulo A/D, de 8 canales, los que pueden ser multiplexados; éste tiene una resolución de 10 bits.
- b) Tiene más de 31 líneas I/O, que son compatibles con la norma de niveles lógicos para tecnología TTL/CMOS a 5 voltios, que son suficiente para el PLC prototipo, sin tener que multiplexar ninguna, a excepción de RB7 y RB8, que se ocupan para el programador y la LCD. Se le puede programar “in situ”, es decir “In System Programming”, programado en el sitio de trabajo, además consume poca potencia(menos de 25 mA típico a 5 voltios y 4MHz).
- c) Su sistema de temporización es controlado por un cristal, que lo vuelve estable y preciso.

1.1.2 CONVERTIDOR ANALÓGICO/DIGITAL (HARDWARE EXTERNO)

Para implementar ésta función en el PLC, se aprovecha el convertidor A/D que tiene incorporado el PIC 16F877, éste genera una señal digital usando la técnica de Conversión de aproximación sucesiva, usando una frecuencia de muestreo de 10.416 KHz¹¹.

Este tiene 8 entradas analógicas asignadas al puerto A y al puerto E que multiplexadas pueden conectarse a la entrada analógica del ADC, teniendo como voltaje de referencia positivo un voltaje interno igual a VDD y como voltaje de referencia negativo un voltaje igual a VSS; pero por la arquitectura propia del PLC, no se puede habilitar las ocho entradas, sino solamente 5; ya que se ha destinado los pines RA4 (Generación de BEEP) y el puerto E como control de LCD, por lo que la configuración del registro SFR ADCON1 es 5/0 para PCFG3:PCFG0; ver Tabla 1.1.

Con la idea de proteger las entradas analógica de cualquier sobrevoltaje o sobrecorriente, se ha dispuesto intercalar un “switch” analógico, de tal manera que la

¹¹ Ver sección 1.3.3.3

corriente máxima de entrada a las patas del ADC (RA0,RA1, y RA3) sea hasta 20 mA, este “switch” analógico es el DG201ACJ, tiene una alimentación de ± 12 voltios.

ER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

bit 7 **ADFM:** A/D Result Format Select bit
 1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
 0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.
 bit 6-4 **Unimplemented:** Read as '0'
 bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs ⁽²⁾
0000	A	A	A	A	A	A	A	A	VDD	Vss	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	Vss	7/1
0010	D	D	D	A	A	A	A	A	VDD	Vss	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	Vss	4/1
0100	D	D	D	D	A	D	A	A	VDD	Vss	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	Vss	2/1
011X	D	D	D	D	D	D	D	D	VDD	Vss	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	Vss	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	Vss	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	Vss	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input D = Digital I/O

Tabla 1.1 Configuración de entradas del PIC (PUERTO A Y E) en el registro ADCON1.

DIAGRAMA ELÉCTRICO DEL ADC.

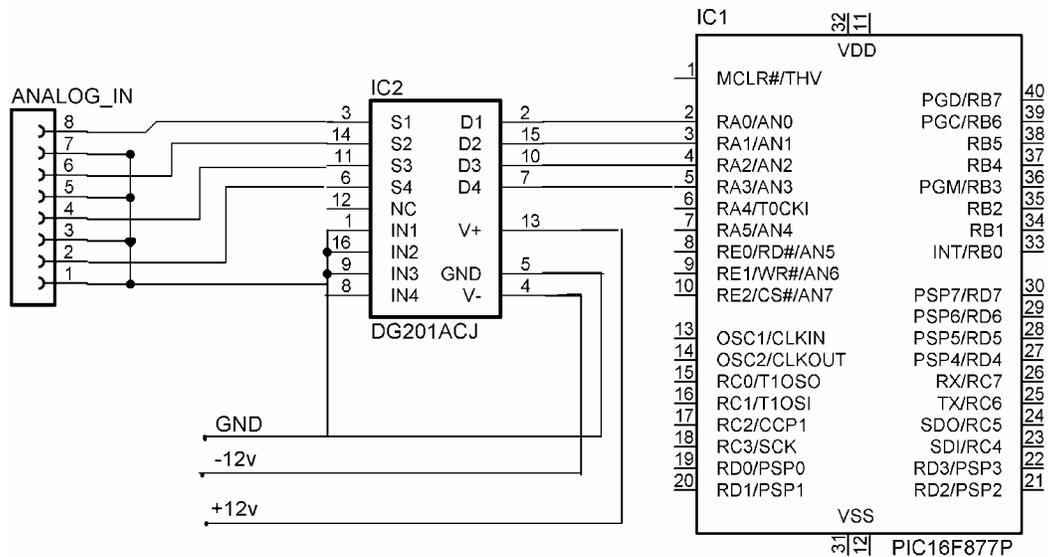


Figura 1.2 Conexión eléctrica de las entradas analógicas y el PIC.

1.1.4 ENTRADAS Y SALIDAS DIGITALES PARA FUNCIONES LÓGICAS.

1.1.4.1 INTERFASE CMOS/TTL

Para la realización del hardware, se usa en esta función (entradas digitales) la parte baja del puerto B (RB0,RB1,RB2,RB3), como entrada, pero para evitar el inconveniente de que las entradas del PLC deben cumplir con la norma CMOS a 12 voltios, y las entradas del puerto B del PIC cumplen con la norma CMOS pero de bajo voltaje (+5 V), se acondicionó la señal de entrada con un convertidor de nivel de voltaje, como el CD4010, que como se muestra en la figura 1.4, la función principal es acoplar lógica CMOS de alto voltaje a bajo voltaje

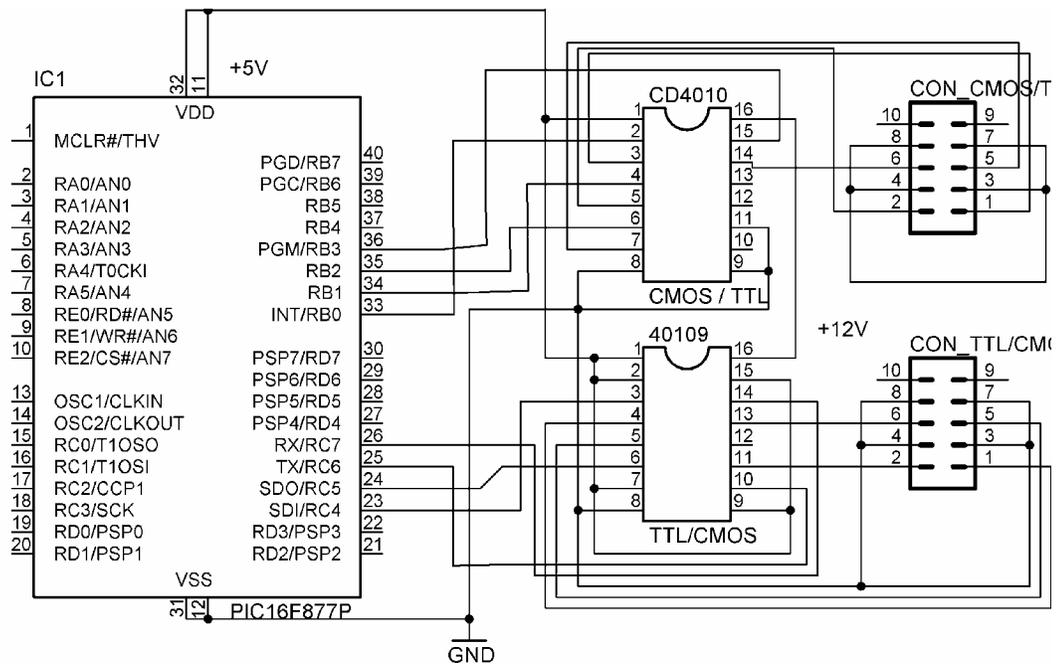


Figura 1.4 Entradas y salidas para implementar funciones digitales en el PLC

Descripción de patas del conector CMOS/TTL, es como se muestra en la Tabla 1.2:

Número de Pin	Nombre	Descripción
1	IN0	Entrada CMOS 1
2	IN1	Entrada CMOS 2
3	GND	Referencia
4	GND	Referencia
5	IN2	Entrada CMOS 3
6	IN3	Entrada CMOS 4
7	GND	Referencia
8	GND	Referencia
9	NC	No conectado
10	NC	No conectado

Tabla 1.2 Descripción del conector de entrada CMOS/TTL

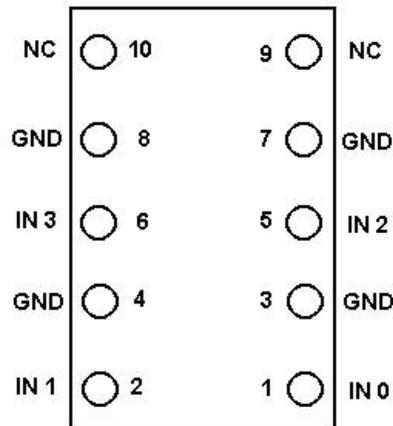


Figura 1.5, Conector de entrada CMOS/TTL

1.1.4.2 INTERFASE TTL/CMOS.

Una de las características eléctricas del PLC, es que tenga cuatro salidas que cumplan la norma de señal para tecnología CMOS, pero con alimentación de 12 voltios, el PIC, soporta la norma para señales CMOS, pero con alimentación de 5 voltios, que a su vez es compatible con TTL. Para evitar esta incompatibilidad, se usará un circuito integrado que servirá de interfase entre la salida CMOS del PIC con alimentación de 5 voltios, y la carga que necesita CMOS con alimentación a 12 voltios, tal integrado es el CD 40109B, que es un integrado con cuatro amplificadores que cambian el nivel de voltaje bajo en su

entrada (TTL) a alto voltaje en su salida (CMOS). La figura 1.6 muestra la descripción gráfica del tal integrado.

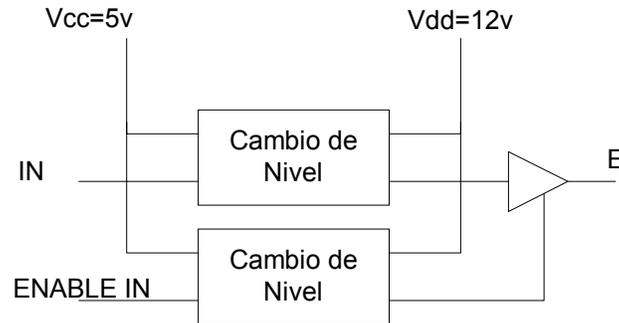


Figura 1.6 Diagrama a bloques (funcional) del IC CD 40109B

Descripción de patas del conector TTL/CMOS, es como se muestra en la Tabla 1.3:

Número de Pin	Nombre	Descripción
1	OUTL1	Salida CMOS 1
2	OUTL3	Salida CMOS 3
3	GND	Referencia
4	GND	Referencia
5	OUTL2	Salida CMOS 2
6	OUTL4	Salida CMOS 4
7	GND	Referencia
8	GND	Referencia
9	NC	No conectado
10	NC	No conectado

Tabla 1.3 Descripción del conector de entrada TTL/CMOS

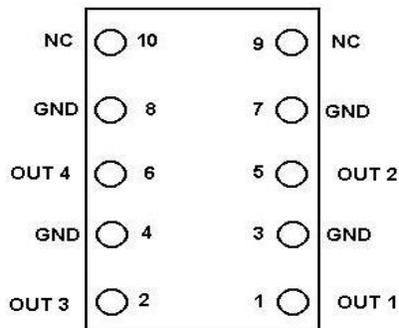


Figura 1.7, conector de salida TTL/CMOS.

La conexión eléctrica del IC STA401A con carga es como se presenta en la figura 1.10, los niveles de corriente mostrados son en modo de saturación, asumiendo una carga de 4Ω . El diodo volante (damper diode) solo se ocupa para absorber la FEM¹² inducida, cuando está conmutando carga de tipo reactiva, la cual pudiera dañar al transistor de salida, con carga de tipo resistiva no tiene efecto alguno.

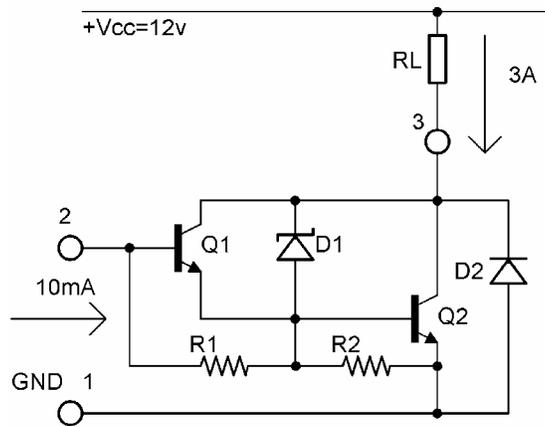


Figura 1.10, Diagrama interno del STA401A, muestra las corrientes de saturación.

BETA DE SATURACIÓN:

El β de saturación es igual a 300, (3000 mA/ 10 mA), al hacer un análisis de malla en el circuito de entrada, se obtiene como resultado la Ecuación 1.1, que resulta para la RB, que se coloca en el pin 2 de la figura 1.10, se expresa así:

$$Rb_{\min} = \frac{(Vin - [V_{BE}^{Q1} + V_{BE}^{Q2}])}{Ib_{\max}}$$

Ecuación 1.1 Resistencia de Base

Donde:

$$Rb_{\min} = \frac{(5.0 - [0.7 + 0.7])}{10 \times 10^{-3}} = 360 \Omega$$

¹² FEM: Fuerza Electromotriz Inducida por cargas Reactivas (Inductivas o capacitivas)

1.1.5 MANEJADOR PARA EL DISPLAY MATRICIAL

La Matriz LCD, se conectara como se muestra la figura 1.11:

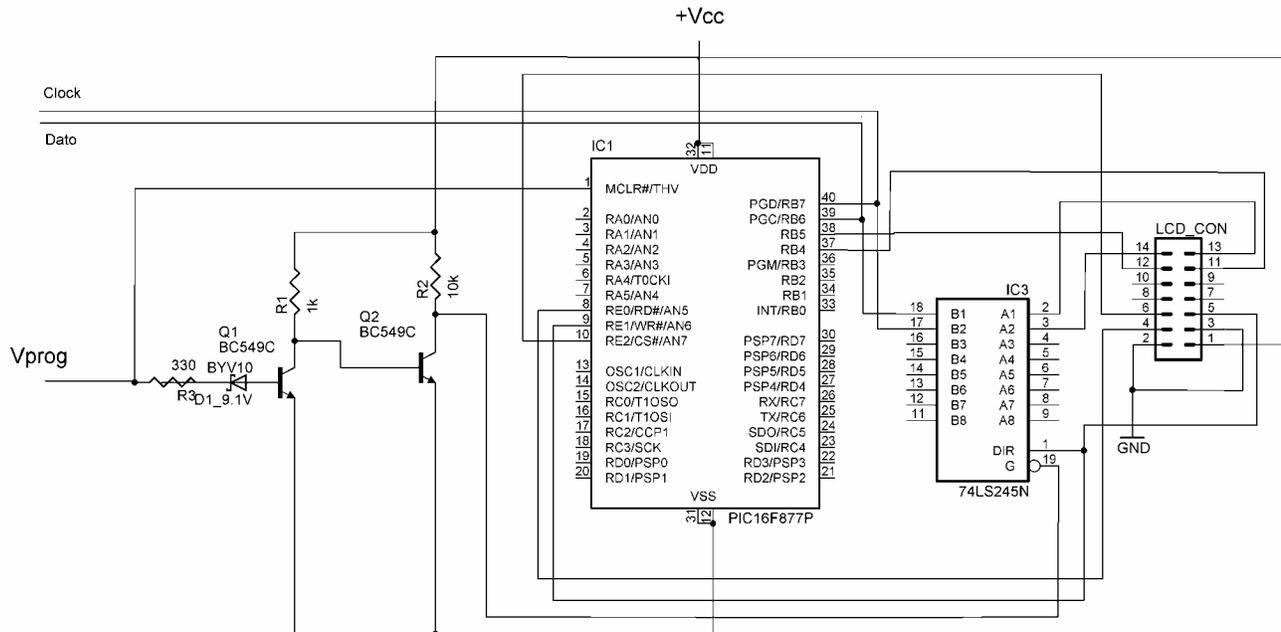


Figura 1.11, Conexión eléctrica de la Matriz LCD.

Para que la Matriz LCD funcione, luego de haber programado al PIC16F877, se manda un pulso que habilite el IC 74LS245, haciendo llegar un cero en su pata 19 (G), esto se logra con un circuito auxiliar detector de alto voltaje (mayor de 9.1V), formado por dos transistores conectados en cascada, que su entrada la controla un diodo Zener de 9.1 voltios, que su relación de entrada –salida se comporta como se muestra en la tabla 1.4.

V(MCLR)	ENABLE 74LS245
>9.1 Voltios	+5 Voltios
<9.1 Voltios	0 Voltios

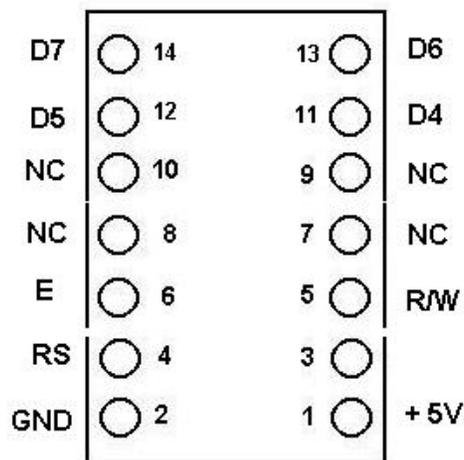
Tabla 1.4 Relación funcional entre la entrada y salida del circuito detector de alto voltaje.

Para complementar el proceso, hay que colocar un cero en la pata 1 (DIR), con esto; las señales que están presentes en RB7 y RB6, llegarán hasta D7 y D6 del LCD, con el resto de patas no se tiene problema ya que se conectan directamente.

A continuación se describe el conector LCD_CON para la LCD:

Número de Pin	Nombre	Descripción
1	+5V	Alimentación VCC
2	GND	Referencia
3	NC	No conectado
4	RS	Selección de Registro
5	R/W	Selección de Escritura y lectura
6	E	Inicio de escritura o lectura
7	NC	No conectado
8	NC	No conectado
9	NC	No conectado
10	NC	No conectado
11	D4	Dato
12	D5	Dato
13	D6	Dato
14	D7	Dato

Tabla 1.5 Descripción del conector LCD_CON



Conector para el LCD

Figura 1.12, Conector LCD_CON.

1.1.6 UNIDAD PROGRAMADORA

La Unidad Programadora se conecta desde el puerto paralelo de la computadora hasta las patas PB7 (data in/out), PB6 (clock) y MCLR (con lógica negada), ésta última es el control del modo de trabajo, donde si es mayor de 9V el PIC se coloca automáticamente en modo de programación y si es menor en modo de trabajo normal. Permitiendo la programación “In System Programming”.

El diseño de éste programador se basa en el IC 74LS05, que tiene por característica principal ser un Inversor con salida de colector abierto, esto último explica la existencia de las resistencias que conectan a la salida con $+V_{cc} = 5\text{ V}$. Las resistencias en la entrada de cada inversor, se utilizan para que en ausencia de la conexión física con el puerto LPT del PC, la compuerta “vea” un cero lógico, necesario para que el PIC tenga:

- a) +5V en la pata MCLR.
- b) +5V en la pata RB7 (data) y RB6 (clock).

En el caso de que RB7 y RB6 sean salidas, ellas comandarán el estado de la conexión, ya que una salida en ALTO de una compuerta de colector abierto, es equivalente a una conexión flotante, en otras palabras, la salida de la compuerta de colector abierto “sigue” a la salida del PIC (Puerto B).

Cuando RB7 y RB6 sean entradas, pero no controladas por el programador ($V_{MCLR} \neq 12\text{ V}$, sino 5.0V), en éste caso, serán controladas por la pata D7 y D6 de la matriz LCD, entonces, las salidas de colector abierto de la compuerta seguirán al valor que se coloque en dichos pines.

El diagrama de la unidad programadora tipo TAIT, se muestra en la figura 1.13.

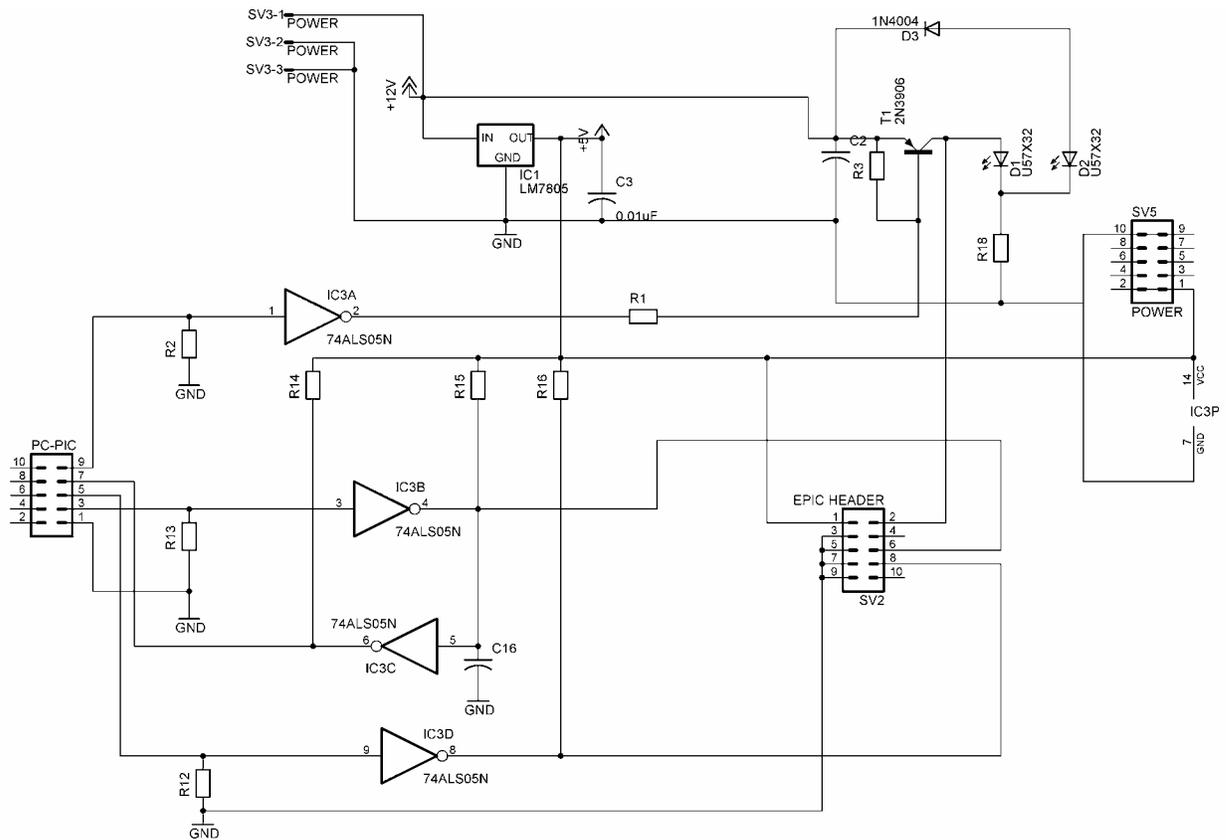


Figura 1.13, Programador para PIC16F877

Descripción de conectores:

Conector Número de Pin	PC-PIC	EPIC HEADER	SV3 POWER	SV5 POWER
1	GND(Referencia)	Alimentación +5V	Alimentación +12V	Alimentación +5V
2	NC(No conectado)	MCLR(PIC)	GND(Referencia)	NC(No conectado)
3	D0(LPT)	GND(Referencia)	GND(Referencia)	NC(No conectado)
4	NC(No conectado)	NC(No conectado)		NC(No conectado)
5	D1(LPT)	GND(Referencia)		NC(No conectado)
6	NC(No conectado)	RB7(PIC) Datos		NC(No conectado)
7	ACK(LPT)	GND(Referencia)		NC(No conectado)
8	NC(No conectado)	RB6(PIC) Clock		NC(No conectado)
9	D3(LPT)	GND(Referencia)		NC(No conectado)
10	NC(No conectado)	NC(No conectado)		GND(Referencia)

Tabla 1.6 Descripción de Conectores del Programador para el PIC16F877

1.2 MÉTODO DE FABRICACIÓN DEL PROTOTIPO

En la fabricación del hardware del prototipo, se han ocupado los siguientes recursos:

- a) SOFTWARE: EAGLE (Easily Applicable Graphical Layout Editor)
- b) HARDWARE: BROMOGRAFO
- c) ACCESORIOS: TABLETA PRESENSIBILIZADA, SOSA CÁUSTICA, ÁCIDO NÍTRICO COMERCIAL, THINNER.

El proceso de fabricación general se enumera a continuación:

1. Dibujar en el editor de circuitos EAGLE, el diagrama eléctrico del PLC
2. Con el diagrama eléctrico diseñado, se procede con la herramienta “BOARD” de EAGLE, para obtener el diagrama impreso(pistas).
3. Imprimir el diseño de las pistas (top y botton) en un acetato.
4. Con el BROMOGRAFO, imprimir el diseño de las pistas en la tableta.
5. Limpiar y quemar la tableta cobreada.
6. Soldar los elementos.

1. Dibujar con el editor de circuitos EAGLE, el diagrama eléctrico del PLC.

Haciendo uso del editor gráfico para dibujar diagramas esquemáticos, se dibujara el diagrama eléctrico del PLC, tal como se muestra en la figura 1.14.

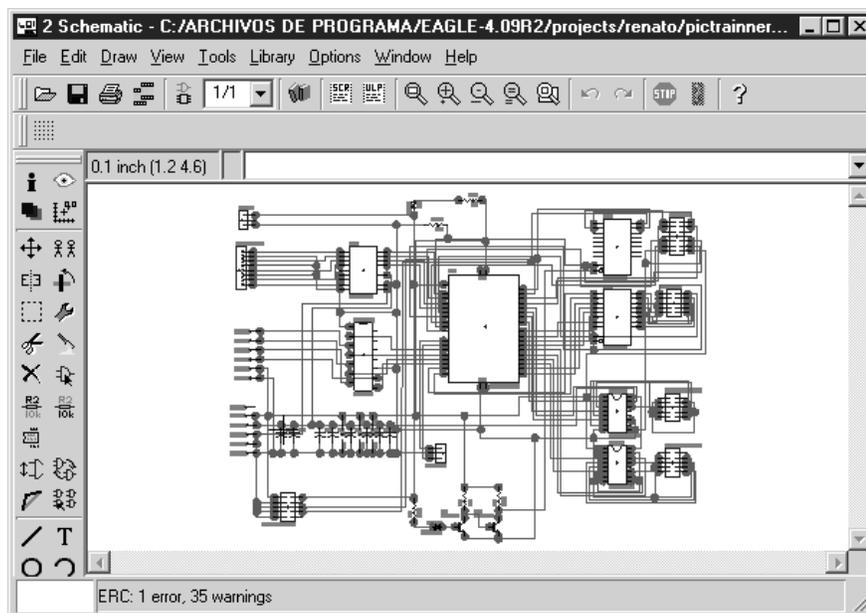


Figura 1.14, Diagrama completo del prototipo de PLC

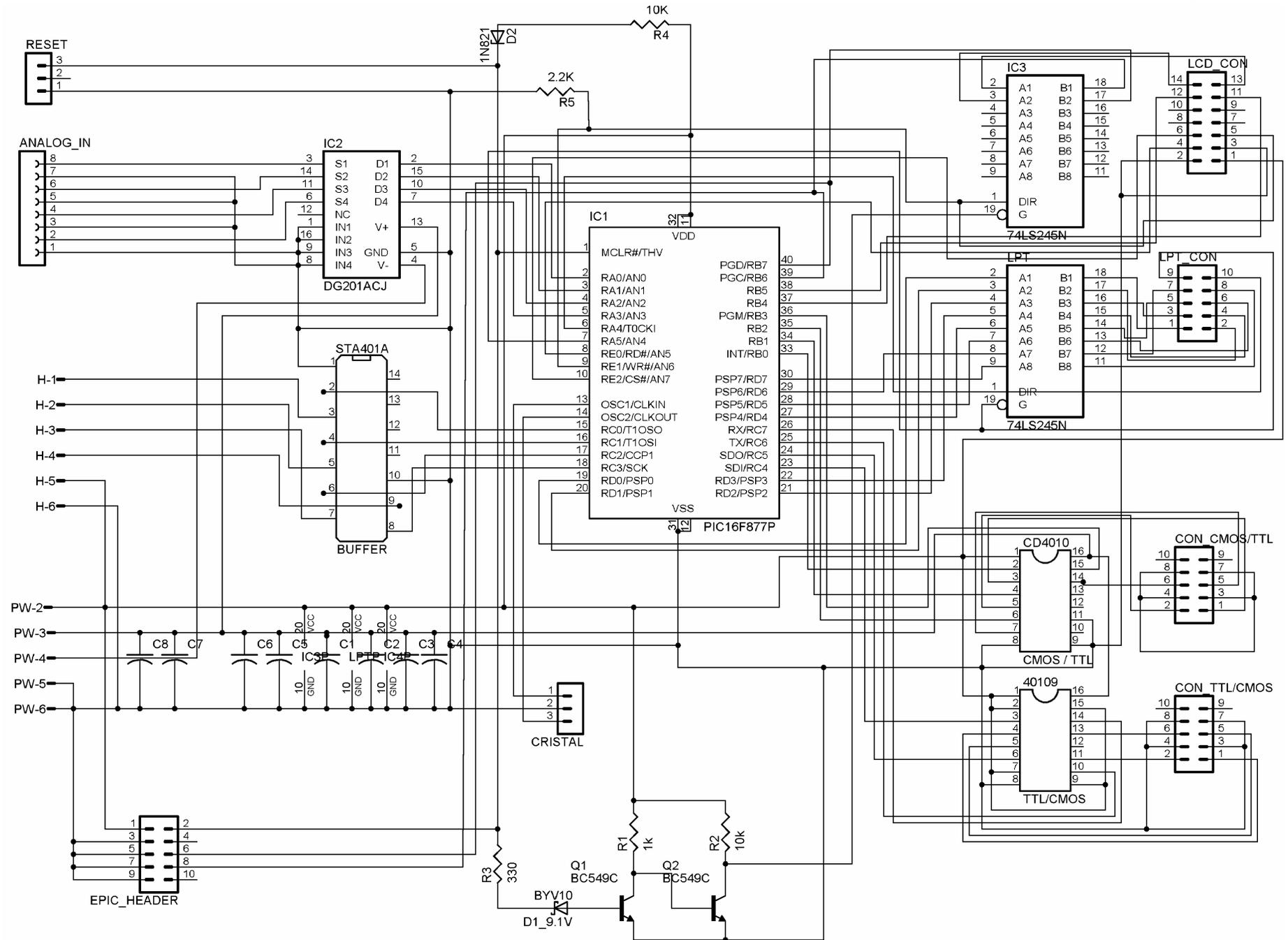


Figura 1.15 Diagrama eléctrico que muestra todos los componentes del PLC Prototipo.

Descripción de los conectores:

a) Conector de Entradas Analógicas:

Numero de Pin	Descripción
1	GND(Referencia)
2	AN4(Entrada analógica 4)
3	GND(Referencia)
4	AN3(Entrada analógica 3)
5	GND(Referencia)
6	AN2(Entrada analógica 2)
7	GND(Referencia)
8	AN1(Entrada analógica 1)

Tabla 1.7 Entradas Analógicas ANALOG_IN.

b) Conector PW: alimentación general de la tarjeta principal.

Numero de Pin	Descripción
1	NC(No conectado)
2	Vcc=+5V (alimentación)
3	Vdd=+12V(alimentación)
4	Vee=-12V(Alimentación)
5	Vss=GND(Referencia)
6	Vss=GND(Referencia)

Tabla 1.8 Conector de Alimentación PW.

c) Conector RESET:

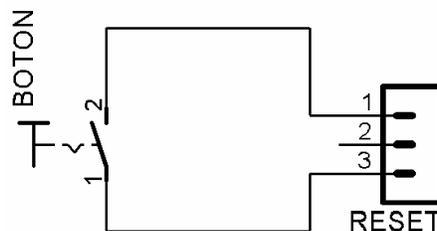


Figura 1.16 Conexión del botón de RESET general.

- d) Conector para las Salidas de Alto Voltaje CON_TTL/CMOS. Ver Tabla 1.3
- e) Conector para las Entradas de Alto Voltaje CON_CMOS/TTL. Ver Tabla 1.2
- f) Conector para la LCD. Ver Tabla 1.5
- g) Conector para las salidas de Alta potencia (3Amp). Ver figura 1.9.
- h) Conector para el programador del PIC EPIC_HEADER. Ver Tabla 1.6.

i) Conector para el Cristal de Cuarzo.

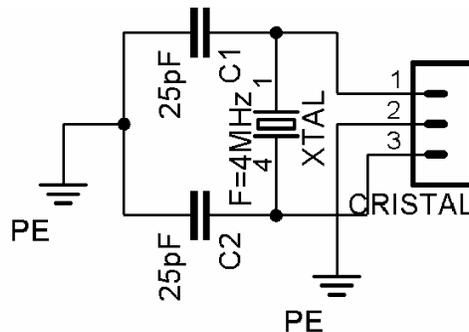


Figura 1.17 Conexión del resonador de cuarzo.

j) Vista de la bornera en el chasis, con todas las conexiones disponibles para la utilización del PLC:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

Figura 1.18 Vista Frontal de la bornera en el Chasis del PLC.

[1] GND	[11] OUTL1 Salida CMOS Digital VOL=0.5max VOH=10Vmin IC-HCF40109B
[2] +12V / 3Amp	[12] OUTL2 Salida CMOS Digital VOL=0.5max VOH=10Vmin IC-HCF40109B
[3] +5V / 3Amp	[13] OUTL3 Salida CMOS Digital VOL=0.5max VOH=10Vmin IC-HCF40109B
[4] IN1 Entrada CMOS Digital Vi=-0.5 a 13V Ii=±10mA VIH=9Vmin VIL=1.5Vmax IC- HCF4010B	[14] OUTL4 Salida CMOS Digital VOL=0.5max VOH=10Vmin IC-HCF40109B
[5] IN2 Entrada CMOS Digital Vi=-0.5 a 13V Ii=±10mA VIH=9Vmin VIL=1.5Vmax IC- HCF4010B	[15] OUTH1 Salida de Alta Corriente Colector Abierto VCEO=60±10V ICmax=4Amp IC-STA401A
[6] IN3 Entrada CMOS Digital Vi=-0.5 a 13V Ii=±10mA VIH=9Vmin VIL=1.5Vmax IC- HCF4010B	[16] OUTH2 Salida de Alta Corriente Colector Abierto VCEO=60±10V ICmax=4Amp IC-STA401A
[7] IN4 Entrada CMOS Digital Vi=-0.5 a 13V Ii=±10mA VIH=9Vmin VIL=1.5Vmax IC- HCF4010B	[17] OUTH3 Salida de Alta Corriente Colector Abierto VCEO=60±10V ICmax=4Amp IC-STA401A
[8] AN1 Entrada Analógica Digital, Vi=0min a 5Vmax Iimax=20mA IC-DG201A	[18] OUTH4 Salida de Alta Corriente Colector Abierto VCEO=60±10V ICmax=4Amp IC-STA401A
[9] AN2 Entrada Analógica Digital, Vi=0min a 5Vmax Iimax=20mA IC-DG201A	[19] AN5 Entrada Analógica Digital, Vi=0min a 5Vmax Iimax=20mA IC-DG201A
[10] AN3 Entrada Analógica Digital, Vi=0min a 5Vmax Iimax=20mA IC-DG201A	[20] AN4 Entrada Analógica Digital, Vi=0min a 5Vmax Iimax=20mA IC-DG201A

Tabla 1.9 Descripción de los bornes

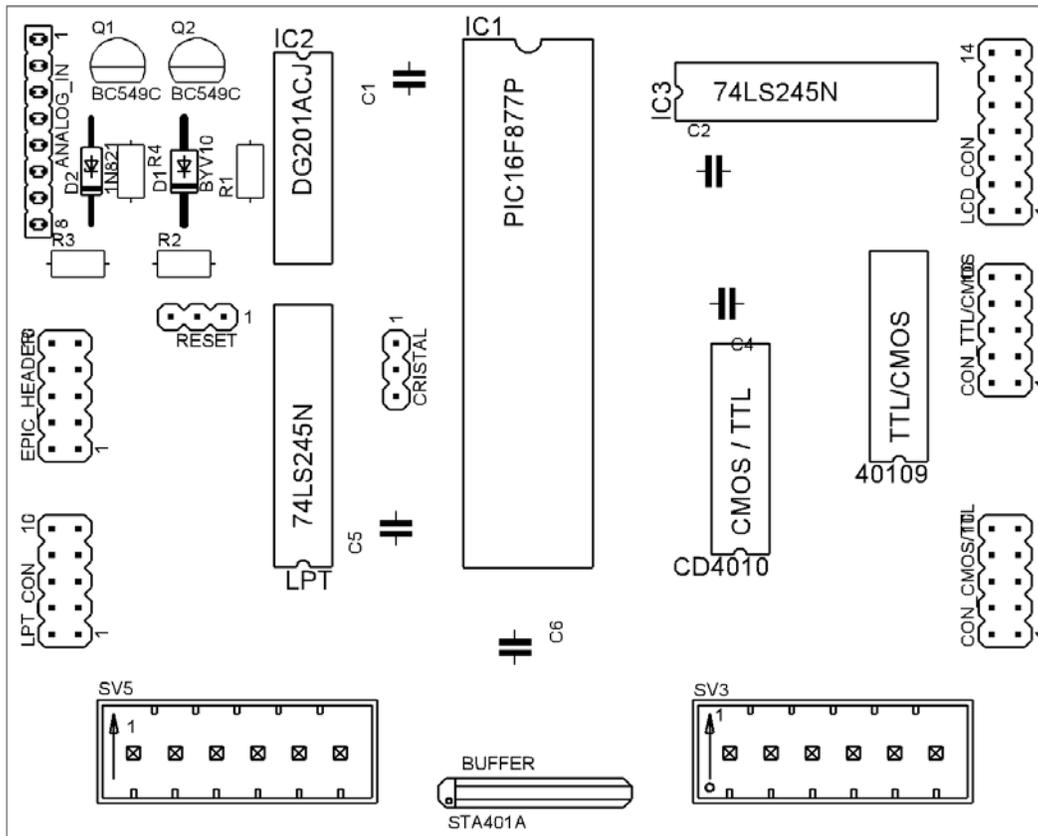


Figura 1.21 Distribución de los componentes en el PCB

3. Procedimiento de impresión: utilizando un impresor láser, por tener la mayor calidad de estampado y posibilidad de utilizar acetatos de tipo comercial; se imprime la parte superior e inferior en el mismo acetato.

4. Luego se quema con el bromógrafo, la tableta presensibilizada. El procedimiento sugerido es el siguiente:

1. Adentro del bromógrafo, colocar el acetato en la parte superior de la tableta.
2. Activar el presurizador del bromógrafo.
3. Ajustar el tiempo de exposición, uno o dos minutos, según la calidad de la tableta.
4. Limpiar la tableta con una solución de sosa cáustica¹³ al 10%, diluida con agua.
5. Quitar el cobre residual de la placa con una solución de ácido nítrico¹⁴ comercial al 25%. (tener precaución de no absorber el vapor resultante por la toxicidad).

¹³ Soda Cáustica o Hidróxido de Sodio: Químico muy corrosivo, fórmula es NaOH.

¹⁴ Ácido nítrico: líquido incoloro y corrosivo cuya fórmula química es HNO₃.

1.3 PROGRAMACIÓN Y FUNCIONAMIENTO PARTICULAR DEL PROTOTIPO

1.3.1 DESCRIPCIÓN DEL LENGUAJE ENSAMBLADOR

1.3.1.1 ENSAMBLADOR MPASM

En el entorno de Microsoft Windows, se cuenta con el ensamblador de Microchip (MCU) el cual es gratuito y puede descargarse desde su pagina web¹⁵ y es el MPASM. Para usar este ensamblador Microchip MPASM requiere un PC/AT de IBM® o compatible, utilizando el OS¹⁶ MS-DOS ® V5.0 o mayor, Microsoft ® Windows 95/98/NT/2000.

Éste ensamblador (MPASM) soporta todas las versiones de microcontroladores de la familia de Microchip¹⁷.

1.3.1.1.1 VISIÓN GENERAL DEL ENSAMBLADOR MPASM

El ensamblador MPASM puede ser usado en dos formas:

- ❑ Para generar código absoluto que puede ser ejecutado directamente por un microcontrolador (PIC).
- ❑ Para generar código objeto el cual se puede conectar con otros módulos, armados o compilados separadamente.

a) GENERANDO CÓDIGO ABSOLUTO

El código absoluto es la salida predeterminada de ensamblador MPASM. Este proceso es demostrado en la siguiente figura:

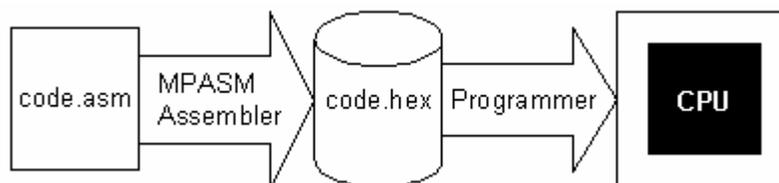


Figura 1.22 Generación de Código Absoluto

¹⁵ www.microchip.com

¹⁶ OS: Sistema Operativo

¹⁷ Debe pertenecer a la familia Microchip MCU.

Cuando un archivo fuente es armado de ésta manera, todos los valores usados en el archivo fuente deben estar definidos dentro, o en archivos que han estado explícitamente incluidos, también llamados archivos de cabecera. Si el proceso de ensamblaje procede sin errores, entonces un archivo HEX¹⁸ será generado, conteniendo el código ejecutable en lenguaje de máquina para el dispositivo que va a ser programado, el cual está en blanco o sin programa. Este archivo puede entonces ser usado por un programador para descargar el archivo al microcontrolador.

b) GENERANDO CÓDIGO OBJETO

El ensamblador MPASM también tiene la habilidad de generar un módulo objeto que puede ser utilizado por un enlazador como MPLINK Microchip, que usa módulos para formar el código ejecutable final. Este método es muy útil para crear módulos reutilizables, como se muestra en la siguiente figura:

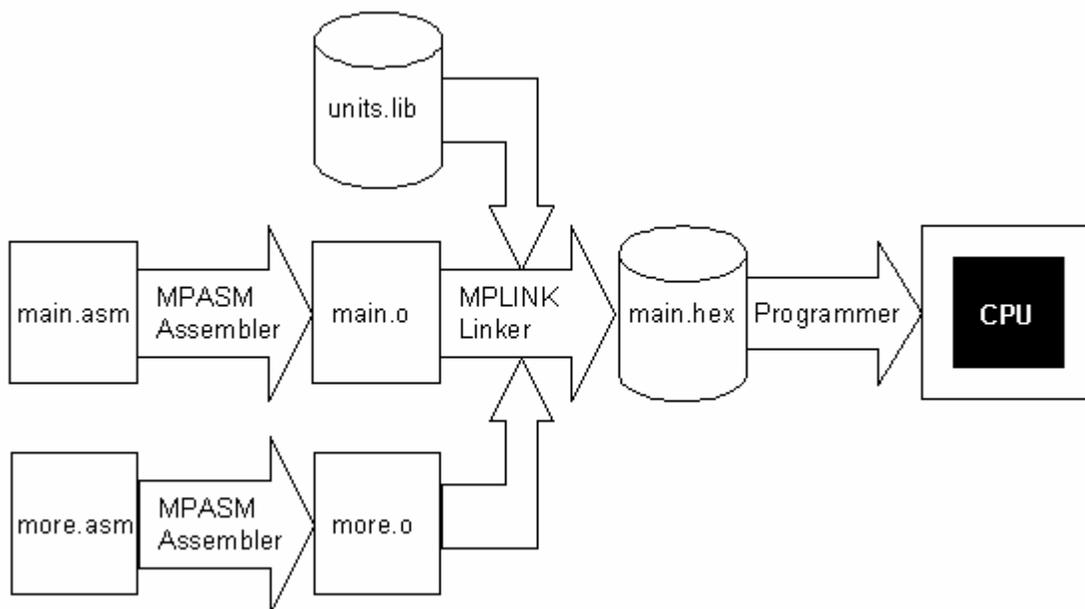


Figura 1.23, Generación de Código Abierto

¹⁸ Formato HEX: Tipo de formato estándar INTEL INHX8M para codificación.

1.3.1.1.2 LOS ARCHIVOS DE ENTRADA/SALIDA DEL ENSAMBLADOR

Las extensiones predeterminadas usadas por el ensamblador:

Los Archivos de Entrada:

- Código Fuente (.asm)
- Archivos de inclusión (.inc)

Los Archivos de Salida:

- Listando (.lst).
- Archivo de Error (.err).
- Archivo Hex(.hex, .hxl, .hxh).
- Archivo de Depuración(.cod).
- Archivo Objeto(.o).

a) ARCHIVO DE CÓDIGO FUENTE (.ASM):

El archivo de código fuente puede ser creado usando cualquier programa editor de texto ASCII¹⁹. Cada línea del archivo fuente .ASM, debe de contener cuatro tipos de información diferente:

- Etiqueta.
- Mnemotécnicos.
- Operando.
- Comentarios.

El orden y la colocación de estos se describen a continuación:

- a) Las etiquetas deben empezar la columna.
- b) Los mnemotécnicos pueden empezar columna.
- c) Los operandos son los valores de los mnemotécnicos (tercera columna).
- d) Los comentarios pueden seguir a los operandos, mnemotecnia o etiquetas, y pueden empezar en cualquier columna. En ancho máximo de la columna es 255 caracteres.

¹⁹ La generación del archivo ASCII se puede hacer con los siguientes programas: NOTEDPAD, EDIT de DOS, etc.

1.3.1.1.3 INSTALACIÓN DEL ENSAMBLADOR

Hay dos versiones de ensamblador proporcionados por Microchip:

- ❑ Versión para MSDOS, llamada MPASM.EXE (MSDOS 5.0 o mayor).
- ❑ Versión para Windows, MPASMWIN.EXE.

MPASM.EXE puede ser usado en MSDOS o una ventana de DOS en NT/2000 Windows 3.x, Windows 95/98. También se puede usar con MPLAB IDE, aunque MPASMWIN.EXE es recomendable.

El MPASMWIN.EXE tiene una interfaz para Windows, la cual se muestra en la siguiente figura:



Figura 1.24, Interfaz gráfica del ensamblador MPASM.

Nota: El macro ensamblador MPASM se instala adjunto al MPLAB – IDE de Microchip.

1.3.1.2 ENSAMBLADOR BAJO LA PLATAFORMA LINUX (RED HAT 7.0)

Uno de los ensambladores de formato INTEL es el GPASM, el cual se puede obtener gratuitamente desde Internet, el archivo asociado es “gpasm-0.9.14-1.i386.rpm”.

GPASM es un programa ensamblador equivalente Microchip™ MPASM, por lo que se puede obtener los mismos resultados con él, obteniendo los archivos necesarios para programar un PIC específico.

El programa se ejecuta desde la línea de comandos, sus argumentos se describen a continuación:

```
[ ]#gpasm <opciones> <archivo.asm>
```

Donde las opciones son:

- a<formato> : produce un archivo HEX con uno de los siguientes formatos:
INHX8M, INHX8S, INHX32.
- d : símbolo, que es equivalente a #define.
- p <procesador> define el tipo de dispositivo a utilizar.
- r<RADIX> tipos de datos utilizados.
- v: Versión del ensamblador.

Ejemplo 1.3 Aplicación del ensamblador GPASM.

- #gpasm prototipo_plc.asm
Descripción: Ésta es la forma mas sencilla de trabajar con el ensamblador, porque se trabaja con los valores por defecto.
- #gpasm
Descripción: Al escribir solamente el nombre del ensamblador, se muestra toda la ayuda disponible por parte del programa.
- #gpasm a INHX8M p PIC16F84A r HEX prototipo_plc.asm
Descripción: Ésta forma genera un archivo HEX a partir del .ASM especificando el tipo de archivo(INHX8M), el microprocesador(PIC16F84A) y el tipo de datos utilizados por defecto(HEX).

1.3.2 DESCRIPCIÓN GENERAL DE LOS PROGRAMADORES

1.3.2.1 PROGRAMADOR BAJO LA PLATAFORMA MICROSOFT (IC-PROG 1.05A)

Programador Prototipo IC-Prog es un software de programación basado en ventanas que permite la programación de diversos dispositivos y soporta numerosos tipos de programadores²⁰ tales como: Memorias, Familia PIC de Microchip, etc.

Para instalar el hardware de programación, se debe de seguir la siguiente rutina:

- a) Conectar el programador al puerto correspondiente (paralelo o serie) por medio de su respectivo cable de 25 pines(LPT) o 9 pines(SERIAL) según sea el caso.
- b) Verificar que la configuración del programador, corresponda con las recomendaciones efectuadas por el fabricante de la interfaz a utilizar. En el caso del PLC, está ocupando la siguiente configuración:
 - Dispositivo Programador: TAIT Serial Programmer.
 - Puerto: LPT1.
 - Sistema Operativo probados: Windows 98/ME/2000.
 - Interfaz: Direct I/O
 - Dispositivos: PIC16F877, PIC16F877A
 - Configuración de Pines del Puerto Paralelo(PC) LPT1:
 - Invertir DATA OUT
 - Invertir DATA IN
 - Invertir CLOK
 - Invertir MCLR
 - Invertir VCC
- c) Proceder a programar el PIC respectivo con la información del archivo HEX.

²⁰ Dispositivos soportados: a) eeproms: 59C11, 59C22, 59C13 b) I²C eeproms: 24C01, 24C02, 24C04, 24C08, 24C16, 24C32, 24C64/65, 24C128, 24C256, 24C512, PCF8572, 24C01, c) Flash Microcontrollers: 89C1051, 89C2051, 89C4051 d) IM-Bus eeproms: NVM3061, MDA2060, MDA2061, MDA2062 e) Microchip PIC devices: 12C508, 12C508A, 12C509, 12C509A, 12CE518, 12CE519, 12C671, 12C672, 12CE673, 12CE674, 16C433, 16C54, 16C56, 16C61, 16C62A, 16C62B, 16C63, 16C63A, 16C64A, 16C65A, 16C65B, 16C66, 16C67, 16C71, 16C72, 16C72A, 16C73A, 16C73B, 16C74A, 16C76, 16C77, 16C84, 16F83, 16F84, 16F84A, 16C505, 16C620, 16C621, 16C622, 16C622A, 16F627, 16F628, 16C715, 16F870, 16F871, 16F872, 16F873, 16F874, 16F876, 16F877, f) Microwire eeproms: 93C06, 93C46, 93C57, 93C56, 93C66, 93C76, 93C86, 93C13, 93C14, etc.

La interfaz gráfica IC-Prog, muestra la información necesaria para programar el dispositivo seleccionado, ver figura 1.25.

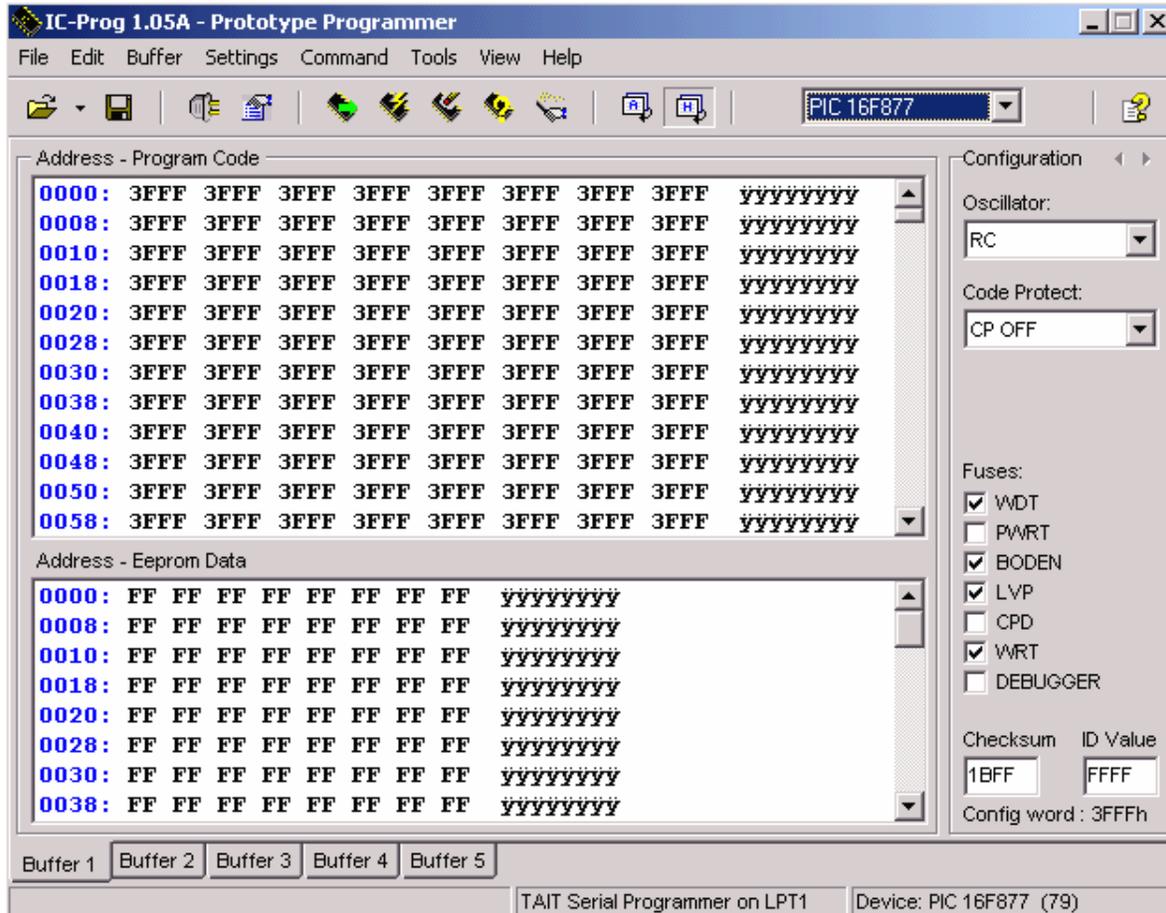


Figura 1.25 Vista de la interfaz grafica IC-Prog 5.05A

Todos los dispositivos poseen al menos un área de memoria de Código en la que se puede almacenar el programa (las memorias EEPROM poseen solo esta área de código).

Otros dispositivos, como la mayoría de los microcontroladores, poseen áreas adicionales de almacenamiento, como el área de Datos. Normalmente el área de Código contiene el programa a ser ejecutado por el microcontrolador, mientras que el área de Datos puede contener tablas, valores constantes, etc.

La mayoría de los microcontroladores (como los PIC) poseen también un área de configuración. La configuración servirá para establecer los parámetros iniciales para cuando el microcontrolador sea energizado. Ésta información es diferente y única, y

depende de la aplicación en la que se desenvuelva PIC; la forma de cómo será configurado se encuentra establecida en las hojas de datos del chip respectivo.

Descripción de las secciones de control en la ventana de IC-Prog:

a) Área de código (Memoria para programas):

Cada fila en el área de Código muestra 8 palabras, por lo que de una fila a la otra la dirección se incrementa en 8. Una palabra posee normalmente una longitud de 16 bits por lo que el IC-Prog mostrará un valor hexadecimal entre 0000 y 3FFF. Algunos dispositivos sólo poseen longitudes de palabras de 14 bits, 12 bits o 8 bits, por lo que el máximo valor hexadecimal es 3FFF, 0FFF 00FF respectivamente, aunque el programa siempre muestra el valor hexadecimal utilizando 4 dígitos. El valor en formato carácter sólo utiliza los 8 bits más bajos de la palabra de 16 bits, ya que el rango estándar de los caracteres sólo va de 0 a 255 (8 bits).

b) Área de datos (Memoria de tipo EEPROM):

Cada fila en el área de Datos muestra 8 palabras, pero en este caso las palabras son siempre de 8 bits, por defecto se muestra las palabras con 2 dígitos hexadecimales, con un valor entre 00 y FF.

c) Área de configuración:

El área de Configuración para el dispositivo, muestra listas de opciones u opciones de selección. El usuario puede seleccionar fácilmente la configuración deseada, y el IC-Prog calculará la correspondiente palabra de configuración. La palabra de configuración se muestra en la parte inferior de esta área. Los elementos dependen del dispositivo, por lo que el área de Configuración será distinta de acuerdo al dispositivo seleccionado (consulte las hojas de datos del dispositivo para encontrar la información de configuración específica). Según la selección realizada, IC-Prog define un cero o un uno para un bit específico en la palabra de configuración. Existen casos en que un elemento de configuración es invertido en el interior del dispositivo. IC-Prog invierte automáticamente un elemento de configuración en caso de ser necesario. Para habilitar un elemento de configuración marque la casilla de selección, para deshabilitarlo desmarque dicha casilla.

1.3.2.2 PROGRAMADOR BAJO LA PLATAFORMA LINUX RED-HAT 7.0

El programa PICProg V2.2.3 es un programa de distribución gratuita, el cual permite programar o escribir en la memoria de los dispositivos PICmicro de Microchip un programa, por medio del Hardware tipo TAIT, descrito anteriormente (sección 1.1.6). Se ejecuta en la línea de comandos TERMINAL así:

```
#picprg -p[0-2] -i[hex] -e[hex] <Archivo.HEX>
```

La interfaz gráfica se presenta en la siguiente figura:

```
Linux PIC programmer v2.2.3-20010810 Copyright 1994-1998 by Brian C. Lane
PIC16F84 CLOCK: RC CP: OFF WD: ON PU: ON

[L]oad a file
[W]rite RAM to file
[D]isplay RAM
[F]ill RAM

[R]ead contents of PIC into RAM
[P]rogram PIC from RAM
[V]erify PIC with RAM
[E]rase PIC
[B]lanchek PIC

[C]onfigure PICPGMR
[T]est PICPGMR
[Q]uit
```

Figura 1.26 Vista del PICPRG 2.2.3 Linux

El programador tiene las siguientes opciones:

- [L] Cargar un archivo HEX a la memoria del programador Picprg.
- [W] Guardar la información leída de un dispositivo(PIC) a un archivo.
- [D] Mostrar el contenido de la memoria del programador.
- [F] Ajuste de la memoria del programador.
- [R] Leer el contenido del PIC.
- [P] Escribir el programa HEX a la memoria del PIC.
- [V] Verificar el programa descargado en el PIC con el programa en memoria del programador.
- [E] Borrar la memoria del PIC

- [B] Verificar que el PIC esté en blanco.
- [C] Configurar el PICprg para el Hardware a utilizar, la configuración es de la siguiente forma:
 - Printer Port: 0
 - Vpp: +05 OFF
 - Vdd: +04 OFF
 - PGM: +06 OFF
 - Clock: -03 OFF
 - Data to PIC: -02 OFF
 - Data from PIC: -10 OFF
- [T] Examinar el hardware de programación
- [Q] Salir del programa.

Ejemplo 1.4 Aplicación del programador PICPRG.

- #picprg prototipo_plc.hex
 Descripción: Esta es la forma más sencilla de trabajar con el programador, porque se trabaja con los valores por defecto; se descarga inmediatamente el archivo prototipo_plc.hex al PIC, seccionado automáticamente.
- #picprg
 Descripción: Despliega en pantalla la interfaz gráfica, teniendo las herramientas necesarias para configurar y programar el PIC.
- #picprg -p0 -ihex prototipo_plc.hex
 Descripción: Programa de forma inmediata al PIC, especificando el puerto(-p0 es LPT1), tipos de datos (-ihex) y el archivo conteniendo el programa para el PIC.

1.3.3 DISEÑO DE MANEJADORES DE LOS COMPONENTES DEL PLC

El diseño de las funcionalidades del PLC utilizando el PIC16F877 de Microchip™, se basa en las siguientes especificaciones:

- a) Funciones básicas: AND, OR, NOT, NAND, NOR, XOR.
- b) Retardo de activación.
- c) Retardo de desactivación.
- d) Relé de Temporización.
- e) Contador progresivo/regresivo.
- f) Entradas analógicas.
- g) Entradas y Salidas según la normativa CMOS.
- h) Salida de potencia de hasta 3Amp máximo.
- i) I/O para señales TTL.

1.3.3.1 MÓDULO CONTADOR

El contador, funciona como un dispositivo diseñado para llevar el registro de pulsos que se introducen desde un circuito de acondicionamiento, éste conteo puede ser ocupado para diferentes actividades, donde se necesite cuantificar materiales, procesos, etc.

El contador tiene las siguientes características:

- El contador se puede programar para que funcione de forma ascendente (progresivo) o descendente (regresivo).
- Es posible especificar el número de cuenta máxima que deba realizar, en el rango de 0 a 999.
- Cuenta con una señal la cual se activa cuando existe sobre flujo ascendente (900→0) o descendente (0→999), esta señal es generada en la salida (OUTL).
- Los pulsos que alimentan este circuito, son enviados a la entrada de alto voltaje CMOS (IN); la frecuencia máxima de los pulsos es de 30Hz²¹.

²¹ Frecuencia máxima del contador es debido a razones internas del programa, el cual es lento para la captura del pulso. Las pruebas de frecuencia se han realizado en el laboratorio.

A continuación se muestra un circuito propuesto para trabajar con el contador:

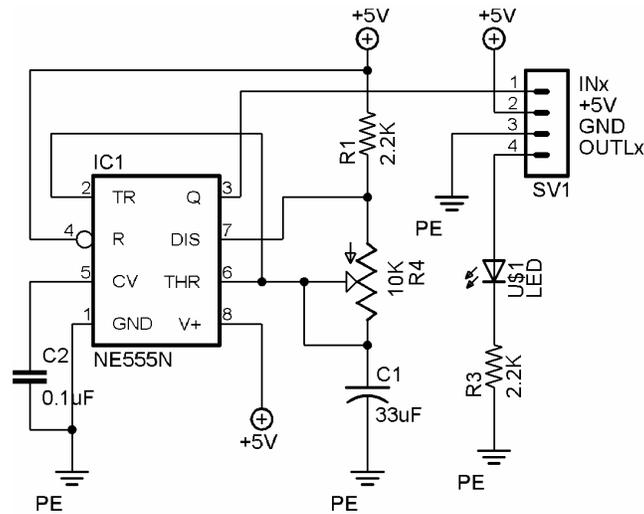


Figura 1.27 Generador de pulsos para el MÓDULO contador.

A continuación se presenta el diagrama de flujo respectivo:

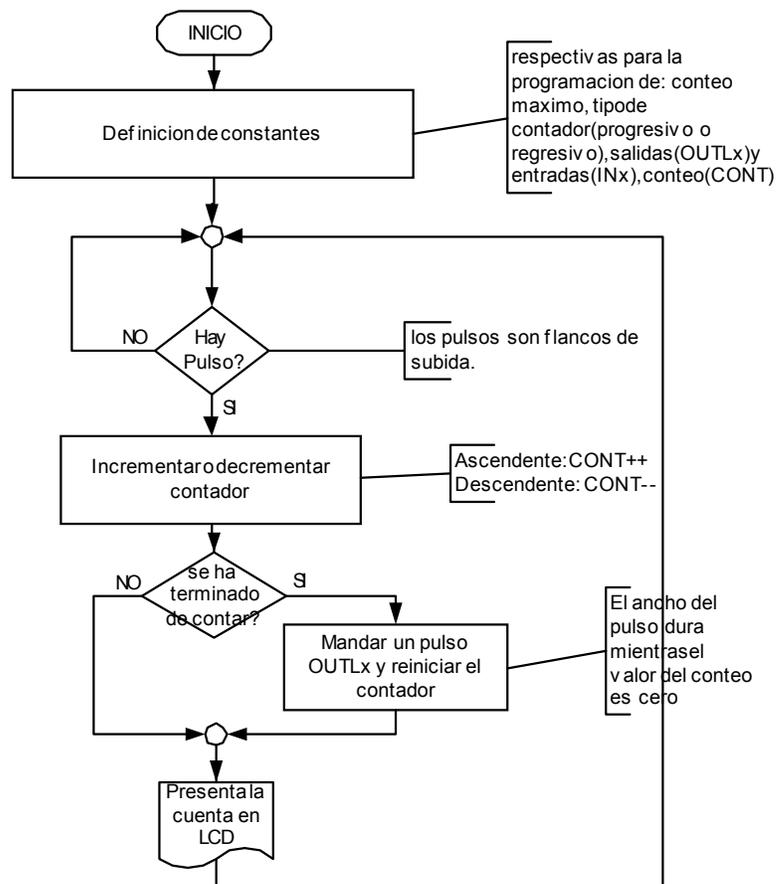


Figura 1.28 Diagrama de flujo del Módulo Contador Progresivo/Regresivo

1.3.3.2 MÓDULO GENERACIÓN DE SEÑALES TEMPORIZADAS

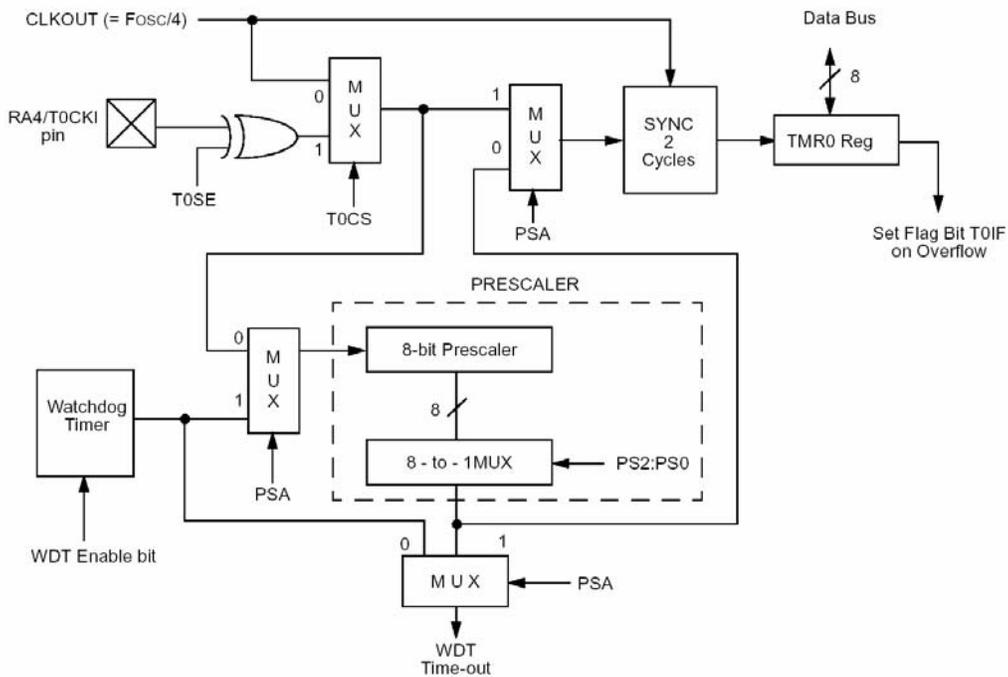
En la generación de señales temporizadas, se utilizan recursos disponibles por el microprocesador, los cuales son: Temporizador principal (TMR0) y Temporizador secundario (TMR1).

1.3.3.2.1 MÓDULO TMR0

El módulo TMR0 es un temporizador/contador con las siguientes características:

- El temporizador/contador dispone de 8 bits.
- Puede leerse y modificarse por software.
- Preescalamiento programable vía Software de 8 bits.
- Puede trabajar con una señal de reloj interno o con una señal de reloj externa.
- Dispone de una interrupción por desbordamiento al pasar de FFh a 00h.
- Selección de flanco ascendente o descendente para el flanco del reloj externo.

En la siguiente figura se muestra un diagrama de bloques del TMR0.



Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION_REG<5:0>).

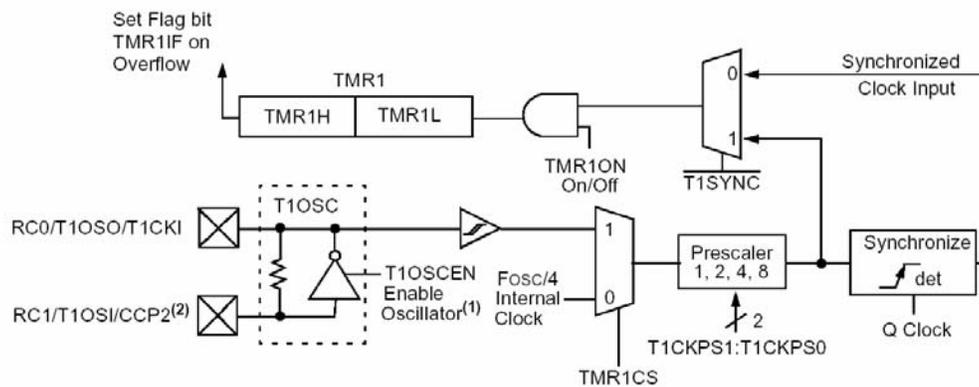
Figura 1.29 Diagrama a bloques de TMR0

1.3.3.2.2 MÓDULO TMR1

El módulo TMR1 es un temporizador/contador de 16 bits, formado por dos registros de 8 bits cada uno TMR1H y TMR1L, que son de lectura y escritura y que son los que guardan el valor de la cuenta en cada momento. El valor de los registros TMR1H y TMR1L se incrementan desde 0x0000 hasta 0xFFFF, en cuyo instante se vuelve a poner a 0x0000 y lo cual activa la bandera TMRIF (PIR1<0>).

El modo de funcionamiento de TMR1 como temporizador, se selecciona poniendo a cero el bit TMR1CS (T1CON <1>). En este modo la señal de reloj de alimentación al TMR1 es el reloj de interno del microcontrolador (FOSC/4). En este modo de trabajo el bit T1SYNC (T1CON <2>) no tiene ningún efecto ya que el reloj interno está siempre sincronizado.

En la siguiente figura se muestra el diagrama de bloques para TMR1:



Note 1: When the T1OSCEN bit is cleared, the inverter is turned off. This eliminates power drain.

Figura 1.30 Diagrama a bloques de TMR1

1.3.3.2.3 USO DE LOS TEMPORIZADORES TMR0 Y TMR1

Para generar temporizaciones se debe tener en cuenta el ancho de los pulsos a generar. Para generar tiempos pequeños se utiliza TMR0 y para tiempos largos se utiliza TMR1.

$$Temporizacion = \frac{4}{XTAL} (\text{prescalamiento})(\text{Conteos_TMRx})(\text{Variable_Local})$$

Ecuación 1.2 Temporización

Donde:

XTAL: es el valor del oscilador colocado al uC.

Preescalamiento: es el valor a dividir del reloj interno, para aumentar el retardo.

Conteos_TMRx: cantidad de conteos que debe realizar el temporizador.

Variable_Local: variable para aumentar el tiempo.

a) Temporización utilizando TMR0.

Por la Ecuación 1.2, se puede tener un número máximo de temporización, que al trabajar con un XTAL = 4MHz, prescalamiento = 256, conteos_TMR0 = 256 y Variable_Local = 256 nos da un valor de temporización de 16.77seg.

b) Temporización utilizando TMR1.

Por la Ecuación 1.2, se puede tener un número máximo de Temporización, que al trabajar con un XTAL = 4MHz, prescalamiento = 8, conteos_TMR1 = 2^{16} y Variable_Local = 256 nos da un valor de temporización de 134.21seg.

En la figura 1.31, se muestra el diagrama de flujo:

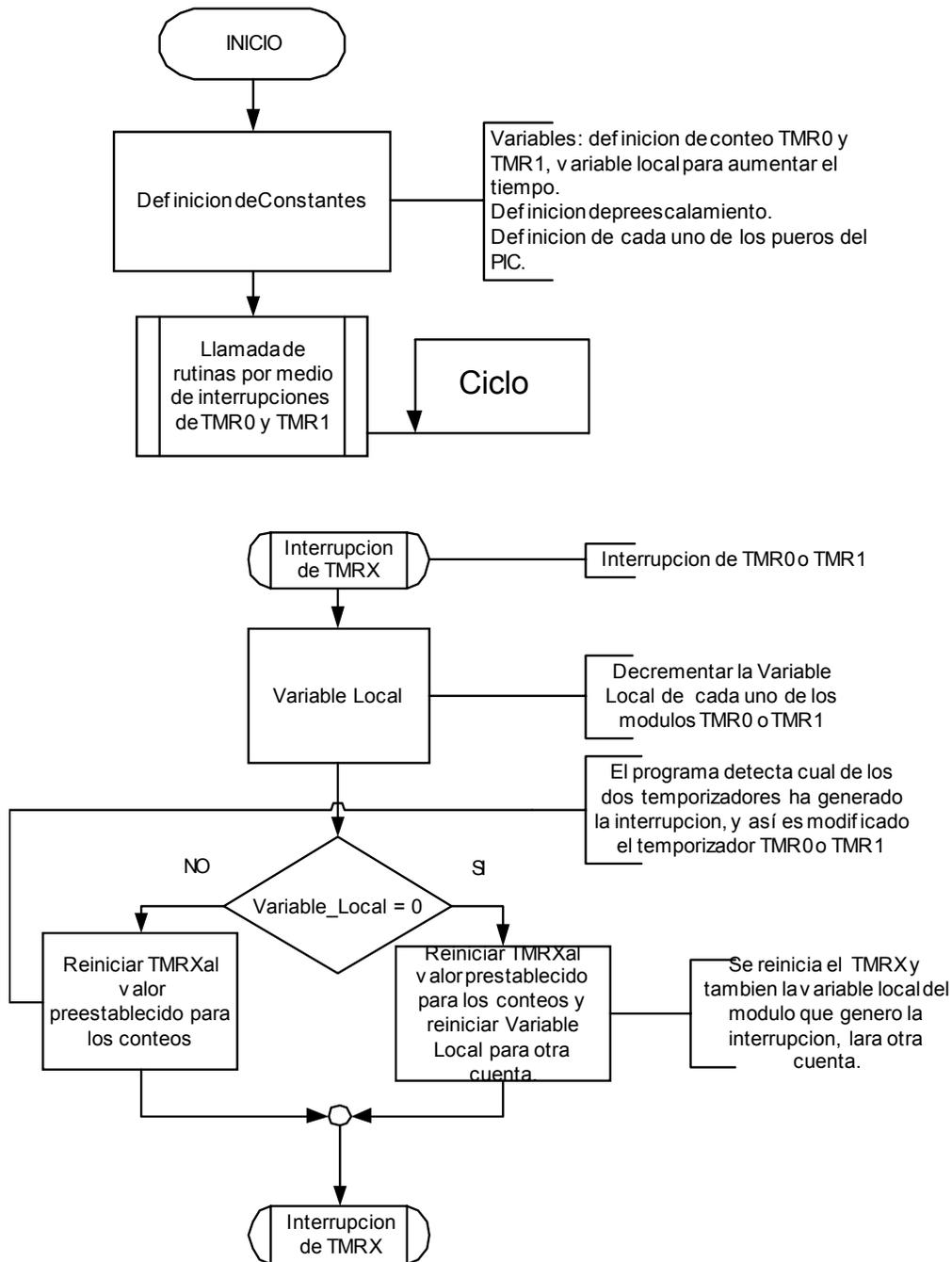


Figura 1.31 Diagrama a bloques de la generación de señales temporizadas

1.3.3.3 MÓDULO DE ENTRADAS ANALÓGICAS

A través de cualquier entrada analógica se aplica la señal analógica a un condensador de captura y retención (sample and hold), ésta es la señal que el convertidor procesa; por lo tanto, el hardware de aproximación sucesiva da como resultado una palabra de 10 bits.

El convertidor A/D puede seleccionar como tensión de referencia la interna $V_{DD}(+5V)$ o bien una externa que se introduzca entre $RA3/AN3/V_{REF+}$ y $RA3/AN3/V_{REF-}$. Cuando se selecciona la tensión de referencia externa, hay que tener en cuenta que existen algunas limitaciones; el máximo voltaje aplicado a la patita V_{REF+} ($RA3/AN3$) será $V_{DD}+0,3V$ y el mínimo $V_{DD}-2,5V$. En cuanto a la tensión V_{REF-} ($RA2/AN2$) la mínima tensión será $V_{SS}-0,3V$ y la máxima $V_{REF+} - 2V$, así por ejemplo, si la tensión de alimentación es de 5V, la tensión en $RA3/AN3$ no podrá exceder de 5V. Por lo que el máximo voltaje en V_{REF-} será de 3V. Siempre se ha de cumplir que $V_{REF+} - V_{REF-} \leq 2V$.

El convertidor A/D tiene como característica especial el ser capaz de seguir trabajando mientras el dispositivo esté en el modo SEP o de bajo consumo de potencia, para ello el oscilador interno RC debe conectarse al convertidor.

El módulo A/D cuenta con cuatro registros de configuración y obtención de resultados, estos son:

- ADRESH : Parte alta del resultado de la conversión.
- ADRESL : Parte baja del resultado de la conversión.
- ADCON0: Registro de Control 0.
- ADCON1, Registro de Control 1.

Los pines del PORTA están multiplexados con el convertidor analógico digital, y pueden configurarse de la siguiente manera:

- a) Entradas analógicas.
- b) Entrada de tensión de referencia.
- c) Entradas o Salidas digitales.

En la siguiente figura se muestra el diagrama esquemático del A/D:

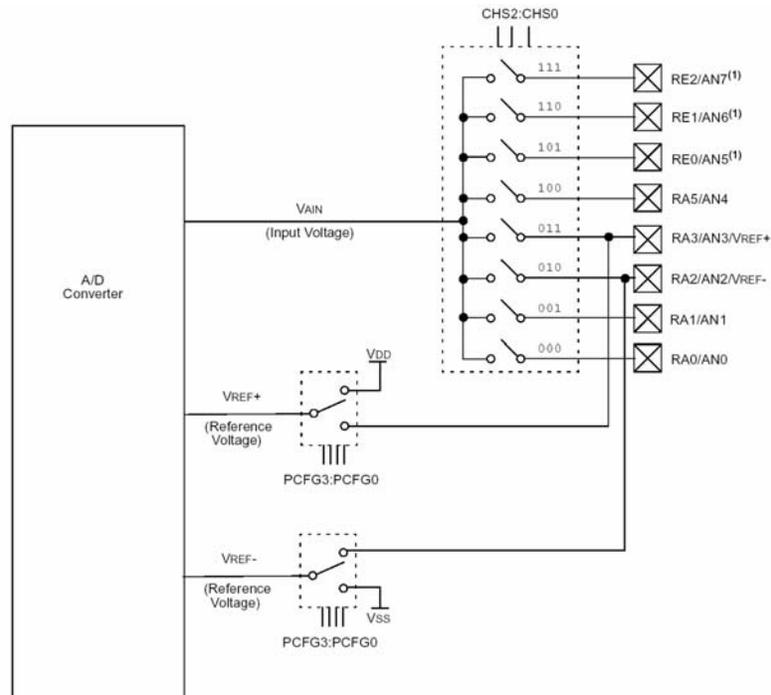


Figura 1.32 Diagrama esquemático del ADC PIC16F877.

Requisitos de conversión: para que el convertidor A/D pueda tener la exactitud especificada, debe permitirse la carga del condensador (C_{HOLD}) que mantiene la entrada a un nivel de tensión. El modelo de una entrada analógica se muestra en la figura. 1.33

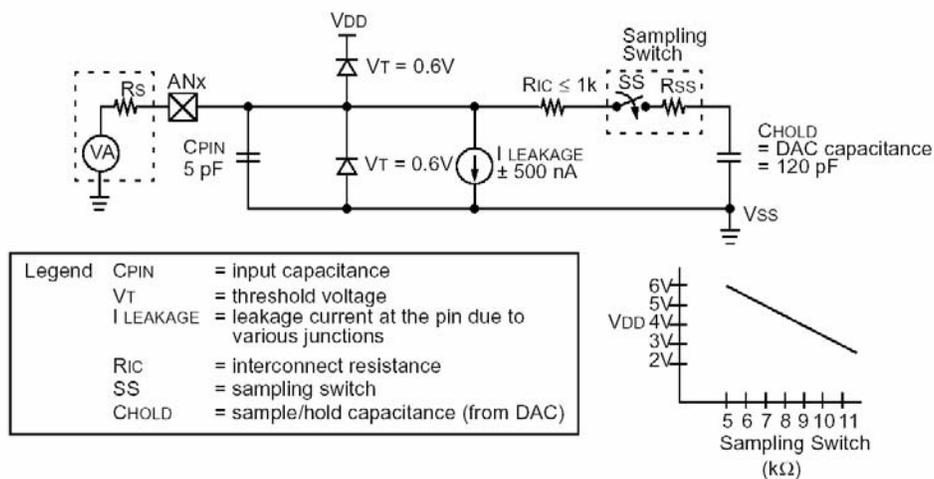


Figura 1.33 Modelo de una entrada analógica

De la impedancia de la fuente (R_S) se toma una muestra a través del interruptor (R_{SS}) la impedancia afecta directamente al tiempo de carga del condensador C_{HOLD} . La

impedancia recomendada para las fuentes analógicas es de 10 kΩ. Cuando la impedancia disminuye también disminuye el tiempo de adquisición. Antes de realizar, la conversión debe seleccionarse el canal de la entrada analógica.

Para calcular el tiempo mínimo de adquisición, se utiliza la ecuación 1.3, que asume un error de ½ LSB para 1024 pasos del convertidor A/D.

$$\begin{aligned}
 T_{ACQ} &= \text{Amplifier Settling Time} + \\
 &\quad \text{Hold Capacitor Charging Time} + \\
 &\quad \text{Temperature Coefficient} \\
 &= T_{AMP} + T_C + T_{COFF} \\
 &= 2\mu\text{s} + T_C + [(\text{Temperature} - 25^\circ\text{C})(0.05\mu\text{s}/^\circ\text{C})] \\
 T_C &= \text{CHOLD} (R_{IC} + R_{SS} + R_S) \ln(1/2047) \\
 &= -120\text{pF} (1\text{k}\Omega + 7\text{k}\Omega + 10\text{k}\Omega) \ln(0.0004885) \\
 &= 16.47\mu\text{s} \\
 T_{ACQ} &= 2\mu\text{s} + 16.47\mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05\mu\text{s}/^\circ\text{C})] \\
 &= 19.72\mu\text{s}
 \end{aligned}$$

Ecuación 1.3 Tiempo mínimo de conversión.

Para seleccionar el reloj del módulo ADC, se define el siguiente el T_{AD} (al tiempo de conversión por bit). El convertidor A/D requiere un mínimo de 12 T_{AD} para la conversión de los 10 bits, La señal de reloj para la conversión A/D se selecciona por software mediante los bits ADCS1:ADCS0.

Para realizar conversiones correctas, el tiempo mínimo de T_{AD} de 1,6 uS.

En el uso del ADC como utilidad en este prototipo, no se necesitan altas velocidades de conversión, por lo que se selecciona un XTAL de 4MHz, y un tiempo de $T_{AD} = 32 T_{OSC} = 8\mu\text{Seg}$, por consiguiente se necesitan 12 T_{AD} para una conversión completa, teniendo un tiempo por muestra de $T_{\text{Sample}} = 96\mu\text{Seg}$. Según el teorema de Nyquist²² se deduce que $F_{\text{Sample}} = 10.416\text{KHz}$; esto indica que se pueden muestrear señales menores a 5KHz.

²² Teorema de Nyquist: “La frecuencia mínima para muestrear una señal, debe ser al menos el doble de esta”

DESCRIPCIÓN DEL MANEJADOR DEL MÓDULO ADC

El manejador de ADC lee una muestra de la señal de entrada, la compara con un valor preestablecido(definido por el usuario), para activar o desactivar equipos(motores, luces, alarmas, etc.); a continuación se definen las características de trabajo:

- Posee una variable, la cual puede ser programada con un valor de voltaje(0.2V a 5.0V), dividiendo así los rangos ALTO y BAJO, ésta variable es Valor_X(Ver figura 1.34). El controlador, está definido para que se pueda seleccionar una salida del PLC (OUTL), en la cual estará presente un nivel de voltaje(5V o 0V) según sea el caso.



Figura 1.34 Rangos de la señal para el puerto de entrada analógica

- Para proteger el programa contra ruido eléctrico, se usa el método llamado Histéresis; que consiste en establecer voltajes máximos y mínimos, en los cuales es aceptable el funcionamiento. El voltaje de histéresis es 200mV. En la figura 1.35, se muestra el comportamiento de una señal y la Histéresis:

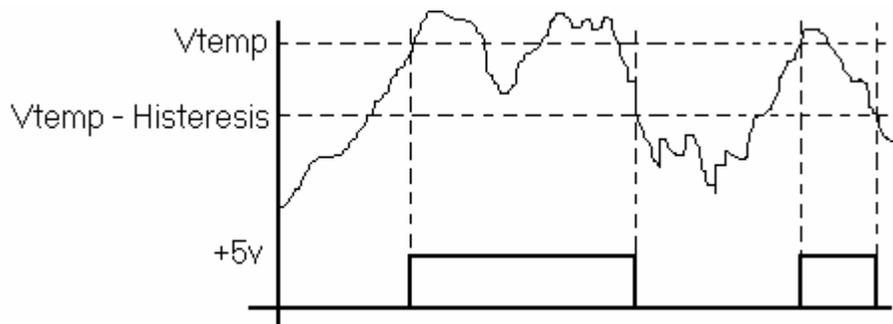


Figura 1.35 Histéresis

A continuación se muestra un circuito propuesto para trabajar con las entradas analógicas del PLC:

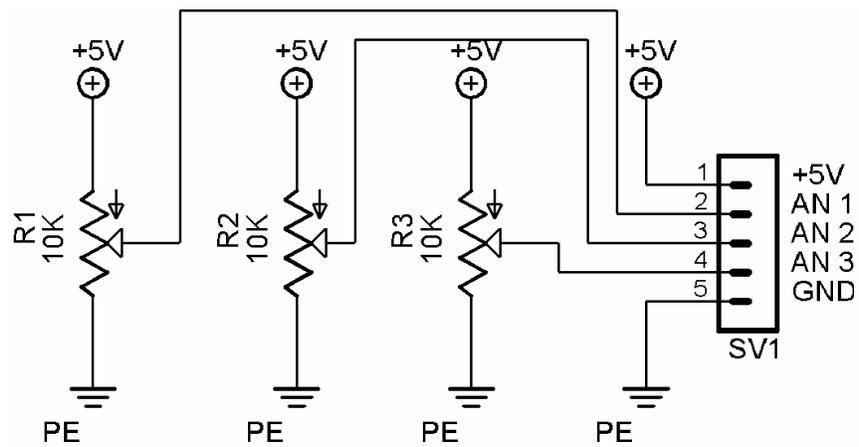


Figura 1.36 Circuito para probar las entradas analógicas del PLC.

A continuación, se presenta el diagrama de flujo del manejador ADC del prototipo:

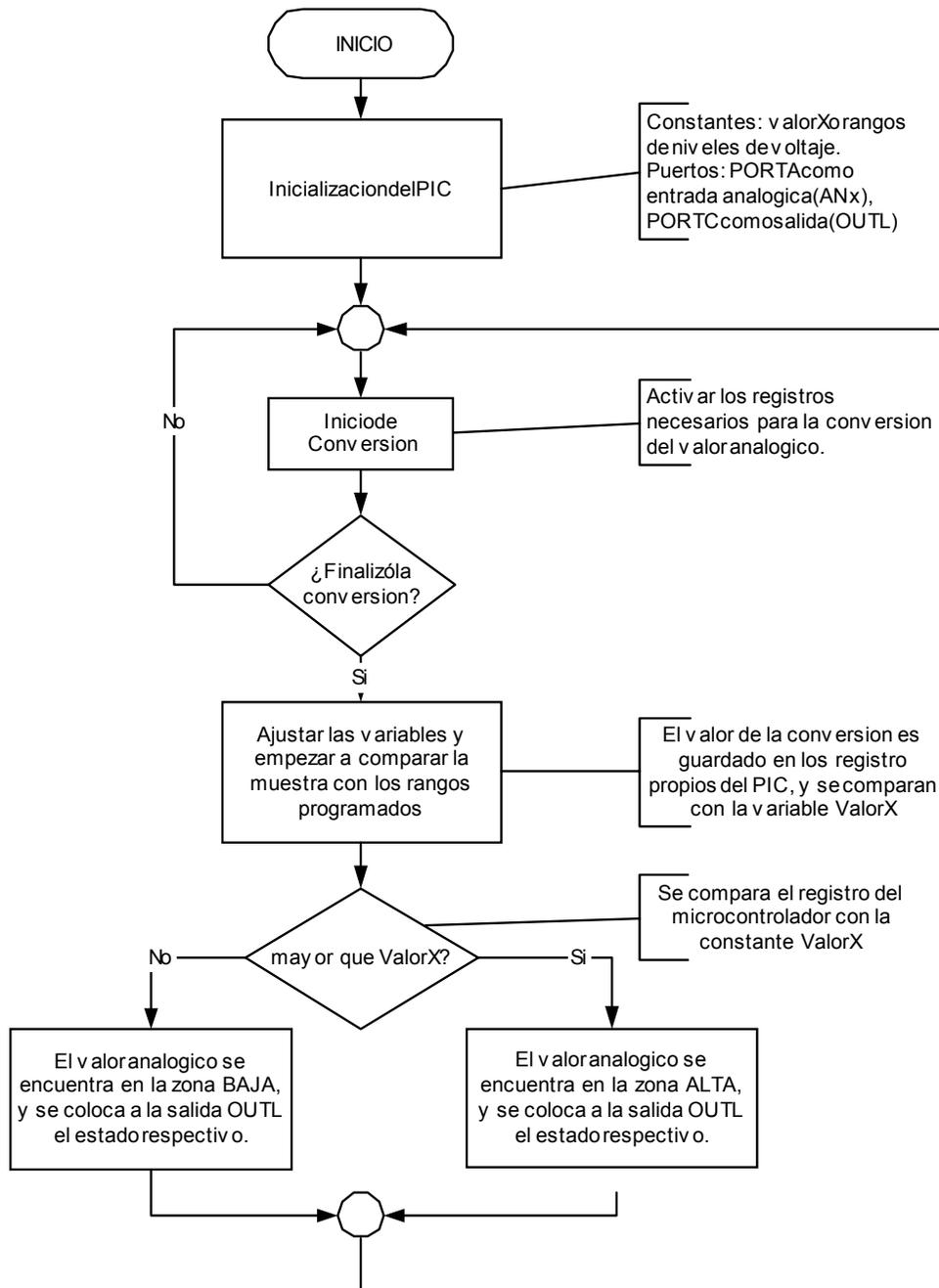


Figura 1.37 Diagrama de flujo del manejador ADC

1.3.3.4 MÓDULO DE ENTRADAS Y SALIDAS CMOS

El manejador para probar esta característica, se basa en la lectura de las entradas de alto voltaje (RB0, RB1, RB2, RB3) y la escritura en el puerto de salida de alto voltaje (RC4, RC5, RC6, RC7). A continuación se muestra el circuito para probar este módulo:

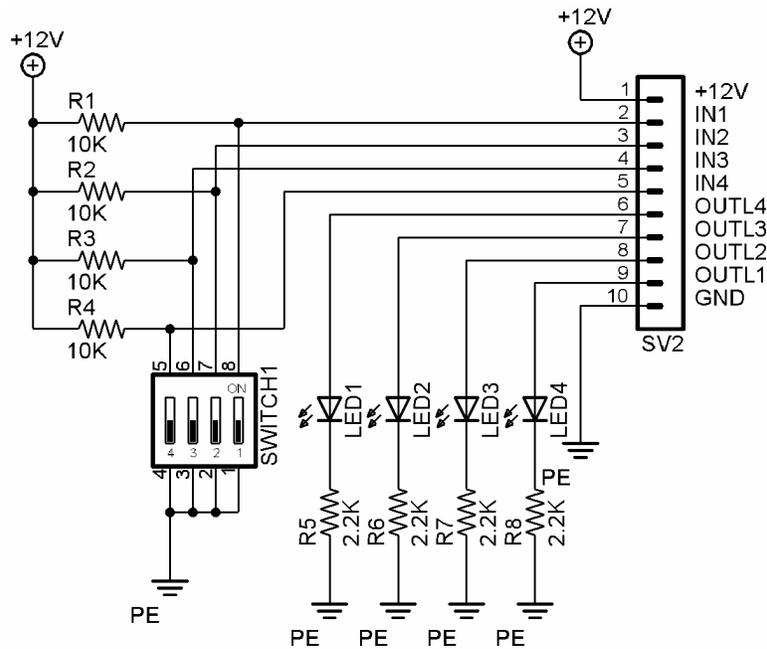


Figura 1.38 Circuito de prueba para E/S CMOS

Lo que se muestra en el siguiente flujograma:

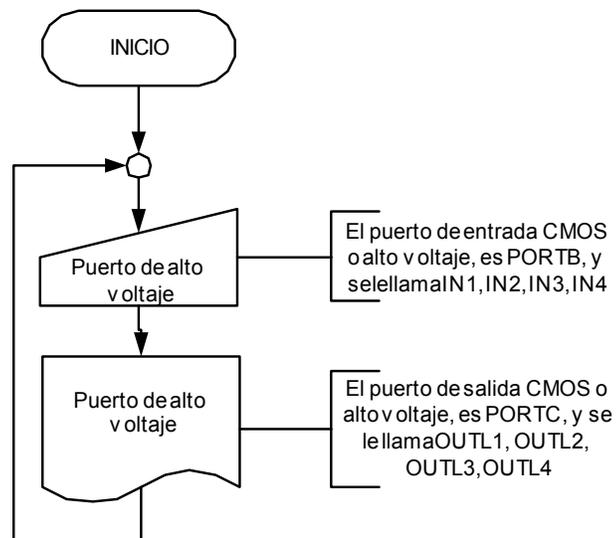


Figura 1.39 Diagrama de flujo del manejador de entradas y salidas

1.3.3.5 PUERTO DE SALIDA DE POTENCIA (3AMP MÁXIMO)

Las salidas de potencia, son controladas por el uC por medio de las del Puerto C (RC0, RC1, RC2, RC3); éstas salidas son conectadas a un PUENTE DARLINGTON que es el STA401 descrito en la sección 1.1.4.3, las salidas son llamadas OUTH1, OUTH2, OUTH3, OUTH4.

Para probar ésta salida (Figura 1.40), el manejador controla un motor paso a paso unipolar, el cual se hace girar en un sentido (el sentido de giro se especifica en IN2); la velocidad de giro es determinada por un tren de pulsos, que están presentes en la entrada de CMOS IN1.

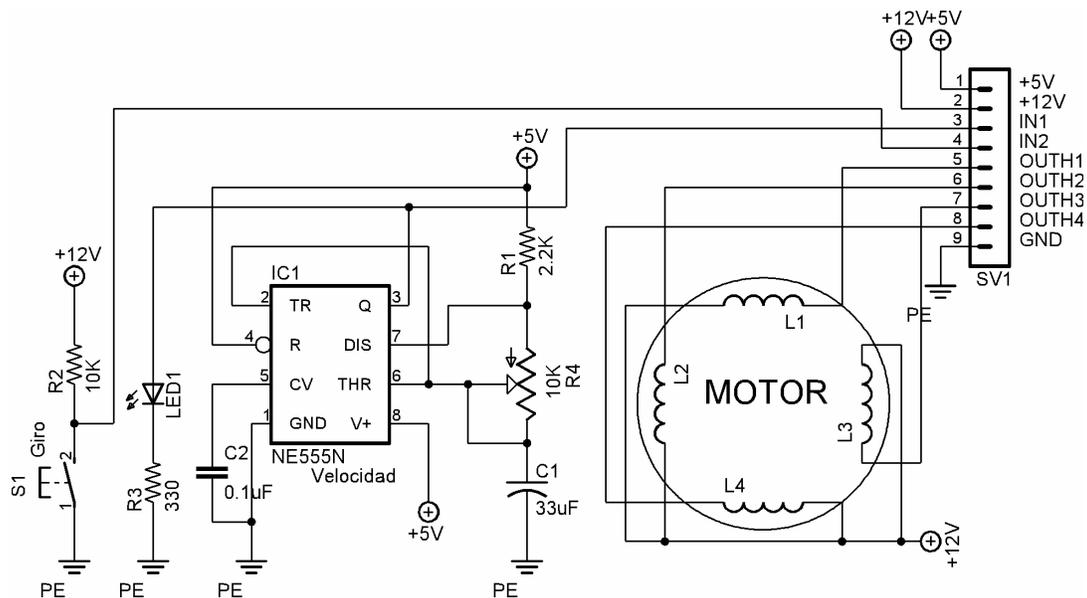


Figura 1.40 Circuito de conexión de un Motor Paso a Paso Unipolar

La siguiente figura se muestra el diagrama de flujo para el control de un motor paso a paso unipolar:

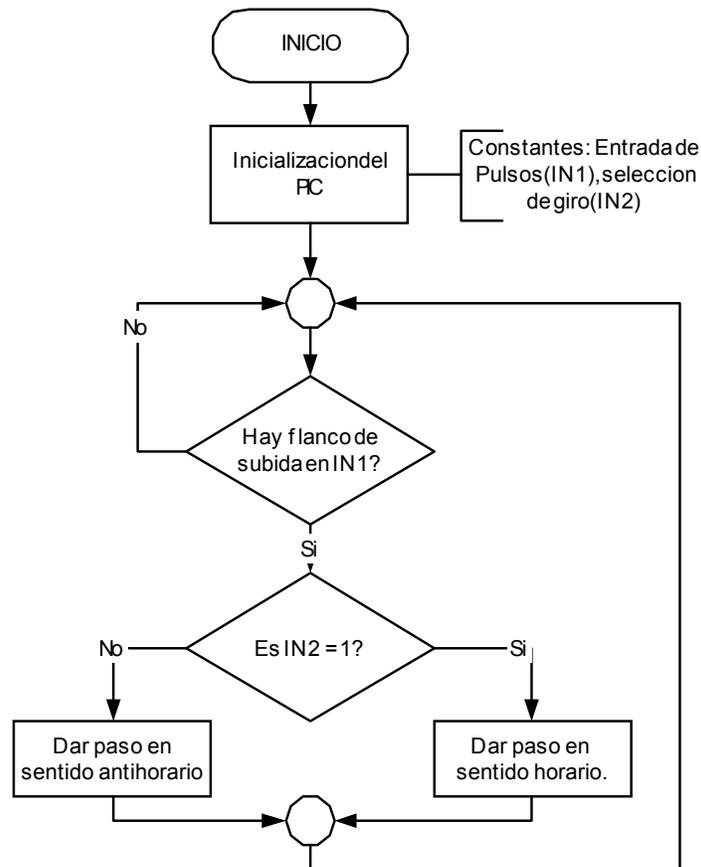


Figura 1.41 Diagrama de flujo del control de un motor paso a paso unipolar.

1.3.3.6 FUNCIONES BÁSICAS: AND, OR, NOT, NAND, NOR, XOR

La implementación de funciones lógicas básicas, se hace por medio de la programación de una tabla de verdad, el cual si se construyera el circuito equivalente, sería muy complejo, por lo que teniendo la tabla, se puede obtener en valor equivalente de salida teniendo un valor a la entrada.

Ejemplo 1.5 Implementación de un tabla de verdad.

a) TABLA DE VERDAD

Entradas de alto voltaje					Salidas de alto voltaje			
ITEM	IN4	IN3	IN2	IN1	OUTL4	OUTL3	OUTL2	OUTL1
0	0	0	0	0	1	1	0	0
1	0	0	0	1	0	1	0	1
2	0	0	1	0	0	1	0	0
3	0	0	1	1	0	1	1	0
4	0	1	0	0	1	0	0	0
5	0	1	0	1	0	0	1	1
6	0	1	1	0	0	1	0	0
7	0	1	1	1	0	0	1	0
8	1	0	0	0	1	0	0	0
9	1	0	0	1	0	0	1	1
10	1	0	1	0	0	1	0	0
11	1	0	1	1	0	0	1	0
12	1	1	0	0	1	1	0	0
13	1	1	0	1	0	1	1	1
14	1	1	1	0	0	1	1	0
15	1	1	1	1	0	1	1	0

Tabla 1.10 Tabla de verdad

b) RESOLUCIÓN DE LA TABLA PARA OBTENER LAS ECUACIONES.

Utilizando el método de mapa K, se obtienen las siguientes ecuaciones:

1) $OUTL4 = \overline{IN1 + IN2}$ Ecuación 1.4

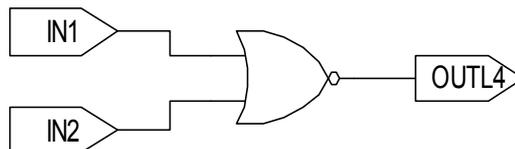


Figura 1.42 Diagrama Esquemático

2) $OUTL3 = \overline{IN4} + IN3 + (IN2)(\overline{IN1}) + (IN4)(IN3)$ Ecuación 1.5

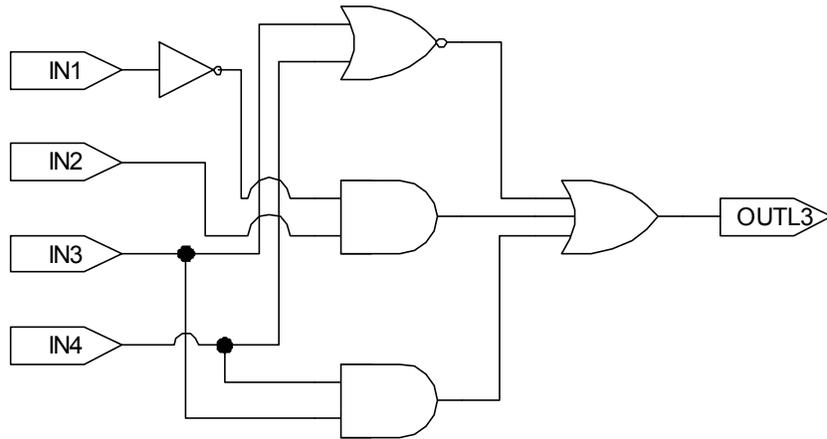


Figura 1.43 Diagrama Esquemático

3) $OUTL2 = (IN4)(IN3)(IN2) + (IN2)(IN1) + (IN3)(IN1) + (IN4)(IN1)$ Ecuación 1.6

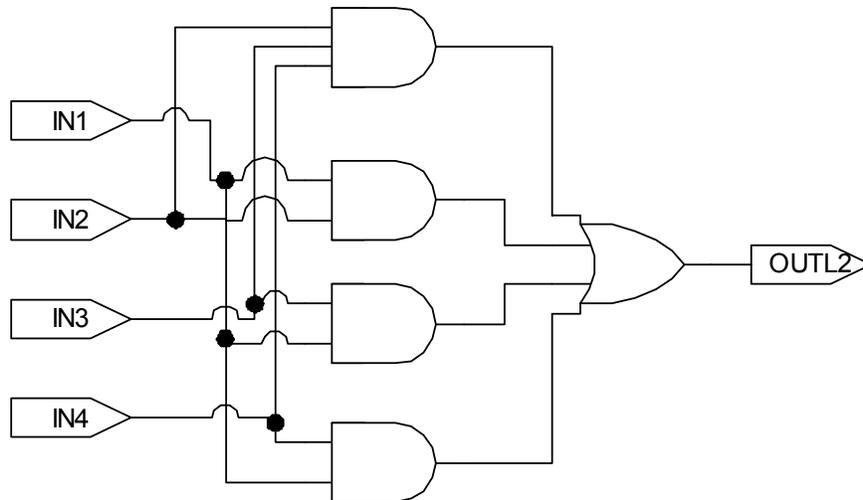


Figura 1.44 Diagrama Esquemático

4) $OUTL1 = (\overline{IN2})(IN1)$ Ecuación 1.7

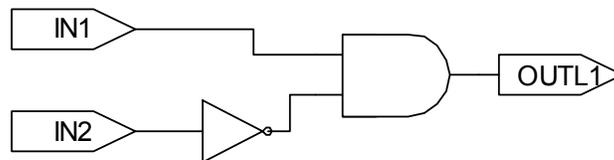


Figura 1.45 Diagrama Esquemático

- c) Con las Ecuaciones 1.6-1.7 y dibujar los diagramas equivalentes, se procede a la construcción física, es cuando se hace uso de un microcontrolador para que los sistemas con compuertas se reduzcan a un simple código de programa. En la siguiente figura se presenta el diagrama de flujo:

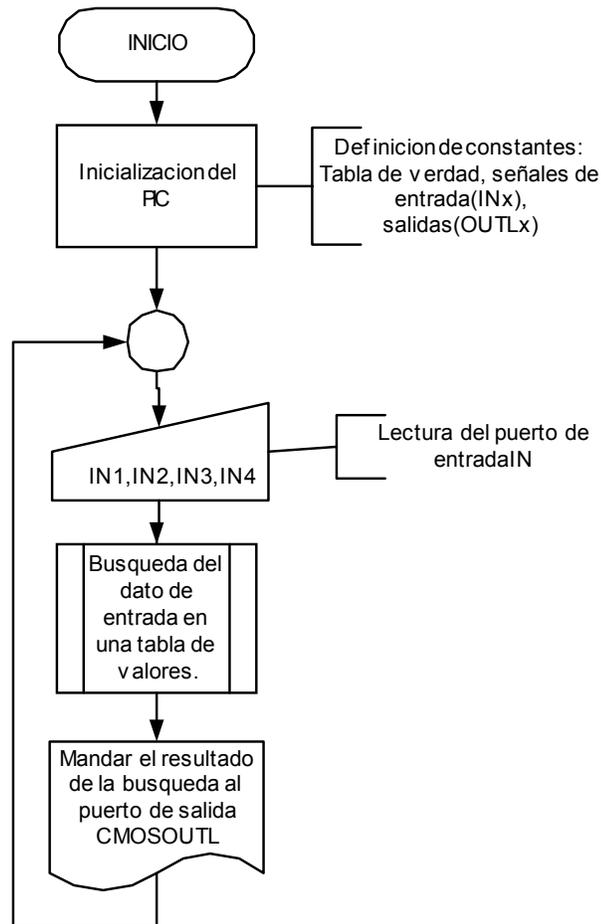


Figura 1.46 Diagrama de Flujo de la implementación de Funciones Básicas Lógicas.

NOTA: El circuito para realizar la prueba de la Tabla de verdad, se puede utilizar el de la Figura 1.38.

2. CAPITULO II

“DISEÑO Y CREACIÓN DEL SOFTWARE DE PROGRAMACIÓN DEL PROTOTIPO
DE PLC (CONTROL LÓGICO PROGRAMABLE) BASADO EN EL
MICROCONTROLADOR PIC16F877

2.1 DISEÑO DEL ALGORITMO PARA EL SOFTWARE DE CONFIGURACIÓN DEL PLC

2.1.1 FLUJOGRAMA GENERAL

El software de configuración del PLC, se basa en el siguiente algoritmo gráfico:

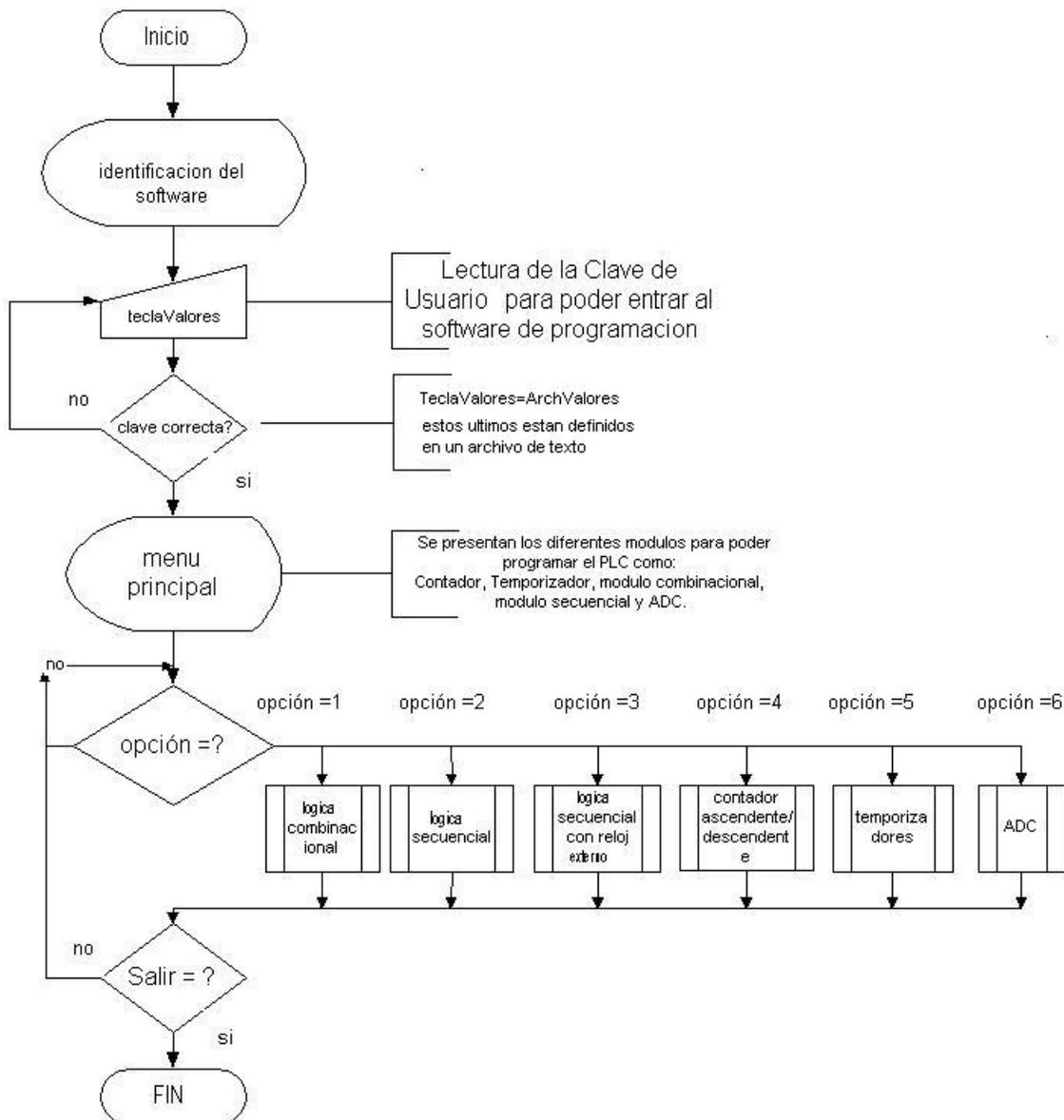


Figura 2.1: Algoritmo gráfico para el configurador del PLC.

2.1.2 FLUJOGRAMA DEL MÓDULO CONTADOR

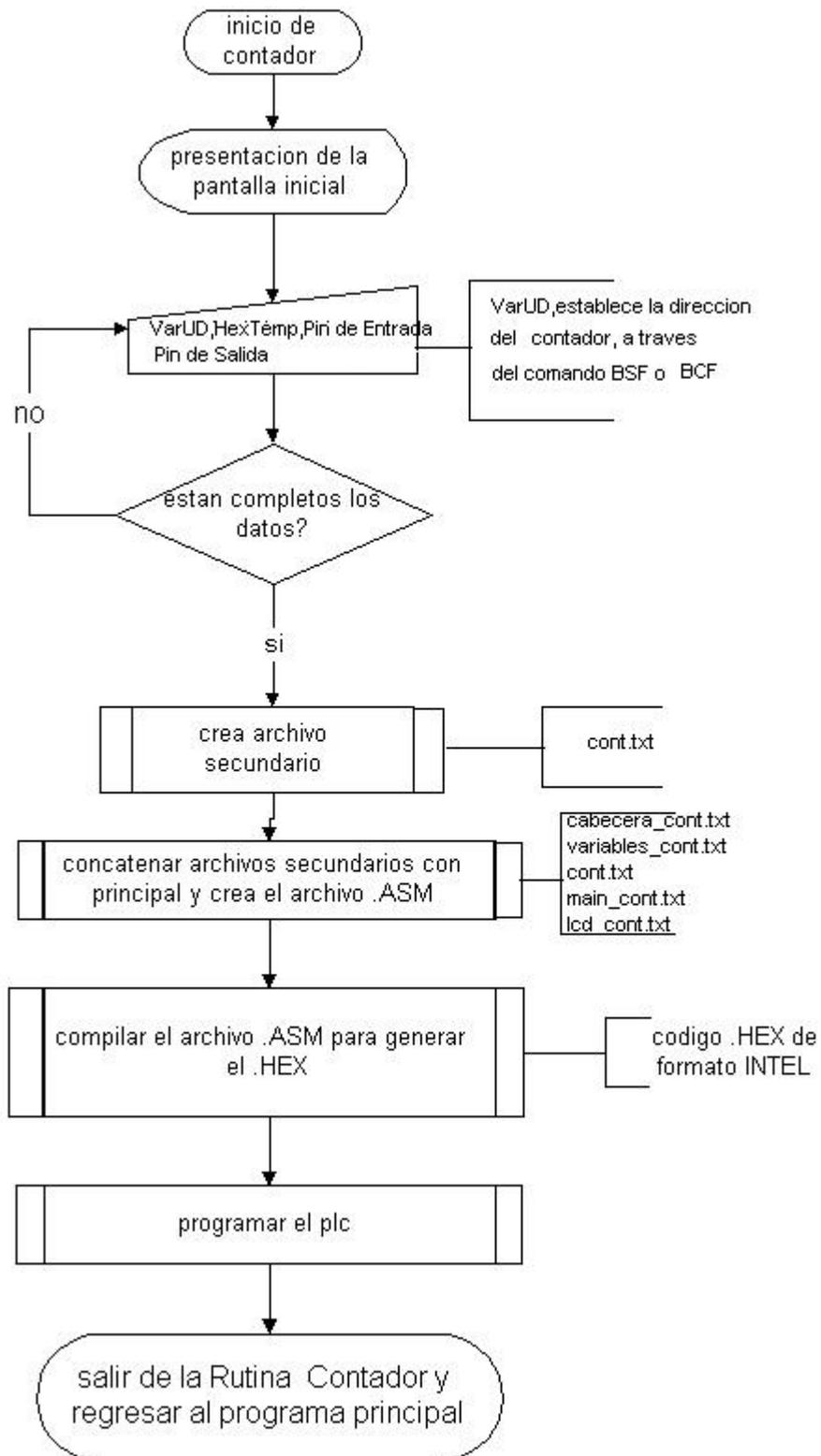


Figura 2.2: Rutina Contador.

2.1.3 FLUJOGRAMA DEL MÓDULO COMBINACIONAL

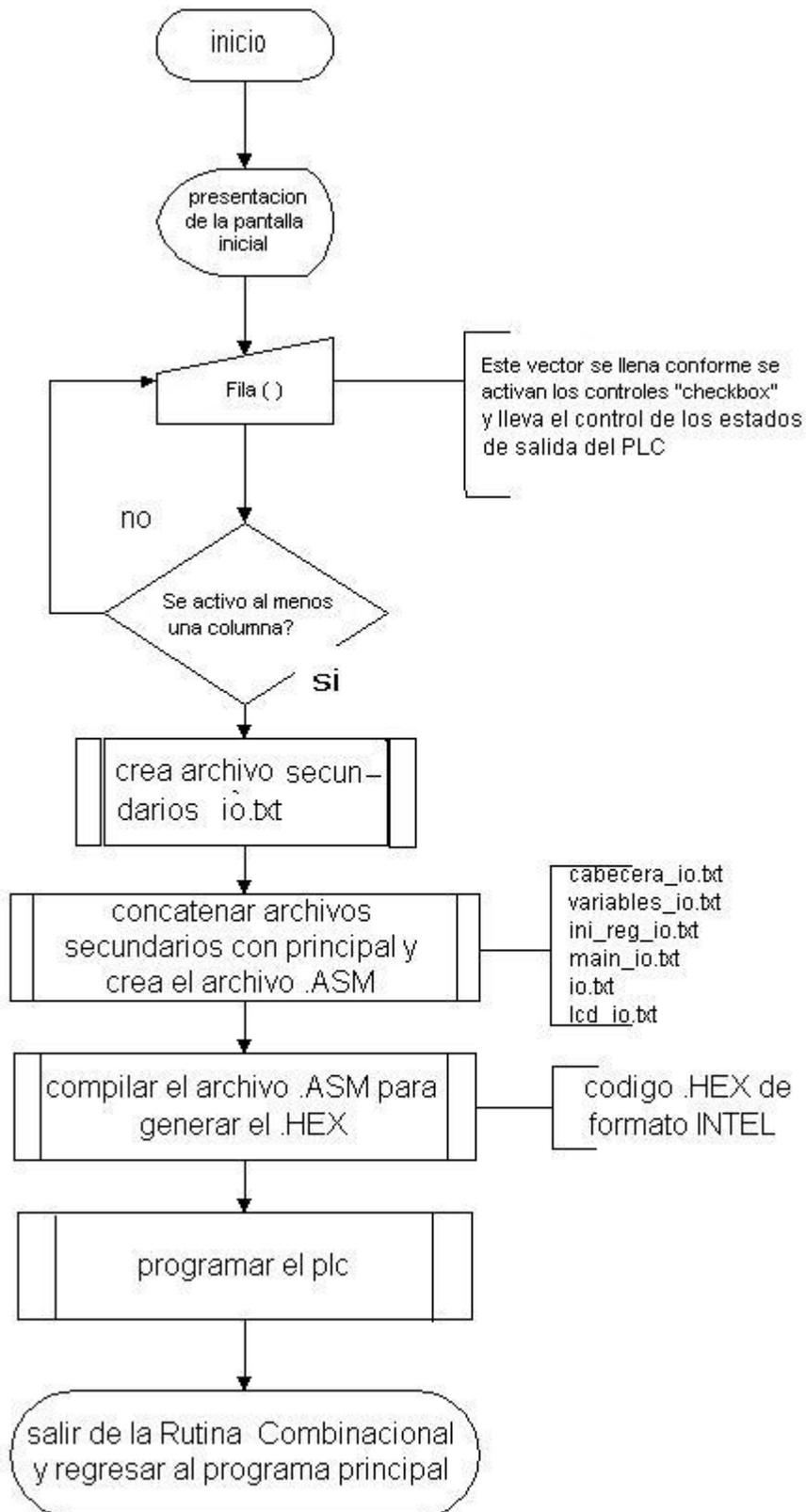


Figura 2.3: Rutina Módulo Combinacional (Tabla de Verdad)

2.1.4 FLUJOGRAMA DEL MÓDULO TEMPORIZACIÓN

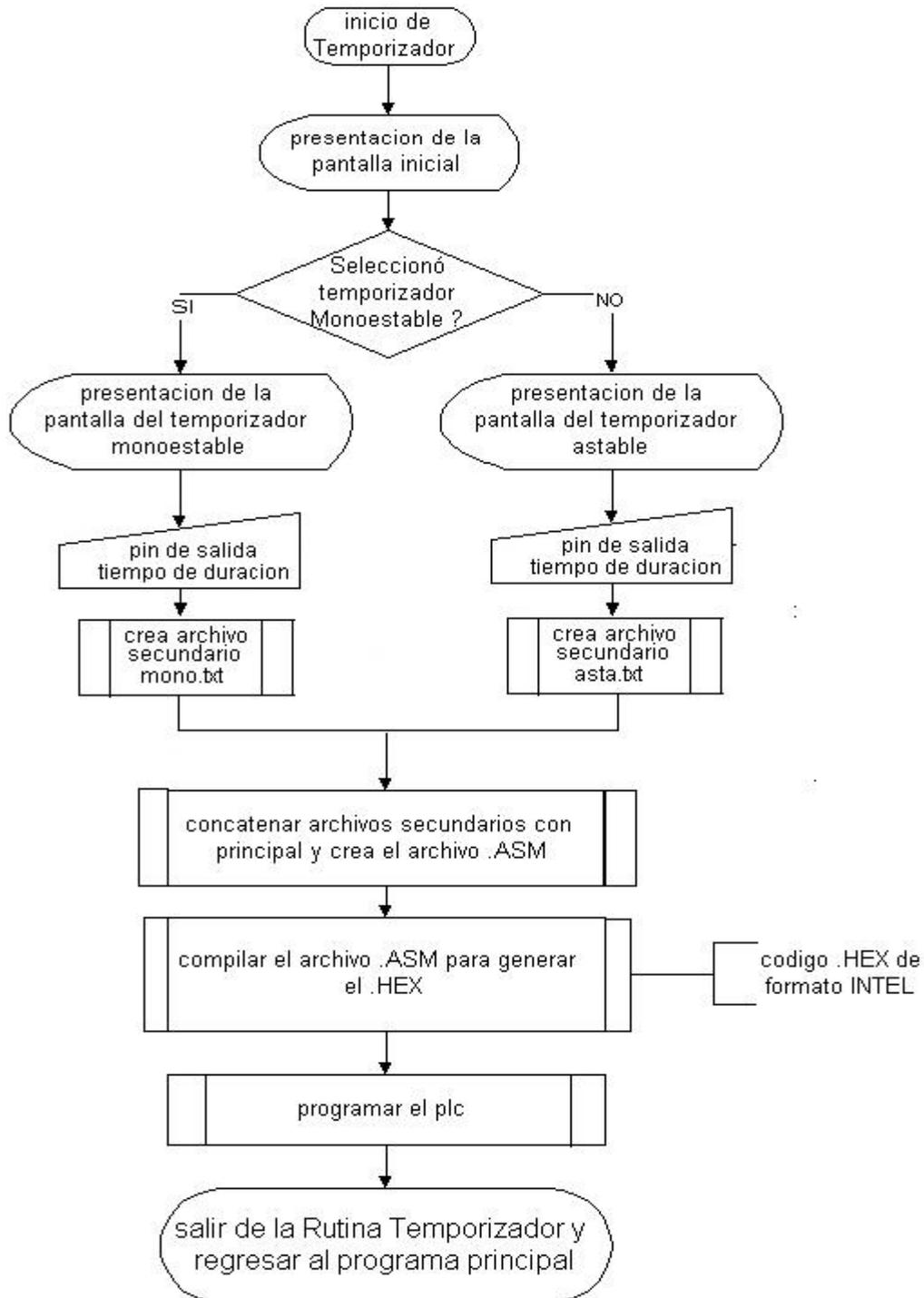


Figura 2.4: Módulo temporización

2.1.5 FLUJOGRAMA DEL MÓDULO CONVERTIDOR ANALÓGICO DIGITAL

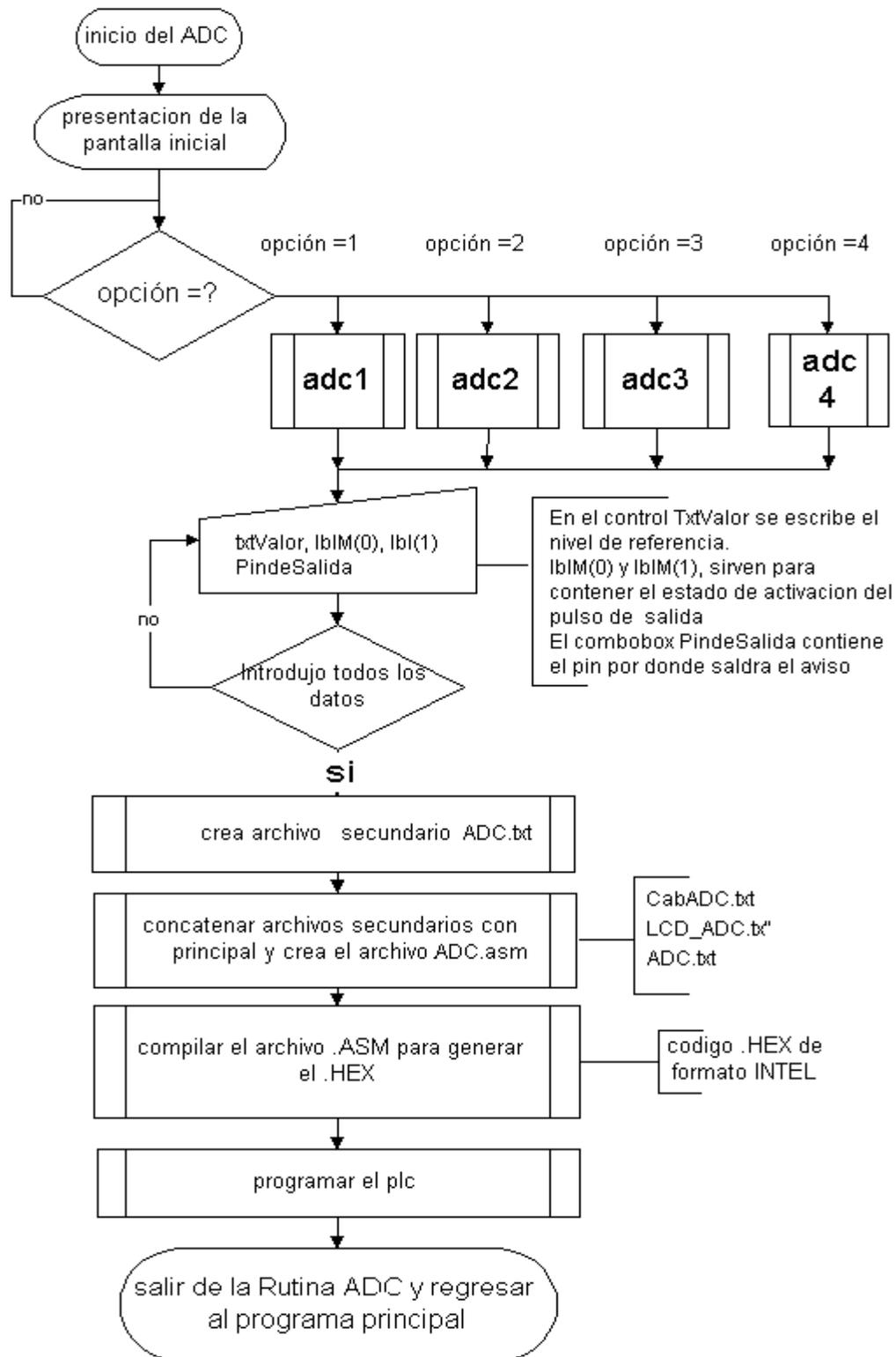


Figura 2.5: Módulo Configuración del ADC

2.1.6 FLUJOGRAMA DEL MÓDULO SECUENCIAL

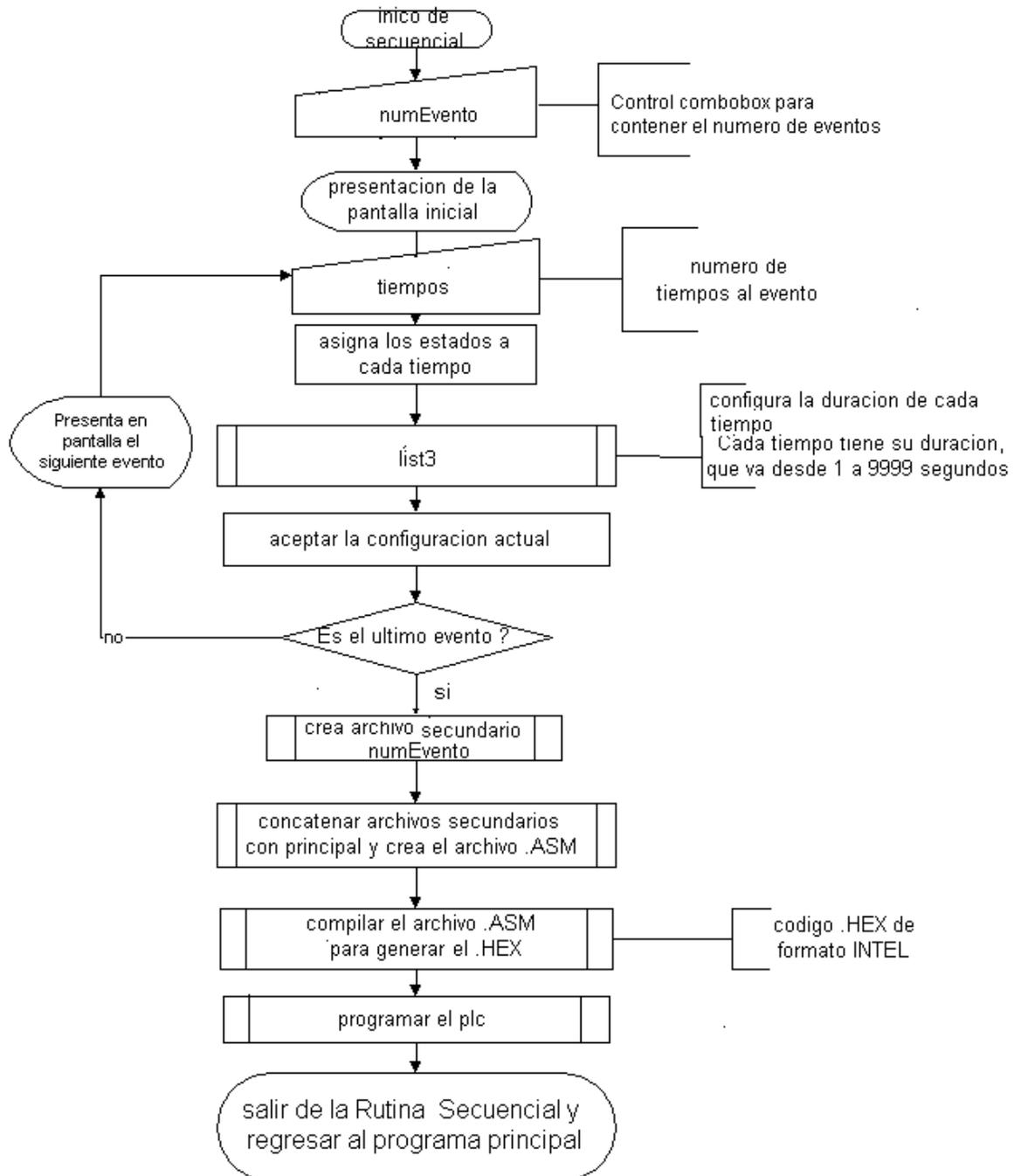


Figura 2.6: Módulo Secuencial

2.2 IMPLEMENTACIÓN DEL SOFTWARE DE CONFIGURACIÓN DEL PLC

2.2.1 MANUAL DEL USUARIO DEL OCRPLC²³.

El sistema ha sido desarrollado en lenguaje Visual Basic versión 6.0 de Microsoft, su lógica general es la de un sistema modular controlado principalmente por eventos que provienen del teclado o clic del ratón.

VENTANA PRINCIPAL



Figura 2.7: ventana principal de la aplicación OCRPLC.

Ésta se aparece al cargar la aplicación, al iniciarla aparece como en se muestra la figura 2.7, en la que se puede seleccionar a través de una barra cualquier función que será desarrollada por el PLC.

²³ OCRPLC, son iniciales que identifican a los alumnos que realizaron éste trabajo (Osmín Cartagena Renato PLC).

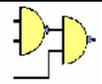
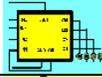
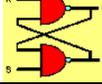
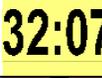
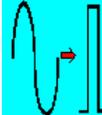
Icono	Descripción
	Lógica Combinacional
	Lógica Secuencial de eventos automáticos con tiempos variables
	Lógica Secuencial con control de reloj externo
	Contador ascendente – descendente
	Temporizadores monoestables y astables
	Convertidor Analógico Digital

Tabla 2.1: Diferentes iconos de la barra de herramientas para activar las funciones del PLC

Como muestra la Tabla 2.1, al hacer clic sobre el icono “compuertas” se podrá configurar el módulo Combinacional, además se establecen los estados de las salidas de alta corriente del PLC (Q4, Q3, Q2, Q1).

Cuando se hace un clic en el ícono “bloque de función”, se carga el evento secuencial, en este se asignan datos tales como: número de eventos, número de tiempos a cada evento, duración y estado de cada tiempo, pero cuando se hace un clic en el ícono “flip flop RS”, se carga el evento secuencial con control de tiempo externo a través una señal de reloj, en este se podrá asignar datos tales como: número de eventos, número de tiempos a cada evento, duración y estado de cada tiempo

Al hacer clic sobre el ícono “32:07”, (cronómetro), sirve para asignar las características del contador. En el ícono “reloj” para el temporizador ya sea del tipo astable o monoestable y por último en el ícono “señales” para establecer las características y parámetros del conversor de analógico a digital (ADC).

2.2.1.1 LÓGICA COMBINACIONAL

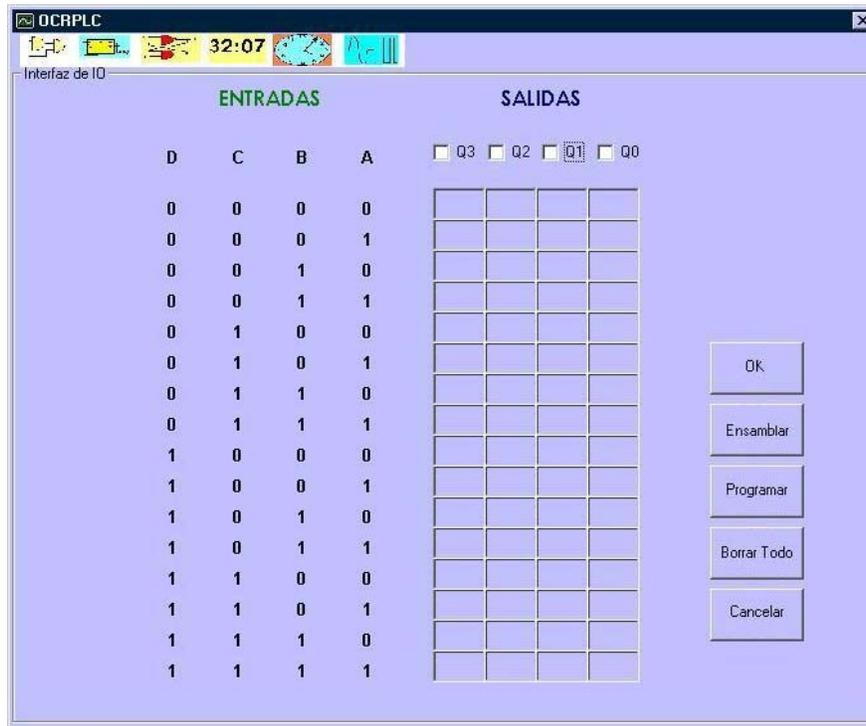


Figura 2.8: ventana de asignación de las salidas para la lógica.

Este módulo configura cualquier circuito combinacional con cuatro entradas y cuatro salidas, las entradas y salidas son CMOS a 12 V. Las señales entraran por IN1, IN2, IN3, IN4 y el resultado saldrá por cualquiera de las cuatro líneas: OUT4, OUT3, OUT2, OUT1, programando así cualquier función: AND, OR, XOR, o combinaciones entre ellas. En la figura 2.9 se muestra las posibles conexiones.

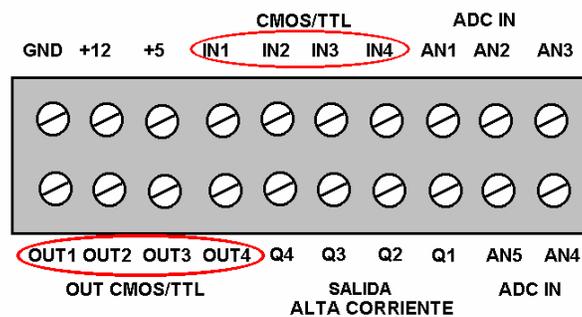


Figura 2.9: Entradas y salidas CMOS en la bornera del PLC

2.2.1.2 LÓGICA SECUENCIAL.

Al iniciar este módulo, el sistema consulta si desea cargar los datos de la última sesión como se ve en la figura 2.10, con el objeto de configurar al PLC con éstos, pensando en que solo se va a reestablecer la configuración perdida del PLC, previamente configurado, quedando como opción solamente compilar y programar al PLC., todo lo anterior se cumple cuando al hacer clic sobre la opción “SI”.

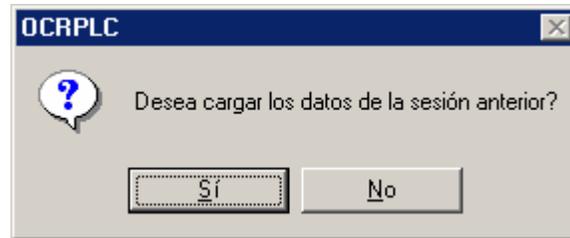


Figura 2.10: Muestra la opción de cargar los datos de la última sesión.

Al escoger la opción NO, inicia la presentación de una serie de ventanas que sirven para capturar los datos de una nueva configuración, como se muestra en las figuras 2.11 a y b.

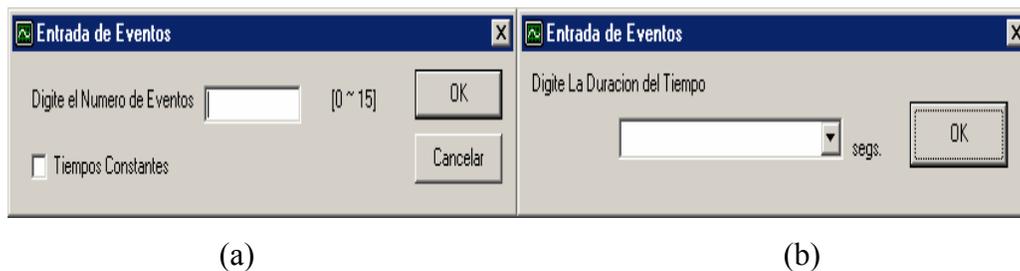


Figura 2.11

a) Ventana de opción para digitar el número de eventos y b) la duración de cada tiempo.

Luego de esto aparece la ventana que se muestra en la figura 2.12 que permite configurar los estados de salida (altos o bajos) en **Q4,Q3,Q2,Q1**, que son las salidas de alta corriente, en cada uno de los eventos seleccionados, con un máximo de 16 eventos, cada evento con un máximo de 16 tiempos como se muestra en la figura 2.12, tales estados dependen de las señales en las patas de entrada **IN4, IN3, IN2, IN1**.

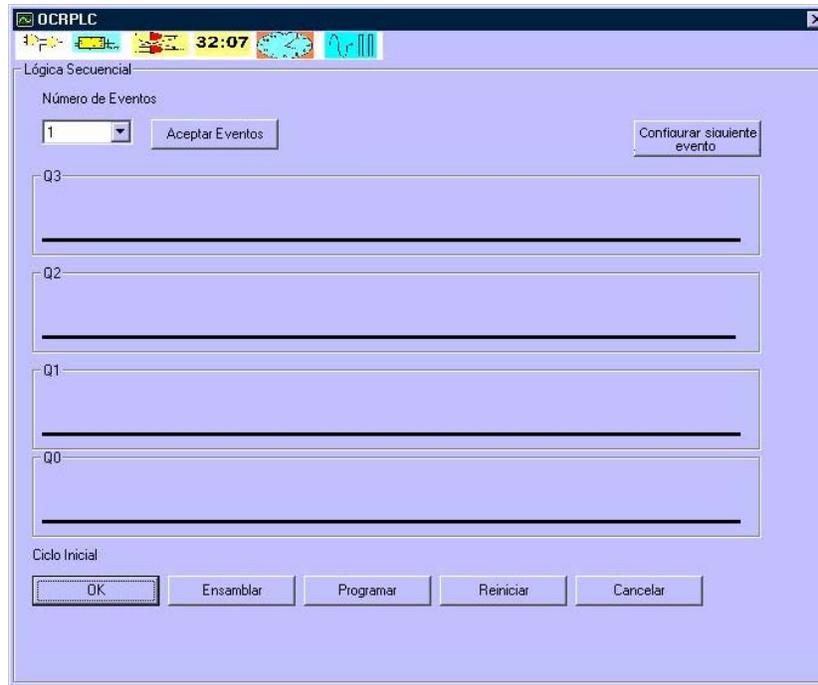
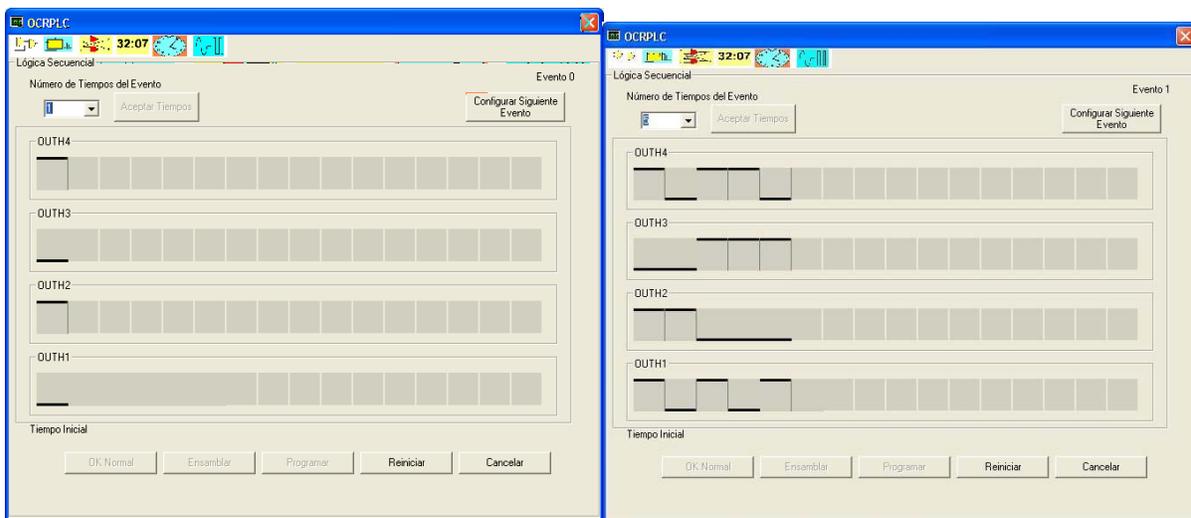


Figura 2.12: Ventana principal para la configuración de la lógica secuencial

Para asignar los estados de salida (Q4~Q1), primero se tendrán que completar los siguientes campos: número de tiempos, aceptar tiempos, si no se hace, aparecerá un mensaje recordándolo. Luego tendrá que establecer la lógica secuencial que se desee en la salida, activando con el clic del ratón cada intervalo de cada salida, en la figura 2.13 se muestra el ejemplo.



(a)

(b)

Figura 2.13 a) configuración para un tiempo y b) configuración para 5 tiempos

Al terminar de configurar los eventos seleccionados, el sistema avisa y se activa el botón OK, haciendo clic en él, se podrá compilar el archivo .ASM y generar el .HEX, luego programar el PLC.

En este caso (hardware) las señales se conectarán según se muestra en la figura 2.14, la señal de control de evento se introduce en IN4, IN3, IN2, IN1 y la secuencia de salida del evento seleccionado se obtendrá en los pines de alta corriente Q4, Q3, Q2 y Q1. Además en OUT2 y OUT3, se obtiene la función OR y AND de las salidas Q4~Q1, respectivamente, en OUT4 se obtiene un pulso de nivel alto que indica que el nivel de voltaje en la entrada AN1 ha sobrepasado el umbral²⁴, señal que indica un fallo general.

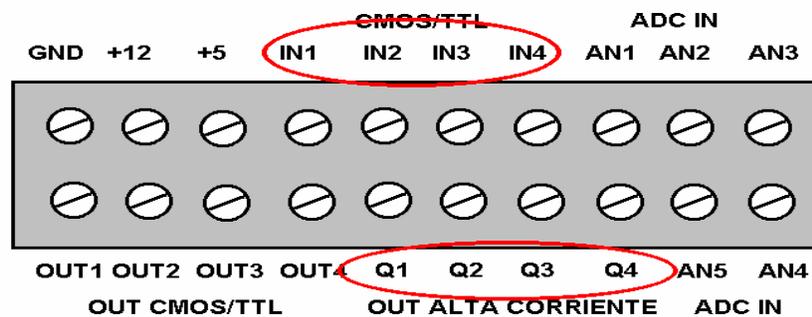


Figura 2.14 Conexiones posibles para la aplicación de lógica secuencial.

²⁴ El umbral de voltaje que activa la salida OUT4 con un pulso en alto está programado en 2.5 V.

2.2.1.3 LÓGICA SECUENCIAL CON RELOJ EXTERNO

Éste módulo se inicia cuando se hace clic en el icono “circuito integrado con rayo”, apareciendo la ventana que muestra la figura 2.15.



Figura 2.15: Ventana para introducir el número de eventos, con un máximo de 16, numerados 0~15

Al digitar el número de eventos y hacer clic en el botón OK, aparece la ventana que permite configurar los estados de salida (altos o bajos) en **Q4,Q3,Q2,Q1**, que son las salidas de alta corriente, en cada uno de los eventos seleccionados, tales estados dependen de las señales en las patas de entrada **IN4, IN3, IN2, IN1**.

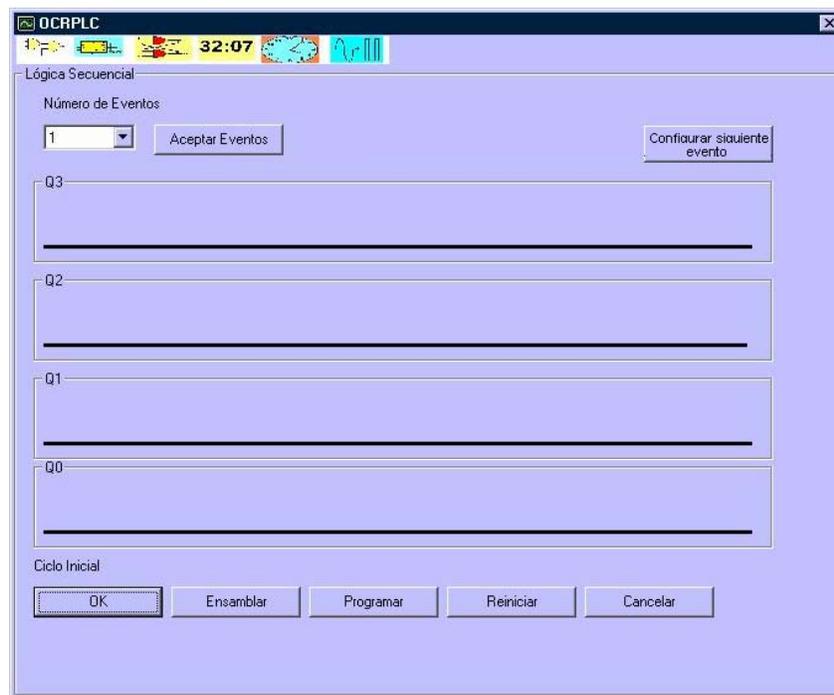
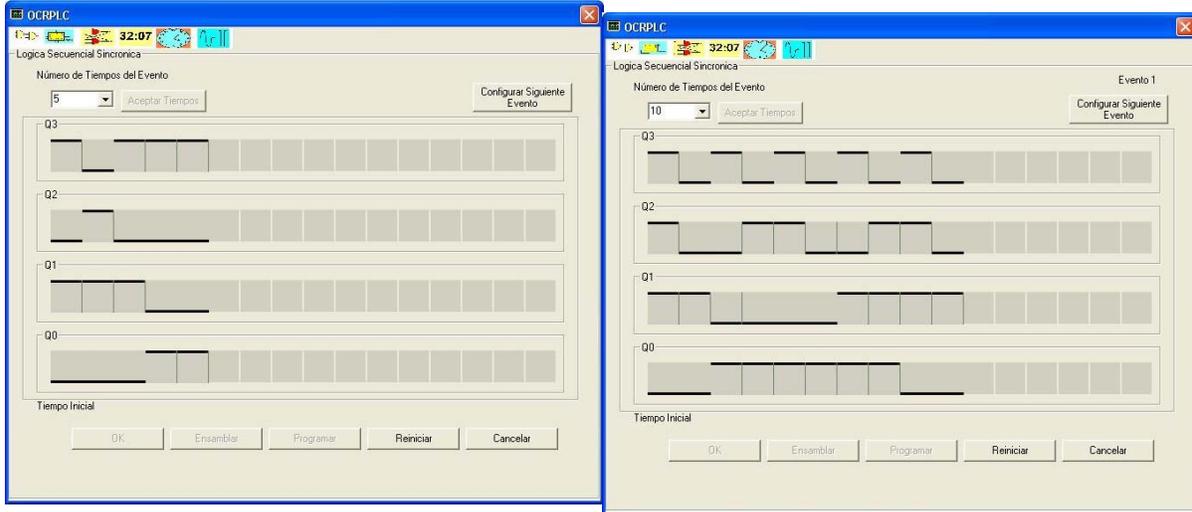


Figura 2.16: ventana principal para la configuración de la lógica secuencial con reloj externo

Para asignar los estados de salida (Q4~Q1), primero se tendrán que completar los siguientes campos: número de tiempos, aceptar tiempos, si no se hace, aparecerá un mensaje recordándolo.

Luego tendrá que establecer la lógica secuencial que se desee en la salida, activando con el ratón cada intervalo de cada salida como se ejemplifica en la figura 2.17 a y b.



(a)

(b)

Figura 2.17: a) configuración para cinco tiempos y b) configuración para diez tiempos

Al terminar de configurar los eventos seleccionados, el sistema envía un mensaje de aviso y se activa el botón OK, haciendo clic en él, se podrá compilar el archivo .ASM y generar el .HEX, luego programar el PLC.

En este caso (hardware) las señales se conectarán según se muestra en la figura 3.18, la señal de control de evento se introduce en IN4, IN3, IN2, IN1 y la secuencia de salida del evento seleccionado se obtendrá en los pines de alta corriente Q4, Q3, Q2 y Q1. El pulso de reloj entra por la pata AN1, que aunque tiene función analógica, se ha configurado para responder a una señal en la que un voltaje de 0 a 2.5 es equivalente a un “bajo”, arriba de esto será un “alto”

A continuación se muestran las conexiones posibles en la bornera del PLC:

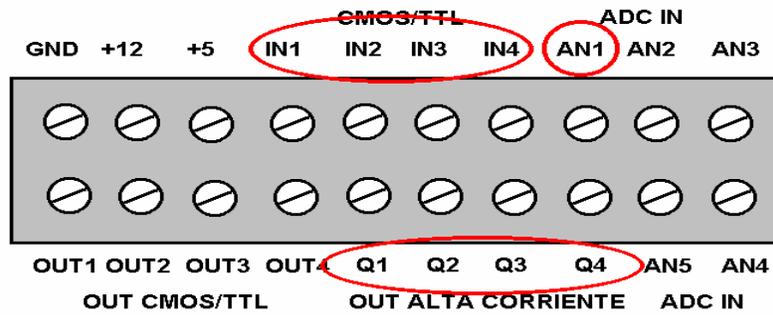


Figura 2.18 Conexiones posibles para la aplicación de lógica secuencial.

2.2.1.4 MÓDULO CONTADOR



Figura 2.19 : ventana de asignación de la entrada y salida para el contador.

Aquí los parámetros que deberá asignar son: pin de entrada, establecer por medio de los iconos “arriba” y “abajo”, ascendente/descendente y pin de salida, de lo contrario, el sistema se lo recordara y no dejará que continúe la ejecución del programa

Al final se hará clic en el botón “aceptar”, y tendrá que aceptar el resto de eventos, hasta completar la tarea de programar el PLC

2.2.1.5 TEMPORIZADORES (ASTABLE / MONOESTABLE)

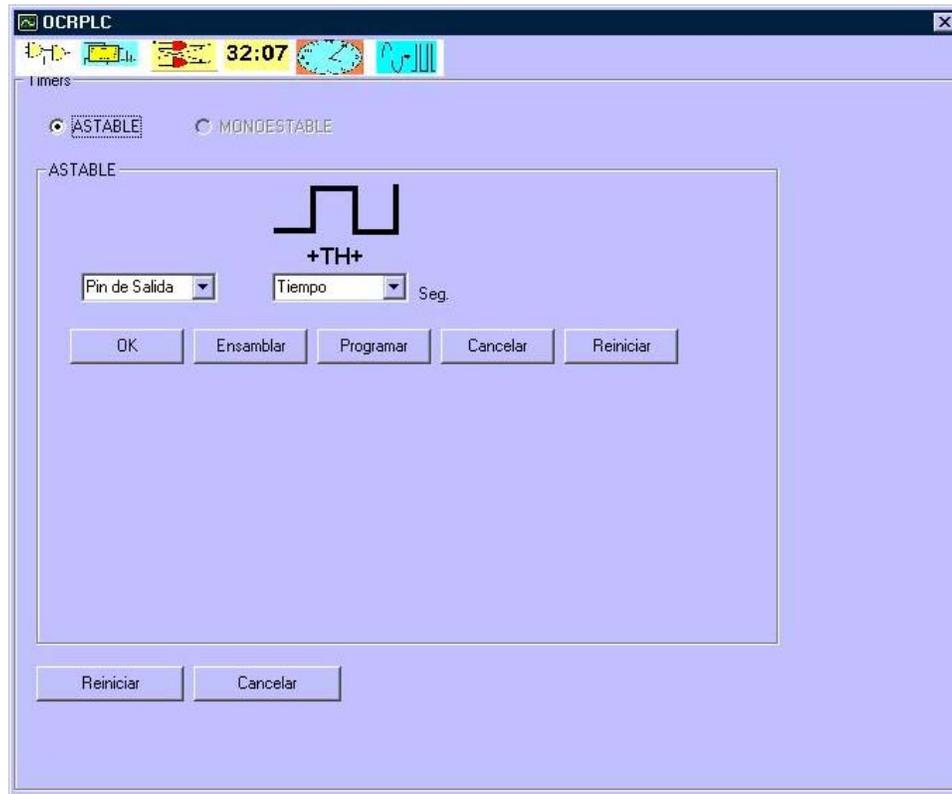


Figura 2.20: controles para programar un Temporizador Astable.

En este caso los campos a los que hay que introducir datos antes de hacer clic en el botón del comando “OK”, son: pin de salida y tiempo, como se ve este control es mas fácil de usar que todos, sus posibilidades de conexión al hardware se muestra en la figura 2.21

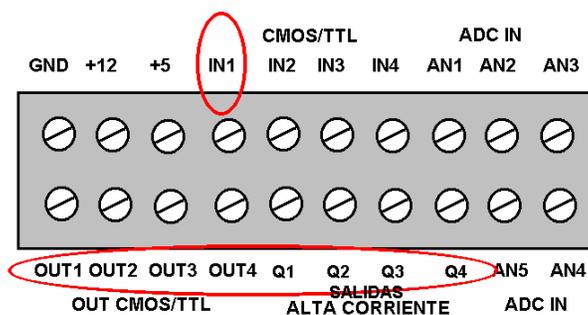


Figura 2.21: Bornera que muestra la conexión para el pulso de inicio del temporizador.

Como se muestra en la figura 2.21, el temporizador monoestable necesita que se le conecte una señal de disparo a través de IN1, y el pulso se puede obtener a través de cualquier salida, ya sea CMOS/TTL o de SALIDAS DE ALTA CORRIENTE. Sin embargo para el TEMPORIZADOR ASTABLE, no se ocupa la entrada IN1, ya que éste genera un tren de pulsos cuadrados que puede ser capturado en cualquier salida (CMOS/TTL o de SALIDAS DE ALTA CORRIENTE)

Al igual que todos los anteriores eventos, si no se ha seleccionado alguno, el sistema le avisará para que lo haga, mientras tanto no deja pasar al siguiente evento.

2.2.1.6 INTERFAZ GRAFICA PARA EL MÓDULO ADC

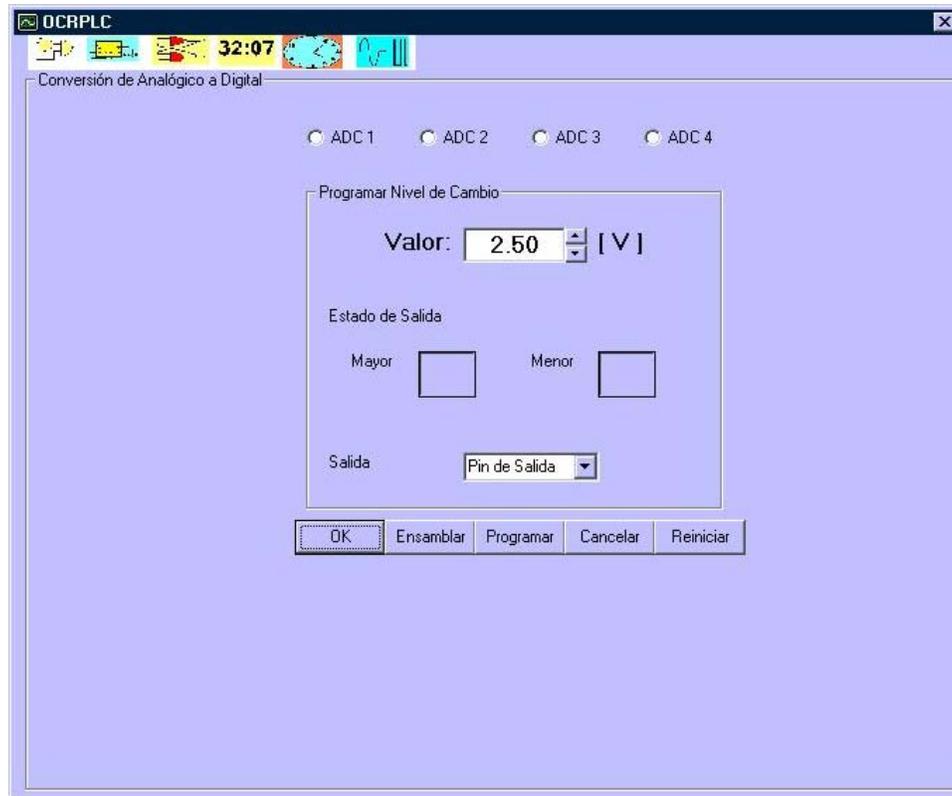


Figura 2.22: Opciones para configurar un ADC.

Aquí, solo se puede seleccionar una entrada analógica, el activar un canal, automáticamente desactiva los otros tres. Luego deberá asignar el nivel de voltaje de referencia, además de esto, deberá establecer los niveles lógicos de la salida, para cuando el voltaje de entrada sea mayor o menor que el nivel de referencia.

Las opciones de conexión para “hardware” se muestran en la figura 2.23, como se ve se puede seleccionar cualquier canal AN1, AN2, AN3, AN4 como entrada y la salida se puede conectar en las salida CMOS/TTL.

A continuación se muestra en la figura 2.23, las posibles conexiones para trabajar con el módulo ACD del PLC:

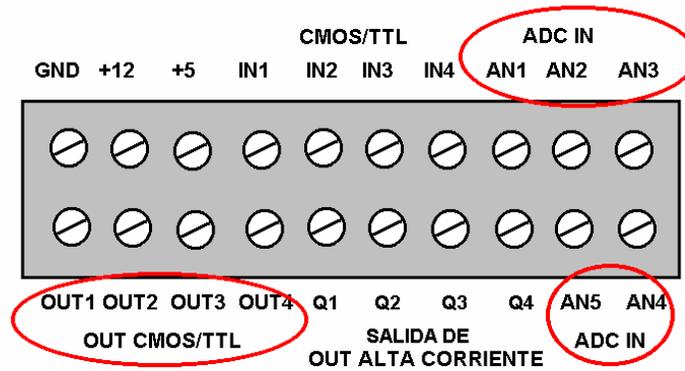


Figura 2.23: Conexión de hardware posible para el ADC.

2.2.2 AMBIENTE LINUX

A continuación se describen las diferentes opciones con las que cuenta la interfaz gráfica.

2.2.2.1 MÓDULO CONTADOR UP – DOWN

Éste módulo consta de muchas opciones las cuales son:

- Habilitación: ésta establece que el contador se debe de programar o no al PLC cuando está activada.
- Descarga: el botón de descarga de programa literalmente programa el módulo el PLC, generando una barra de progreso indicando la actividad.
- Modo: el contador se puede programar de dos maneras, las cuales son como un contador ascendente o descendente respectivamente.
- Valor de conteo: el cual es el número final o inicial dependiendo del modo al cual fue previamente programado.
- Entrada: el pin donde se sensa el pulso de la cuenta.
- Salida: es el pin donde se genera una señal de aviso de fin de cuenta.

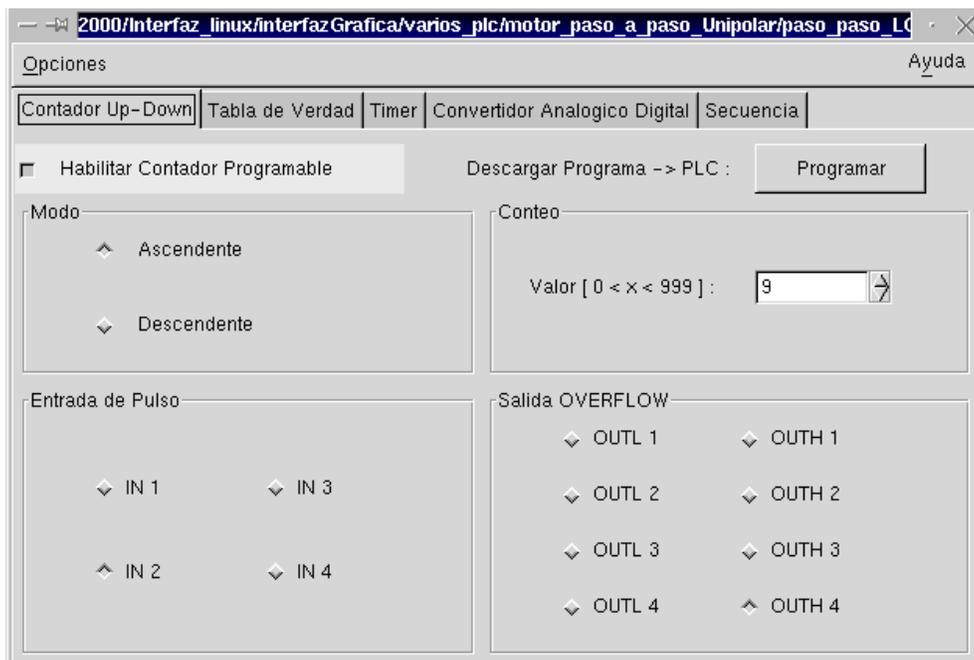


Figura 2.24: Vista grafica del Contador Up – Down

2.2.2.2 MÓDULO TABLA DE VERDAD.

Éste módulo consta de un botón de habilitación de módulo y un botón de programar; pero la parte mas importante es la generación de una tabla de verdad de cuatro entradas y ocho salidas, las cuales se pueden programar con los botones correspondientes. La tabla de verdad se muestra en la siguiente figura: 2.25

programable): /mnt/win2000/Interfaz_linux/interfazGrafica/vari0s_plc/motor_paso_a_paso

Opciones Ayuda

Contador Up-Down **Tabla de Verdad** Timer Convertidor Analogico Digital Secuencia

Habilitar Tabla de Verdad Descargar Programa -> PLC : Programar

IN 4	IN 3	IN 2	IN 1	OUTL4	OUTL3	OUTL2	OUTL1	OUTH4	OUTH3	OUTH2	OUTH1
L	L	L	L	L	H	H	H	H	H	H	H
L	L	L	H	L	L	L	L	L	H	H	H
L	L	L	H	L	L	L	L	L	H	H	H
L	L	H	L	L	L	H	L	L	H	H	H
L	L	H	H	L	L	H	L	L	H	H	H
L	H	L	L	H	H	H	L	H	L	H	H
L	H	L	H	L	H	L	H	H	H	H	L
L	H	H	L	H	L	L	L	L	L	L	H
L	H	H	H	L	H	H	H	H	L	L	H
H	L	L	L	L	H	L	H	L	H	H	H
H	L	L	H	L	L	H	L	H	H	H	L
H	L	H	L	L	H	L	L	H	H	L	L
H	L	H	H	L	H	H	H	L	L	L	L
H	H	L	L	L	L	L	L	H	L	H	H
H	H	L	H	L	H	H	L	H	H	H	L
H	H	H	L	L	H	H	H	H	L	H	H
H	H	H	H	L	L	H	H	H	L	L	H

Figura 2.25: Visualización gráfica de la Tabla de Verdad.

2.2.2.3 MÓDULO TEMPORIZADOR.

El siguiente módulo consta de las siguientes partes:

- Botones de habilitación y de programación: son los que dan el control sobre el módulo.
- Tipo de temporizador: se pueden tener dos maneras de programación; la primera es un generador de señales astables o periódicas y la segunda es un pulso por medio de un evento aplicada a la entrada correspondiente.
- Tiempo: se programa el tiempo del medio ciclo (en el caso de Astable) y el tiempo de activación (en el caso Mono Astable).
- Salida: es el pin por medio del cual sale la señal.

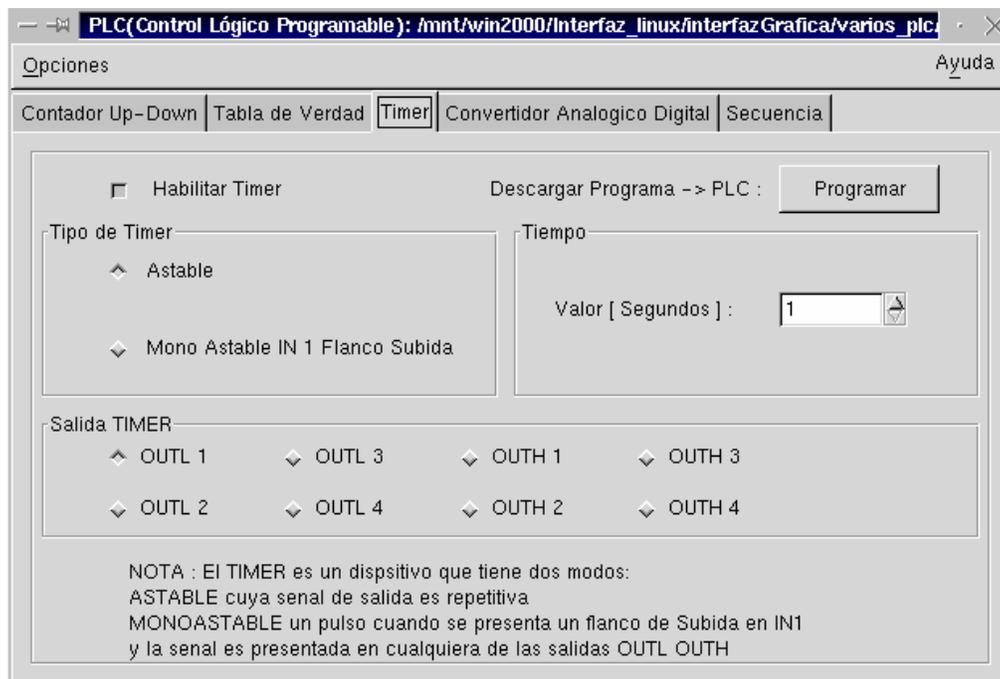


Figura 2.26: Visualización gráfica del Timer

2.2.2.4 MÓDULO CONVERTIDOR ANALÓGICO DIGITAL (ADC)

Éste módulo es el que permite conectarse al mundo analógico o señales que varían su valor en un rango específico.

Tiene dos botones que controlan el módulo los cuales son: Habilitación y Programar.

Cada uno de los submódulos tiene un control sobre un ADC respectivo, la cual se puede destacar cuatro de ellos, según los requerimientos de PLC.

Cada submódulo consta de las siguientes variables:

- Habilitación: control sobre el submódulo.
- Valor de disparo de la señal o rango.
- Selección de estado de salida
- Salida de alarma: donde se genera el pulso de cambio de estado ya sea alto o bajo, y el pulso varía según la programación de selección de estado.

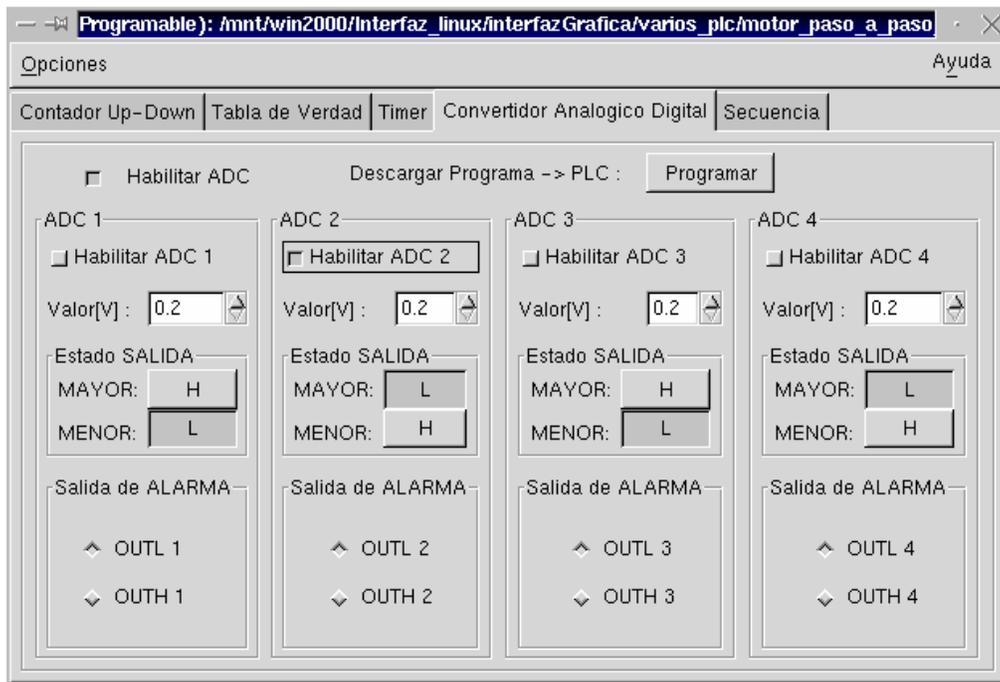


Figura 2.27: Visualización gráfica del Convertidor Analógico Digital

2.2.2.5 MÓDULO DE GENERACIÓN DE SEÑALES SECUENCIALES.

El siguiente módulo permite generar señales secuenciales, las cuales se pueden programar por medio de los botones de la tabla de tiempos. Cada Tx se programa con la casilla de tiempo. También posee una condición de paro de la secuencia en caso de un error en el sistema y genera una señal de alarma, la cual coloca un pulso alto en la salida correspondiente de la selección.

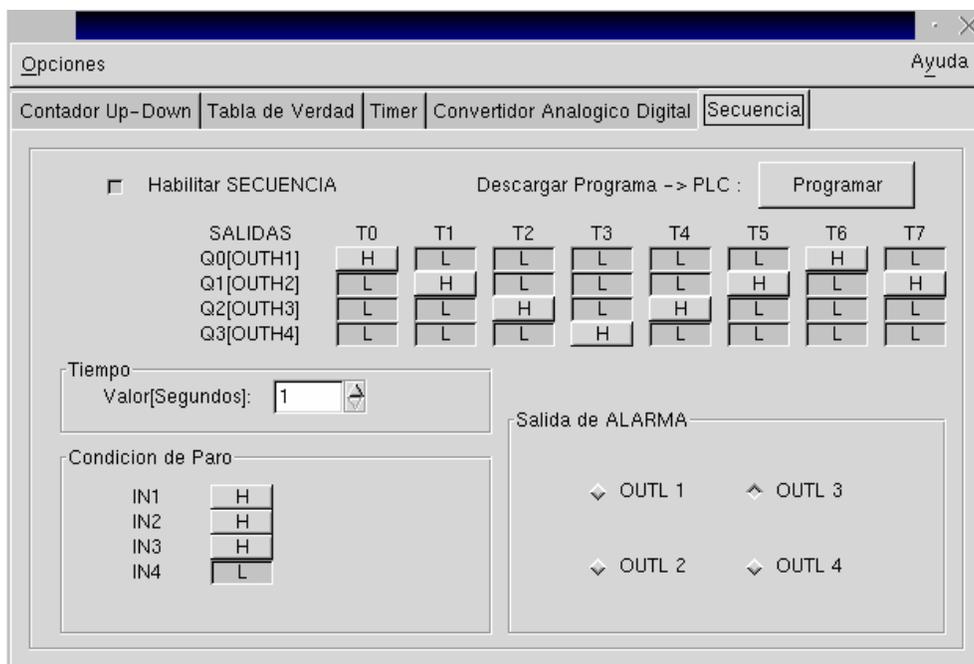


Figura 2.28: Visualización grafica de la Generación de Secuencias.

2.2.2.6 DIFERENTES MENSAJES O CUADROS DE DIALOGO EN EL PROGRAMA.

a) Mensajes de ERROR:

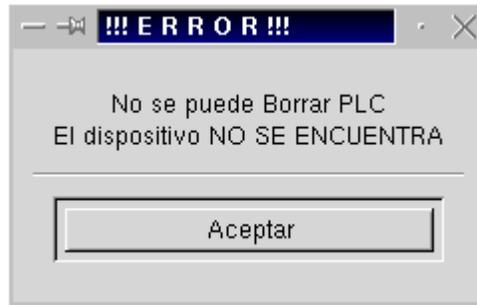


Figura 2.29: Mensaje de ERROR

b) Mensajes de dialogo, como por ejemplo Acerca de...



Figura 2.30: Cuadro de Dialogo

c) Generación de una Barra de Progreso, que indica el estado de Programación del PLC en la rutina correspondiente.

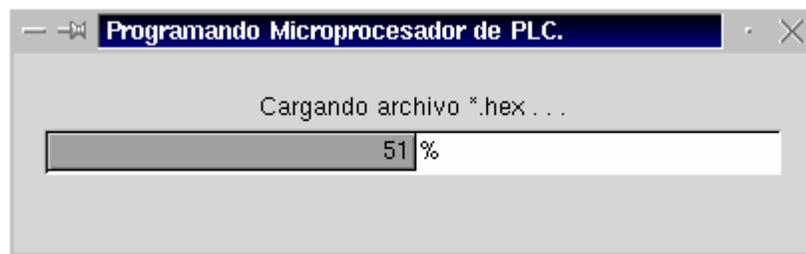


Figura 2.31: Barra de Progreso

2.3 EJEMPLOS DE IMPLEMENTACIÓN DEL PLC

2.3.1 ARRANQUE DE UN MOTOR TRIFÁSICO

Definición del problema:

El arranque de un motor 3Ø de polos conmutables con dos devanados separados (cuatro velocidades), iniciando en la mínima y terminando con la máxima. Con el diagrama de control que se muestra a continuación:

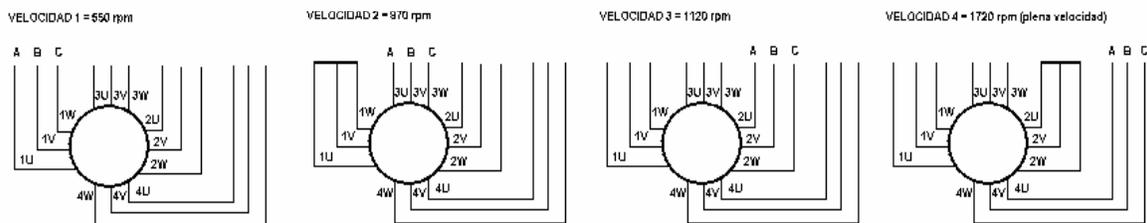


Figura 2.32. Diagrama de conexión para las diferentes velocidades.

Las letras ABC indican la conexión trifásica.

DESARROLLO:

1. El primer paso es obtener el diagrama de tiempo del sistema.

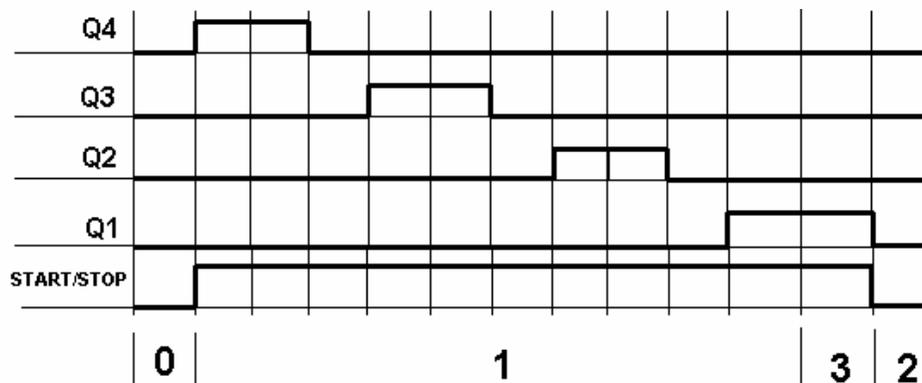


Figura 2.33. Diagrama de temporización del arranque del motor, cada tiempo es igual a dos segundos.

El diagrama de tiempo responde de la siguiente manera: Para obtener la velocidad de 550 rpm, se tiene que alimentar los devanados 1W, 1V y 1U, dejando abiertas el resto de las terminales.

Para obtener la velocidad de 870 rpm, es necesario alimentar los devanados 2W, 2V y 2U.

Para la velocidad de 1120, se alimentan los devanados 3W, 3V y 3U, pero se cortocircuitan los devanados 1W, 1V y 1U.

La última velocidad se obtiene alimentando los devanados 4W, 4V y 4U, pero cortocircuitando 2W, 2V y 2U, según muestra la figura 2.32.

Como se ve en el diagrama de tiempo, al inicio todas las salidas (Q4~Q1) están en cero, indicando que no hay actividad alguna en el motor, llamando a éste evento: STOP, o evento CERO.

Cuando la señal START/STOP, es activada, ésta pasa a ser uno, lo que indica que el motor empezara a funcionar a la velocidad mas baja (550 rpm), aquí inicia el evento UNO, Q4 esta activada (nivel H), luego de cuatro segundos, se desactiva la señal, el sistema reposa ($Q4 \sim Q1 = 0$), durante dos segundos, luego activa a Q3 , que se encarga de alimentar los devanados 2W,2V y 2U, para obtener una velocidad de 870 rpm, cuando han transcurrido cuatro segundos, el sistema vuelve a reposar durante dos segundos, luego de esto se activa Q2, que alimenta los devanados 3W,3V y 3U, al mismo tiempo cortocircuita 1W,1V y 1U, obteniendo una velocidad de 1120 rpm, el sistema vuelve a reposar durante dos segundos. La velocidad Plena (1720 rpm) se obtiene cuando se alimentan los devanados 4W, 4V y 4U directamente y se cortocircuitan los devanados 2W, 2V y 2U, aquí termina el evento UNO, quedando el motor en velocidad nominal, por un periodo indefinido se mantiene el motor a velocidad plena, que corresponde al evento TRES, el cual acaba cuando la señal START/STOP se desactiva (vuelve a cero), siendo este el evento DOS.

2. El segundo paso es obtener la tabla de Verdad del sistema.

Al analizar el diagrama de Temporización, se puede ver que la señal START/STOP es una de las variables de control de entrada, ya que si vale cero el motor esta inoperante, y si vale uno el motor funciona.

La otra señal de control de entrada se ve que es la salida Q1, que durante las tres primeras velocidades del motor esta en cero, y a plena velocidad esta en uno. Con esto se puede configurar la siguiente tabla de verdad:

Eventos	IN2=Q1	IN1 =START/STOP	Descripción
Evento cero	0	0	STOP
Evento uno	0	1	Inicio del arranque
Evento dos	1	0	Stop
Evento tres	1	1	Plena velocidad indefinidamente

Tabla 2.2. Tabla de verdad para el funcionamiento del motor.

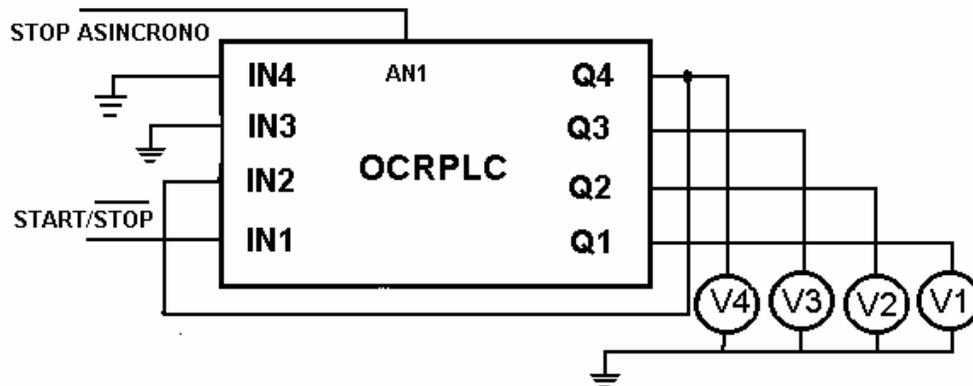


Figura 2.34 conexión para el control de velocidad de un motor trifásico

En la figura 2.34 se presenta v1 como el sistema de conexionado de los contactores - relés de estado sólido para arrancar el motor a la velocidad de 550 rpm alimentando los devanados 1W, 1V y 1U como se muestra en la figura 2.34, de la misma forma funciona para V2, V3, V4, para lograr las velocidades de 870, 1120 y 1720 rpm respectivamente.

3. El tercer paso es programar el PLC.
 - a. Se configura el evento CERO , al inicio; se establece el número de tiempos del evento, que según indica el diseño, es uno, luego se tiene que hacer clic en el botón “aceptar tiempos”. Para asignar un estado alto (uno lógico), tiene que hacer clic en el intervalo de tiempo deseado, si quiere un estado bajo (un cero lógico), no haga nada, este valor lo tiene asignado por defecto, al terminar la asignación haga clic en el botón “configurar siguiente evento”, esto lo llevara al siguiente evento, que en este caso es el evento 1, que se muestra en la figura 2.35, de igual manera se procede para el resto de eventos.

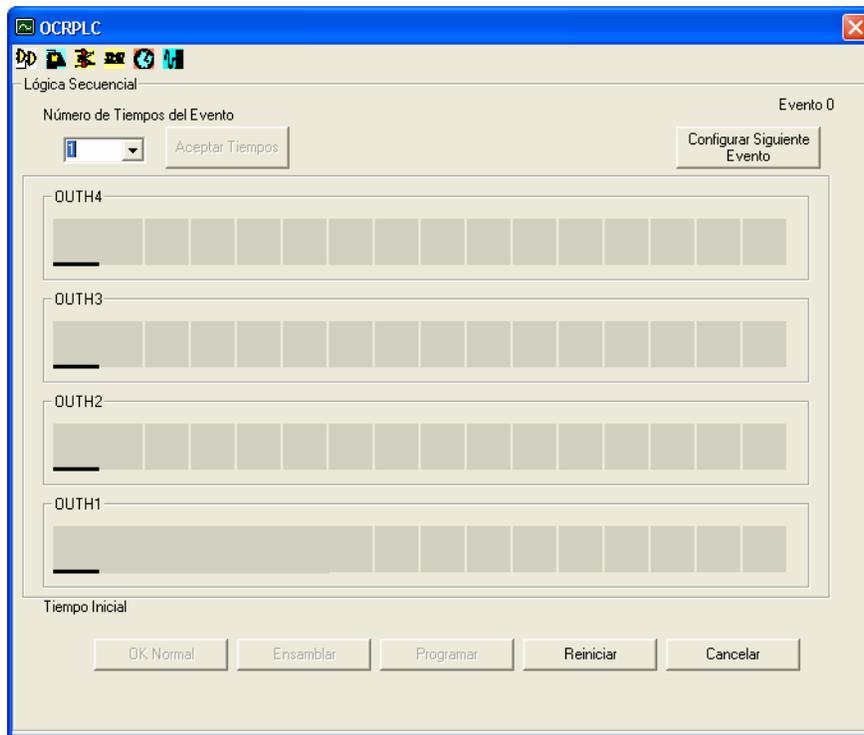


Figura 2.35 Configuración del evento cero

b. Configuración del evento UNO

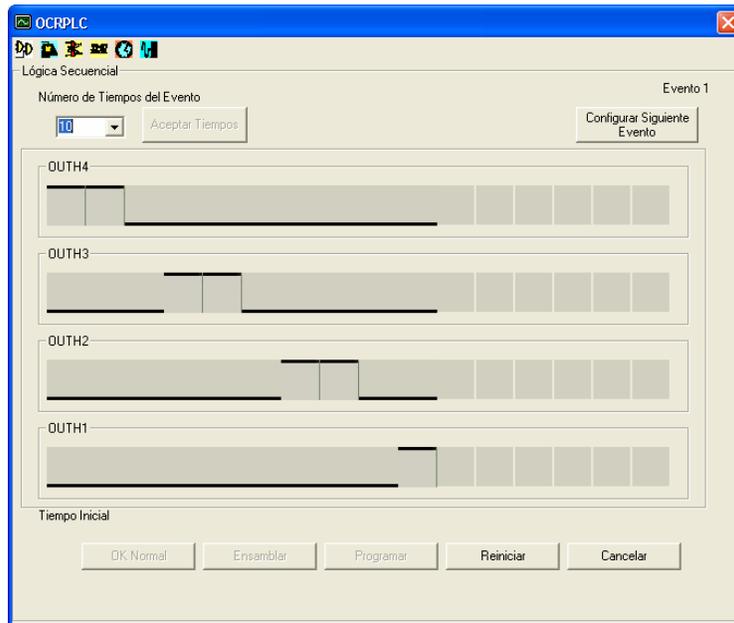


Figura 2.36 Configuración del evento uno.

c. Configuración del evento DOS

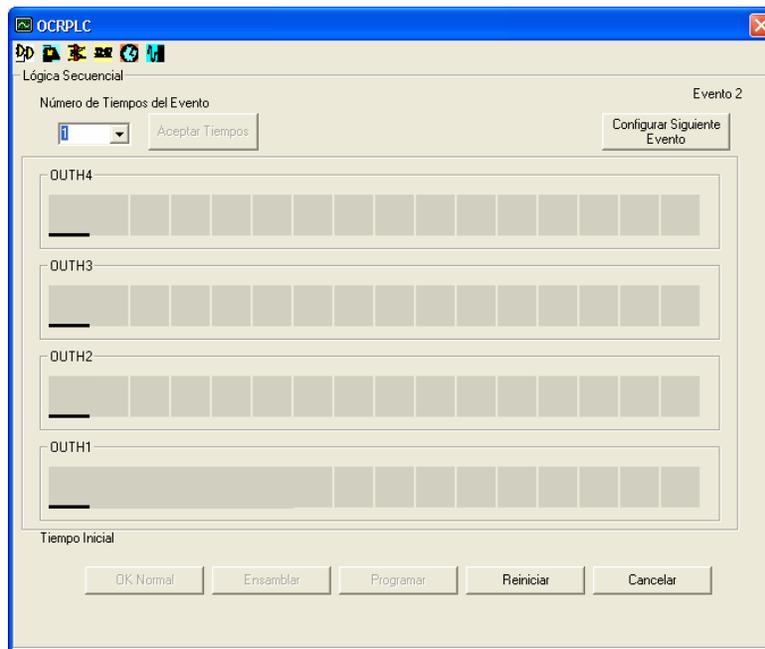


Figura 2.37 Configuración del evento DOS

d. Configuración del evento TRES

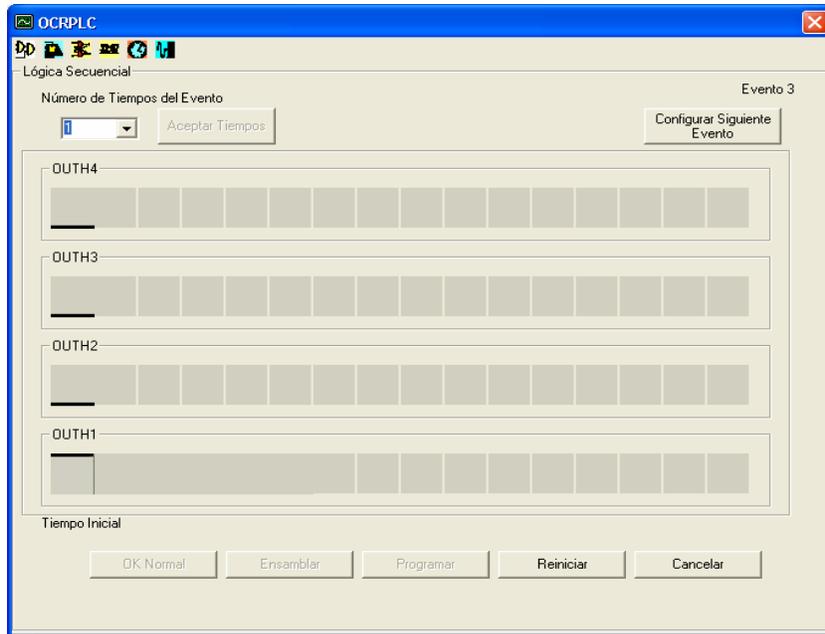


Figura 2.38 Configuración del Evento TRES

2.3.2 ARRANQUE Y PARO SECUENCIAL DE CUATRO MOTORES TRIFÁSICOS

Descripción del problema:

El arranque y paro secuencial de cuatro motores 3Ø, que cumplan el siguiente diagrama de tiempo:

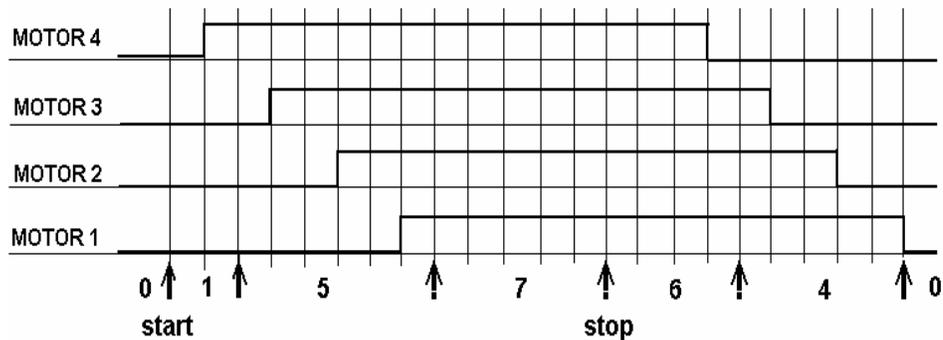


Figura 2.39 Diagrama de tiempo

1. Primer paso:

Como primer paso se analizará el diagrama de tiempo (ver figura anterior numerada 2.39), se ve que en el evento cero, todos los motores están inhabilitados, del evento cero se pasa al uno, cuando la señal START/STOP que inicialmente esta en cero, pasa a uno, justo en este momento inicia la secuencia de arranque.

Luego de dos segundos se activa el motor cuatro, con la ayuda de la función OR entre las salidas de alta corriente ($IN3 = Q4 + Q3 + Q2 + Q1$), que devuelve un UNO cuando cualquiera de los motores esta funcionando, se pasa al evento cinco, que es propiamente el arranque secuencial de los motores 3~1. Luego de esto, se auxilia de la función AND entre las salidas de alta corriente ($IN2 = Q4 * Q3 * Q2 * Q1$), cuando todos los motores estén encendidos $IN2 = H$, ésta transición servirá para que del evento 5 pase al evento 7, que es el evento en el que todos los motores están funcionando por tiempo indefinido.

Ésta situación termina cuando la señal START/STOP pasa manualmente de uno a cero, iniciándose el evento 6, aquí se da el apagado del motor cuatro, que fue el primero en arrancar, con la ayuda de la función AND (obtenida en $IN2$) se pasa del evento 6 al 4, que es el evento para apagar secuencialmente el resto de los motores, se cumplen con la característica de que el primero que encendió es el primero en apagar, y el último que

encendió es el último en apagar, quedando en un nuevo evento cero, que es el modo de motores inhabilitados.

2. El segundo paso es obtener la tabla de verdad:

eventos	IN4	IN3	IN2	IN1	Descripción de los eventos
0	0	0	0	0	Stop general, ningún motor funciona
1	0	0	0	1	Inicio, solo arranca motor 4
2	0	0	1	0	Nunca se da
3	0	0	1	1	Nunca se da
4	0	1	0	0	Apagado secuencial de los motores 3,2,1
5	0	1	0	1	Arrancado secuencial de los motores 3,2,1
6	0	1	1	0	Inicio del apagado secuencial, apaga motor4
7	0	1	1	1	Funcionamiento indefinido de los motores

Tabla 2.3 Tabla de Verdad

Las funciones AND y OR anteriormente mencionadas, se obtienen en las salidas OUTL2 y OUTL3, respectivamente, cumpliendo la siguiente ecuación booleana.

$$\text{OUTL2} = Q4 * Q3 * Q2 * Q1, \quad \text{ecuación 2.1}$$

$$\text{OUTL3} = Q4 + Q3 + Q2 + Q1, \quad \text{ecuación 2.2}$$

El diagrama a bloques del control de arranque y apagado secuencial con PLC, se muestra en la figura 2.40.

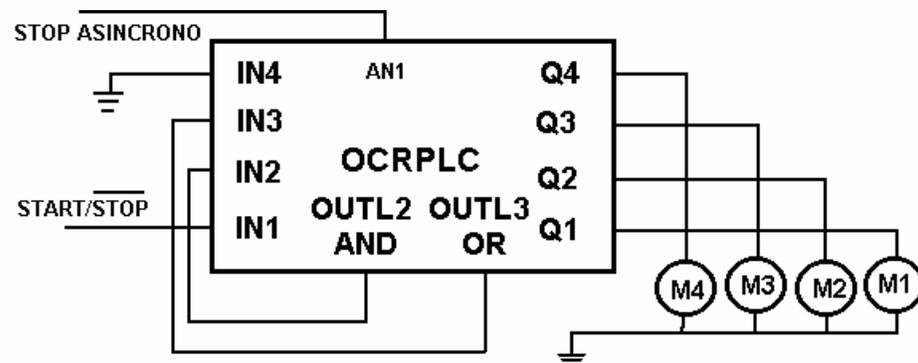


Figura 2.40 Conexión de los cuatro motores con el PLC

A continuación se presentan las pantallas de configuración del PLC, diagramas de configuración de los eventos (0~7) del PLC, que como se mencionó en el ejemplo anterior, primero se configura el evento CERO, se establece el número de tiempos del evento, que según indica el diseño, es uno, luego se tiene que hacer clic en el botón “aceptar tiempos”.

Para asignar un estado alto (uno lógico), tiene que hacer clic en el intervalo de tiempo deseado, si quiere un estado bajo (un cero lógico), no haga nada, este valor lo tiene asignado por defecto, al terminar la asignación haga clic en el botón “configurar siguiente evento”, esto lo llevara al siguiente evento, que en este caso es el evento 1, que se muestra en la figura 2.42, de igual manera se procede para el resto de eventos.

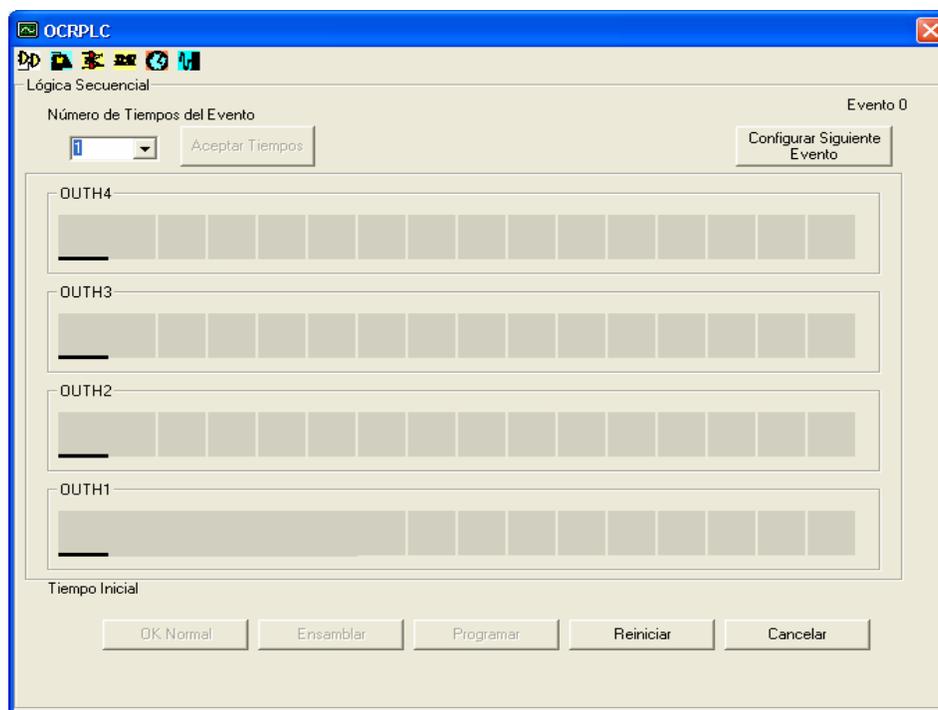


Figura 2.41 Evento CERO

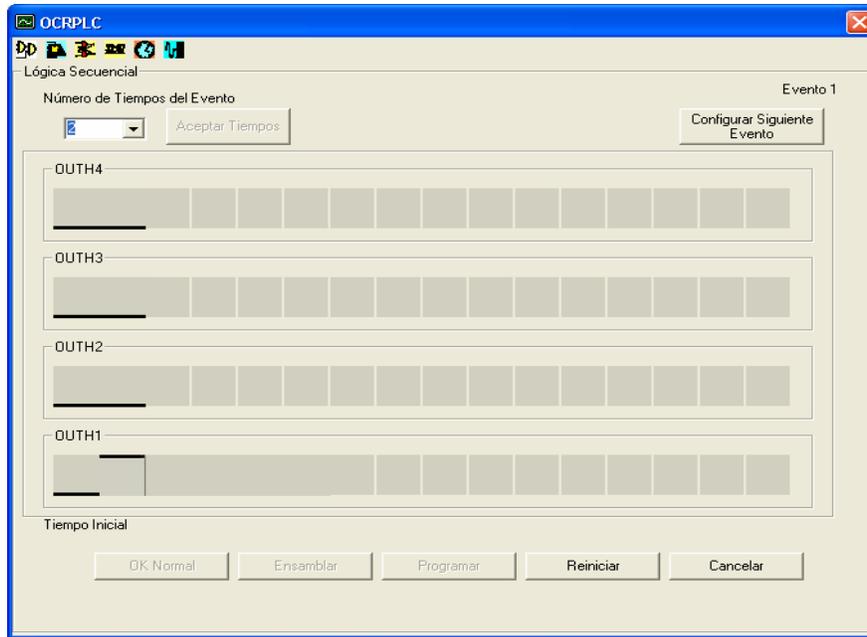


Figura 2.42 Evento UNO

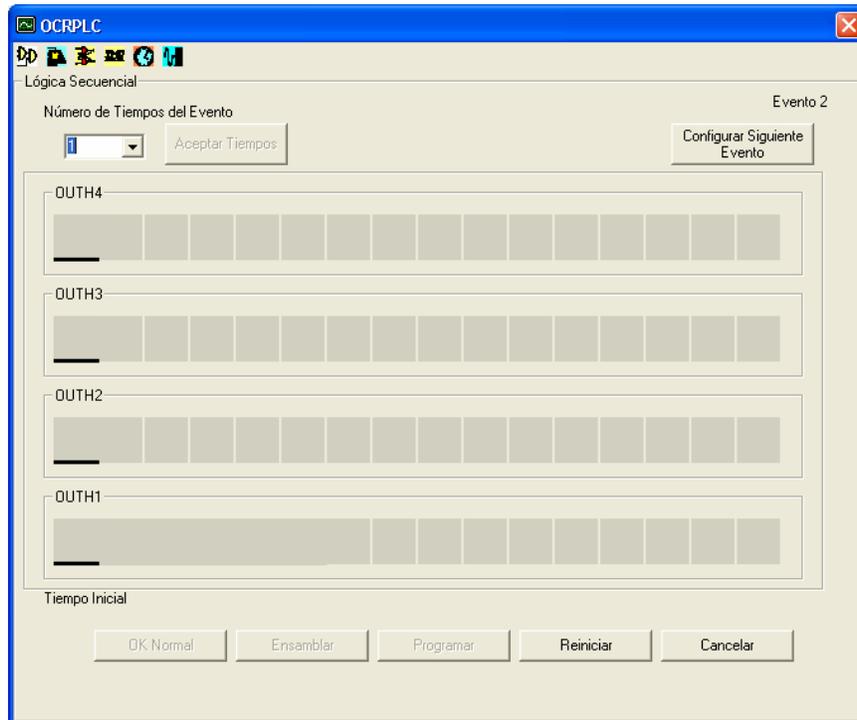


Figura 2.43 Evento DOS

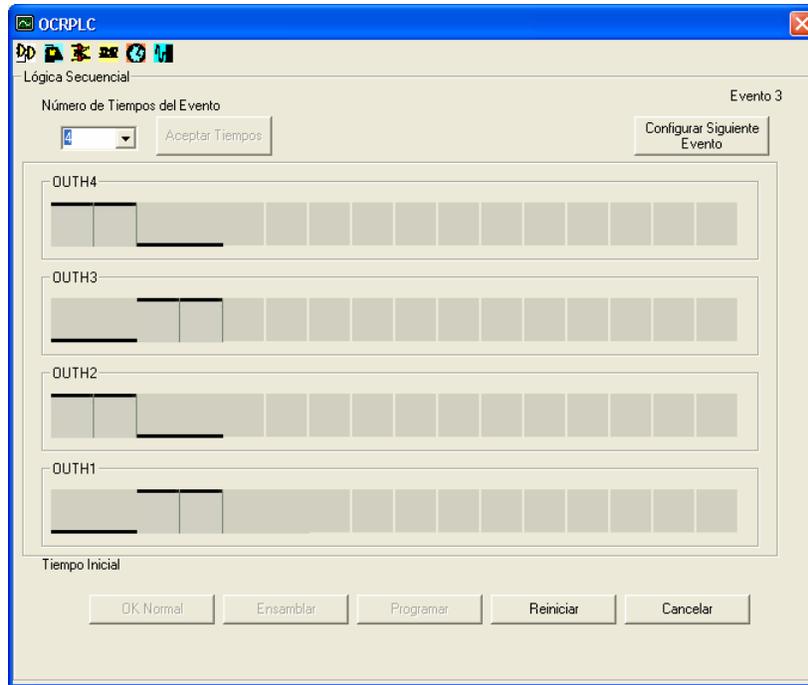


Figura 2.44 Evento TRES

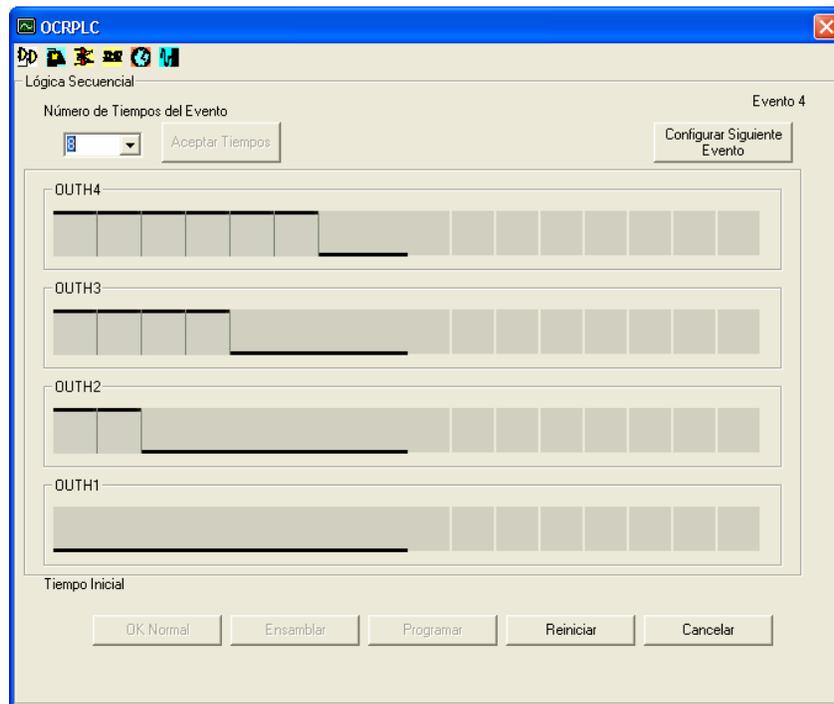


Figura 2.45 Evento CUATRO

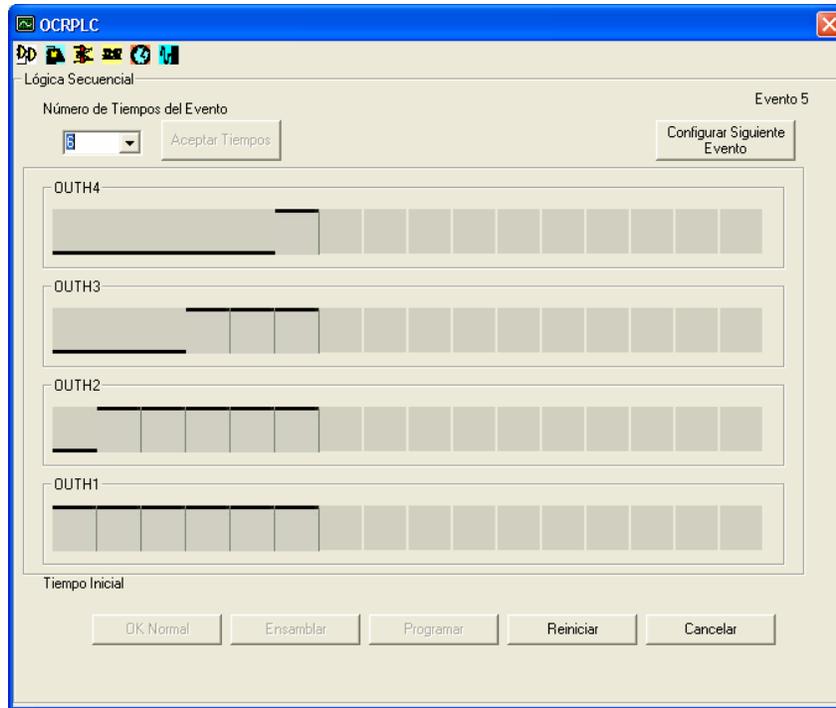


Figura 2.46 Evento CINCO

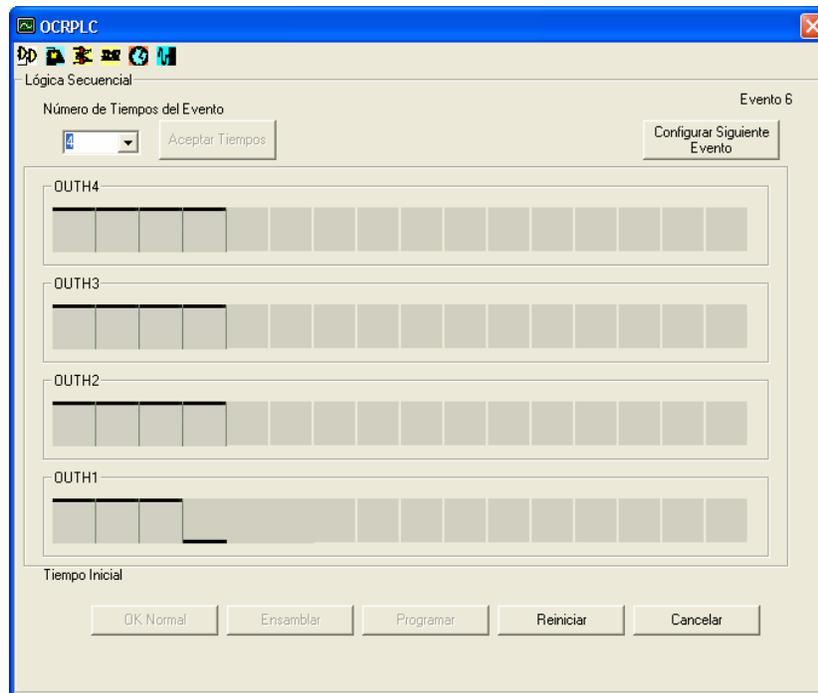


Figura 2.47 Evento SEIS

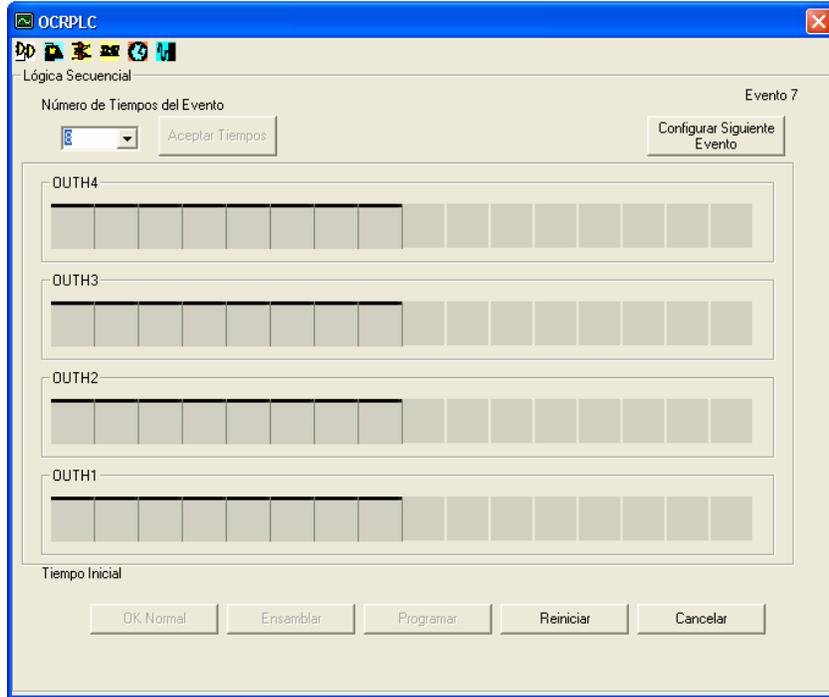


Figura 2.48 Evento SIETE

3. CONCLUSIONES

- ❑ Del estudio de mercado se ha podido descubrir que la función que mas se utiliza es el control de motores, ya sea en el arranque, cambio de la velocidad, paro del motor, así como también el monitoreo y control de la temperatura.
- ❑ Las empresas que utilizan motores; en su mayoría son motores trifásicos, seguidos de los monofásicos y en muy poca cantidad los motores de DC en las empresas salvadoreñas.
- ❑ De acuerdo al estudio de mercado, se nos refleja que sí hay buenas expectativas del mercado consumidor para el producto a implementar.
- ❑ En cuanto al sistema operativo de programación, el más conocido, difundido y más utilizado es el basado en Windows en cualquiera de sus versiones que cualquier otro sistema operativo.
- ❑ Según la marca en nuestro país la mas difundida es SIEMENS por su representatividad, su calidad, soporte técnico, garantía y servicio al cliente, que otras marcas que existen en otros mercados pero que no han sido tan difundidas en nuestro país.
- ❑ La mayoría de empresas no tiene un mantenimiento preventivo sino un mantenimiento correctivo en cuanto a los PLC's.
- ❑ Las empresas que han implementado la automatización son empresas grandes que utilizan PLC's con un número de entradas y salidas de once en adelante de donde se puede dar cuenta de que si valoran los beneficios de la automatización y la importancia que tiene en sus procesos de control y por ende de la producción, calidad, seguridad y ahorro.
- ❑ Basándose en los resultados de las encuestas realizadas se ha notado un poco de desconfianza a equipos manufacturados en El Salvador sin embargo, hay un alto porcentaje²⁵ del mercado que si esta abierto a poderlo comprar y probarlo en sus procesos de control que es el mercado al cual se quiere llegar.

²⁵ Más del 60%, según la pregunta 19 de la encuesta de la primera etapa(Anexo 5).

- ❑ Se diseñó el PLC utilizando la tecnología del PIC ya que nos permite eliminar uno de los circuitos de apoyo²⁶ que se necesitaría si se toma la decisión de utilizar el CPLD, además que posee un mayor control del tiempo real.
- ❑ En cuanto a la programación del PIC, este se programa de una forma secuencial que se acopla mas para el funcionamiento de un PLC.
- ❑ Del estudio financiero considerando solo la materia prima, equipo y ensamble el costo de producción de solo un PLC es de US\$ 351.55 que por supuesto bajaría al incrementar la producción.
- ❑ Que el microcontrolador PIC16F877 es el elemento idóneo para realizar nuestro PLC, ya que con este se facilita la implementación de las características del Prototipo del PLC.
- ❑ El uso del Módulo In Circuit Serial Programming (ICSP) del PIC, facilita la programación, permitiendo que el sistema se vuelva mas versátil.
- ❑ Proteger las patas I/O del PIC con buffer tales como 74LS245 y/o interruptores analógicos, permite que el PIC sea mas confiable, ya que si ocurre algún fallo externo, solo afecta a los circuitos de protección y no al propio PIC.
- ❑ La exactitud y precisión en el manejo del tiempo en los procesos que hace el PIC, se ven enaltecidas por el tipo de oscilador que este ocupa, que es un resonador.
- ❑ Para inmunizar al PLC contra ruido, es necesario trabajar con la norma de voltajes CMOS a 12 voltios, para ello, se ocupara un IC que tiene como función convertir el nivel de voltaje de 5V a uno de 12V y viceversa.
- ❑ Uno de los mejores programas de distribución gratuita para programar PIC es el IC-Prog, ya que este puede programar una gran gama de dispositivos, provee control de señales y “hardware”; también el PicProg en ambiente LINUX es un buen programador de PIC, pero no tiene la gran variedad de dispositivos como su homologado en Windows (IC-Prog).
- ❑ Al crear los programas que tienen la capacidad de descargar cualquier tipo de configuración a nuestro PLC, se puede llegar a obtener un resultado favorable respecto a la utilización, verificando la manipulación fácil para el usuario que era uno de los objetivos que se querían alcanzar con la creación de éste software

²⁶TDA 8764ATS, Convertidor de analógico digital de 10 bits de salida y una sola línea de entrada

- ❑ Al trabajar en lenguaje de bajo nivel con los microprocesadores (ensamblador) se tiene que ser especialistas en el dispositivo, por lo consiguiente la programación se hace lenta y confusa; en cambio al tener la recopilación de la mayoría de los manejadores de los periféricos del procesador en una interfaz gráfica facilita la configuración, solamente con oprimir botones.
- ❑ Una de las mejores características que se puede destacar en el sistema operativo Microsoft Windows a través de Visual Basic 6.0, es su versatilidad de programación y manipulación del ambiente gráfico.
- ❑ Cuando se programa en el ambiente LINUX, se puede destacar muy poca manipulación del ambiente gráfico (Utilización de la versión RED HAT 7.0), la cual dificulta la creación de ventanas que sean amigables para el usuario común, por lo que este debe de tener conocimientos definidos de la construcción física de nuestro PLC para poder configurarlo, y así darle su uso adecuado.
- ❑ Para la creación del software se utilizo GTK+ 1.2v, (Kernel 2.2.16 en la distribución RED HAT 7.0 de LINUX), por lo que recomienda uno igual o superior para su correcto mantenimiento y funcionamiento.
- ❑ El software es capaz de grabar cualquier archivo.hex (Archivo con formato INTEL) pero es importante tener en cuenta las características del PLC.
- ❑ El correcto manipuleo del PLC depende plenamente del completo conocimiento del hardware y software, por lo que cada vez que se configure el PLC, se debe de revisar todas las conexiones de la bornera y posteriormente apretar el botón de reinicio para habilitar los periféricos del microcontrolador (LCD)y los puertos respectivos, evitando así daños al PLC.
- ❑ El diseño del PLC, no abarca la comunicación serie(UART, I²C, sincrona), Modulación por ancho de pulso(PWM) y el convertidores de digital a analógico; por lo que se recomienda para futuros trabajos la implementación de estos recursos que están presentes en la arquitectura propia del procesador utilizado (PIC16F877).

4. BIBLIOGRAFÍA Y FUENTES DE CONSULTA

- Angulo Usategui Jose Maria, MICROCONTROLADOR PIC
Segunda Edición, Mc Graw Hill
España 1999

- Baca Urbina, Gabriel
Evaluación de proyectos. 3ra. Edición
Editorial Mc Graw Hill
México, 1995.

- Balena, Francisco Programación Avanzada con Microsoft VISUAL BASIC 6.0
Mc Graw Hill, 1996

- Colina Schmelkes
Manual para la Presentación de Anteproyectos e Informes de Investigación
Industria Editorial México, 1988

- Deitel /H.M. P.J. Deitel, COMO PROGRAMAR EN C/C ++
Segunda edición, Prentice Hall.

- Haber Audrey & Richard P. Runyon
Estadística General, 1ra Edición Español
FONDO EDUCATIVO INTERAMERICANO, S.A.

- The CMOS Device Manual

- Manual del PIC16F87X Microchip™, 2001
México, 1972.

- Texas Instruments , TTL Logia, 1988

Sitios importantes en Internet:

- www.microchip.com
- www.ic-prog.com
- www.sanken-ele.co.jp/e
- www.guille.costasol.net/tips_msdn/tips02.htm#990514
- www.geocities.com/siliconvalley/garage/6472/menu.htm
- www.vb2themax.com/itembank.asp?pageid=codebank&cat=140cat
- www.gtk.org
- www.xilinx.com
- www.microchip.com
- www.Libros.es.gnome.org

5. ANEXOS

ANEXOS

ANEXO A.1

PRUEBA PILOTO.

¿Nos permite unos minutos de su valioso tiempo?

Estamos realizando una encuesta preliminar, que nos permitirá ,determinar si existe un mercado potencial para la venta de nuestro PLC .

¿Cuántos PLC' estaría dispuesto a comprar para automatizar su empresa?

- 1
- 2
- 3
- ninguno
- mas, cuantas _____

ANEXO A.2

FORMATO DE ENCUESTA, ESTUDIO DE FACTIBILIDAD.



UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Somos egresados de la carrera de ingeniería eléctrica de la UES. Y estamos Solicitando su colaboración para responder las siguientes preguntas las cuales nos permitirán realizar el trabajo de graduación sobre la aceptación de un PLC básico , Agradecemos de antemano sus respuestas .reiterándole que la información proporcionada será utilizada de manera confidencial.

DATOS DE CLASIFICACIÓN

Nombre de la empresa _____

Zona _____

Producto(s) _____

Cual es la actividad de la empresa?

CUESTIONARIO

1. ¿En sus procesos de fabricación, que tipos de equipos tienen?

- Motores,
 - i. Tipo: AC DC
 - ii. Sistema Trifásico Monofásico
 - iii. Cantidad _____.
- Lámparas
- Hornos
- Compresores
- Otros: _____.
- Ninguno

Si su respuesta fue ninguno, pasar a la número 20.

2. ¿Qué tipo de control usan en esta empresa en los procesos de fabricación?

- Manual ON/OFF
- Semiautomático
- Automático PLC's
- Otros : _____.

Si su respuesta es Automático, seguir con la encuesta, si no, pase a la pregunta 15

3. ¿Cuántos PLC posee?

- 1 – 2
- 3 – 5
- Mas.

4. ¿Qué marcas de PLC's tiene en su proceso de producción?

- SIEMENS
- HITACHI
- ABB
- OMRON
- Otros: _____

5. ¿Cómo programa este PLC?

- Manual
- PC
- Otros: _____.

Si su respuesta es manual, pasar a la pregunta 7.

6. ¿Qué Sistema Operativo utiliza para programar el PLC?

- Windows 95/98/Me/2000/XP
- Linux
- Otros: _____.

7. Con relación al tamaño de su PLC, ¿Cuántas entradas y salidas tiene?

- Entradas: _____
- Salidas: _____

8. ¿Qué función(es) realiza(n) el(los) PLC's?

- Control de Luz
- Control de temperatura
- Control de motores
- Otros: _____ .

9. ¿En que forma recibe el mantenimiento el(los) PLC's?

- Interno (Técnico de la empresa)
- Externo (Empresa distribuidora)
- Mixto.

Si su respuesta es Interno, pasar a la pregunta 11.

10. ¿Qué empresa le sirve el mantenimiento?

- _____ .

11. ¿Qué lo motivó para comprar el(los) PLC's?

- Es barato
- Calidad
- Soporte técnico
- Garantía
- Otros: _____ .

12. ¿A que empresa le compró el(los) PLC's?

- _____ .

13. ¿A través de que medio supo de este producto?

- Internet
- Asesoría Técnica
- Seminarios
- Expo – Técnicas
- Otros: _____

14. ¿El(los) PLC que compró le presenta algún tipo de inconveniente?

- Sí, ¿Cuáles? _____
- No

15. ¿Le interesaría controlar en forma automática sus procesos?

- Sí, ¿por que? _____
- No, ¿por que? _____ .

Si su respuesta es No, pasar al numeral 20

16. ¿Que función desea que controle el PLC?

- Control de Luz
- Control de temperatura
- Control de motores
- Otros: _____ .

17. ¿Qué precio estaría dispuesto a pagar por un PLC autóctono, de fácil programación e instalación?

- US\$200.00 – US\$400.00
- US\$401.00 – US\$600.00
- US\$601.00 – US\$800.00
- Mas.

18. ¿Cuántos PLC's compraría?

_____ .

19. ¿Compraría usted un PLC autóctono, de fácil programación, de buena presentación y hechos en El Salvador?

- Definitivamente lo compraría
- Podría comprarlo o no comprarlo
- Definitivamente no lo compraría

DATOS DE CONTROL

Nombre del entrevistador _____

Fecha de entrevista _____

Lugar de la entrevista _____

Muchas gracias por habernos ayudado a recopilar valiosa información

ANEXO A.3

DESCRIPCIÓN GENERAL DEL PIC16F877

LA FAMILIA DEL PIC16F877

El microcontrolador PIC16F877 de Microchip pertenece a una gran familia de microcontroladores de 8 bits (bus de datos interno) que tienen las siguientes características generales que los distinguen de otras familias:

- ❑ Arquitectura Harvard
- ❑ Tecnología RISC
- ❑ Tecnología CMOS

Estas características se conjugan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos y programa y por lo tanto en la velocidad de ejecución. Microchip ha dividido sus microcontroladores en tres grandes subfamilias de acuerdo al número de bits de su bus de instrucciones:

Subfamilia	Bits del bus de instrucciones	nomenclatura
Base - Line	12	PIC12XXX y PIC14XXX
Mid - Range	14	PIC16XXX
High - End	16	PIC17XXX y PIC18XXX

Tabla 1 Clasificación según el número de bits de instrucción

Variantes principales:

Los microcontroladores que produce Microchip cubren un amplio rango de dispositivos cuyas características pueden variar como sigue:

- ❑ Empaquetado (desde 8 patitas hasta 68 patitas)
- ❑ Tecnología de la memoria incluida (EPROM, ROM, Flash)
- ❑ Voltajes de operación (desde 2.5 v. Hasta 6v)
- ❑ Frecuencia de operación (Hasta 20 Mhz)

Empaquetados:

Aunque cada empaquetado tiene variantes, especialmente en lo relativo a las dimensiones del espesor del paquete, en general se pueden encontrar paquetes tipo PDIP (Plastic Dual In Line Package), PLCC (Plastic Leaded Chip Carrier) y QFP (Quad Flat Package), los cuales se muestran en las figuras siguientes:

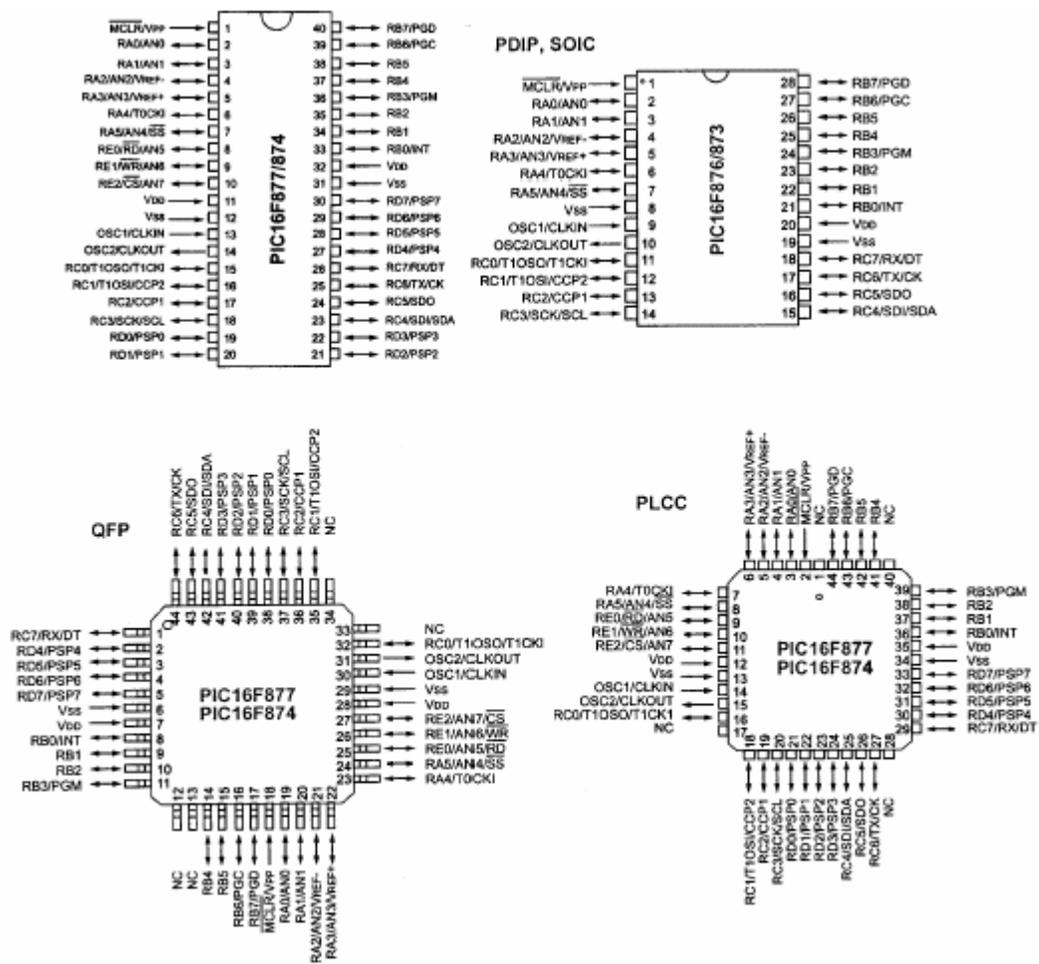


Figura 1 Detalle de construcción del PIC16F877

Nomenclatura:

Además de lo mostrado en la tabla anterior, en el nombre específico del microcontrolador pueden aparecer algunas siglas como se muestra en la siguiente tabla:

Tipo de memoria	Rango de voltaje	
	Estándar	Extendido
EPROM	PIC16CXXX	PIC16LCXXX
ROM	PIC16CRXXX	PIC16LCRXXX
Flash	PIC16FXXX	PIC16LFXXX

Tabla 2 Tipo de memoria y el rango de voltaje de alimentación.

En la siguiente tabla se especifican los rangos de voltaje estándar y extendido manejados por los dispositivos:

Rango de voltaje	EPROM		ROM		Flash	
Estándar	C	4.5 a 6v	CR	4.5 a 6v	F	4.5 a 6v
Extendido	LC	2.5 a 6v	LCR	2.5 a 6v	LF	2 a 6v

Tabla 3 Clasificación del rango de voltaje según las siglas del PIC

Oscilador:

Los PIC de rango medio permiten hasta 8 diferentes modos para el oscilador. El usuario puede seleccionar alguno de estos 8 modos programando 3 bits de configuración del dispositivo denominados: FOSC2, FOSC1 y FOSC0. En algunos de estos modos el usuario puede indicar que se genere o no una salida del oscilador (CLKOUT) a través de una patita de Entrada/Salida. Los modos de operación se muestran en la siguiente lista:

- ❑ LP Baja frecuencia (y bajo consumo de potencia)
- ❑ XT Cristal / Resonador cerámico externos, (Media frecuencia)
- ❑ HS Alta velocidad (y alta potencia) Cristal/resonador
- ❑ RC Resistencia / capacitor externos (mismo que EXTRC con CLKOUT)
- ❑ EXTRC Resistencia / capacitor externos
- ❑ EXTRC Resistencia / Capacitor externos con CLCKOUT
- ❑ INTRC Resistencia / Capacitor internos para 4 MHz
- ❑ INTRC Resistencia / Capacitor internos para 4 MHz con CLKOUT

Los tres modos LP, XT y HS usan un cristal o resonador externo, la diferencia sin embargo es la ganancia de los manejadores internos, lo cual se ve reflejado en el rango de frecuencia admitido y la potencia consumida. En la siguiente tabla se muestran los rangos de frecuencia así como los capacitores recomendados para un oscilador en base a cristal.

Modo	Frecuencia típica	Capacitores recomendados	
		C1	C2
LP	32 khz	68 a 100 pf	68 a 100 pf
	200 khz	15 a 30 pf	15 a 30 pf
XT	100 khz	68 a 150 pf	150 a 200 pf
	2 Mhz	15 a 30 pf	15 a 30 pf
	4 Mhz	15 a 30 pf	15 a 30 pf
HS	8 Mhz	15 a 30 pf	15 a 30 pf
	10 Mhz	15 a 30 pf	15 a 30 pf
	20 Mhz	15 a 30 pf	15 a 30 pf

Tabla 4 Clasificación de la frecuencia de trabajo según el tipo de oscilador.

Cristal externo: En los tres modos mostrados en la tabla anterior se puede usar un cristal o resonador cerámico externo.

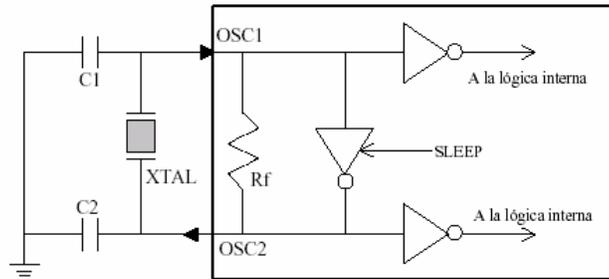


Figura 2 Oscilador con cristal para los tipos LP, XT y HS

Circuito RC externo: En los modos RC y EXTRC el PIC puede generar su señal oscilatoria basada en un arreglo RC externo conectado a la patita OSC1. Ver Figura .3

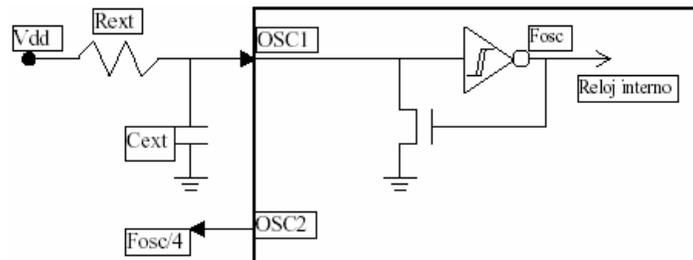


Figura 3 Oscilador tipo RC

Este modo sólo se recomienda cuando la aplicación no requiera una gran precisión en la medición de tiempos. La frecuencia de oscilación depende no sólo de los valores de R_{ext} y C_{ext} , sino también del voltaje de la fuente V_{dd} . Los rangos admisibles para resistencia y capacitor son: R_{ext} : de 3 a 100 Kohms y C_{ext} : mayor de 20 pf.

Oscilador externo: También es posible conectar una señal de reloj generada mediante un oscilador externo a la patita OSC1 del PIC. Para ello el PIC deberá estar en uno de los tres modos que admiten cristal (LP, XT o HS). Ver Figura .4:

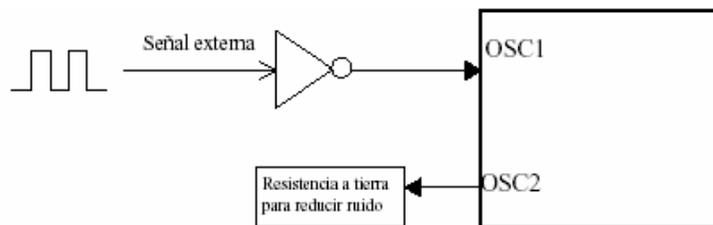


Figura 4 Oscilador externo

CARACTERÍSTICAS GENERALES DEL PIC16F877.

La siguiente es una lista de las características que comparte el PIC16F877 con los dispositivos más cercanos de su familia²⁷:

- ❑ CPU RISC, sólo 35 instrucciones que aprender.
- ❑ Todas las instrucciones se ejecutan en un ciclo de reloj, excepto los saltos que requieren dos.
- ❑ Frecuencia de operación de 0 a 20 MHz (DC a 200 nseg de ciclo de instrucción).
- ❑ Hasta 8k x 14 bits de memoria Flash de programa.
- ❑ Hasta 368 bytes de memoria de datos (RAM).
- ❑ Hasta 256 bytes de memoria de datos EEPROM.
- ❑ Hasta 14 fuentes de interrupción.
- ❑ STACK de hardware de 8 niveles.
- ❑ Reset de encendido (POR).
- ❑ Timer de encendido (PWRT).
- ❑ Timer de arranque del oscilador (OST).
- ❑ Sistema de vigilancia Watchdog timer.
- ❑ Protección programable de código.
- ❑ Modo SEP de bajo consumo de energía.
- ❑ Opciones de selección del oscilador.
- ❑ Programación y depuración serie “In-Circuit” (ICSP) a través de dos patitas.
- ❑ Lectura/escritura de la memoria flash de programa
- ❑ Rango de voltaje de operación de 2.0 a 5.5 volts
- ❑ Alta disipación de corriente de la fuente: 25mA
- ❑ Rangos de temperatura: Comercial, Industrial y Extendido
- ❑ Bajo consumo de potencia:
 - Menos de 0.6mA a 3V, 4 Mhz
 - 20 μ A a 3V, 32 Khz
 - menos de 1 μ A corriente en standby.

²⁷ Familia PIC16F87X .

Periféricos:

- ❑ Timer0: Contador/Temporizador de 8 bits con pre-escalador de 8bits.
- ❑ Timer1: Contador/Temporizador de 16 bits con pre-escalador.
- ❑ Timer0: Contador/Temporizador de 8 bits con pre-escalador y post-escalador de 8 bits y registro de periodo.
- ❑ Dos módulos de Captura, Comparación y PWM.
- ❑ Convertidor Analógico/Digital: de 10 bits, hasta 8 canales.
- ❑ Puerto Serie Síncrono (SSP).
- ❑ Puerto Serie Universal (USART/SCI).
- ❑ Puerto Paralelo Esclavo (PSP): de 8 bits con líneas de protocolo.

DIAGRAMA DE BLOQUES DEL PIC16F877

En la siguiente figura se muestra a manera de bloques la organización interna del PIC16F877.

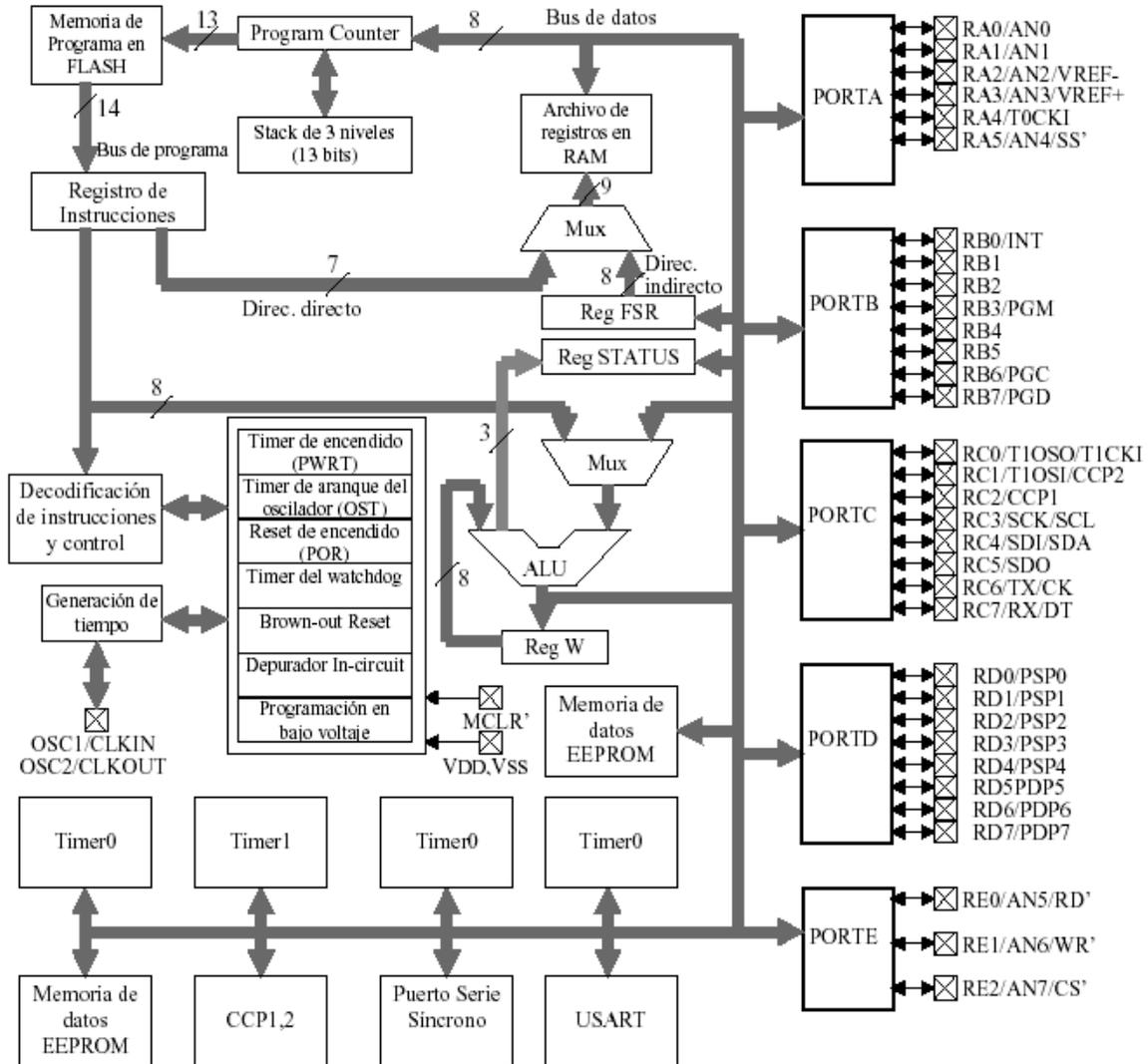


Figura 5 Diagrama de bloques del PIC16F877

DESCRIPCIÓN DE LA CPU

La CPU es la responsable de la interpretación y ejecución de las instrucciones guardadas en la memoria de programa. Muchas de estas instrucciones operan sobre la memoria de datos. Para realizar cualquier tipo de operaciones ya sean aritméticas o lógicas se requiere el uso la Unidad de Lógica y Aritmética (ALU). La ALU controla los bits de estado (Registro STATUS); los bits de este registro se alteran dependiendo del resultado de algunas instrucciones.

Ciclo de instrucción: El registro Contador de Programa (PC²⁸) es gobernado por el ciclo de instrucción como se muestra en la siguiente figura. Cada ciclo de instrucción la CPU lee la instrucción guardada en la memoria de programa apuntada por PC y al mismo tiempo ejecuta la instrucción anterior, esto debido a una cola de instrucciones que le permite ejecutar una instrucción mientras lee la próxima:

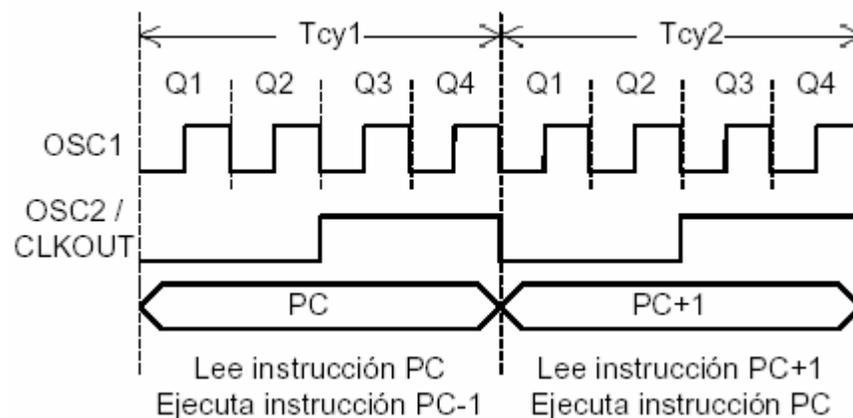


Figura 6 Ejemplificación de dos ciclos de instrucción

Como puede verse, cada ciclo de instrucción (Tcy) se compone a su vez de cuatro ciclos del oscilador (Tosc). Cada ciclo Q provee la sincronización para los siguientes eventos:

- Q1: Decodificación de la instrucción.
- Q2: Lectura del dato (si lo hay).
- Q3: Procesa el dato.
- Q4: Escribe el dato.

Debido a esto cada ciclo de instrucción consume 4 ciclos de reloj, de manera que si la frecuencia de oscilación es Fosc, Tcy será $4/Fosc$.

²⁸ PC: Program Counter

Registros de la CPU:

El registro (PC): Registro de 13 bits que siempre dirección a la siguiente instrucción que se va a ejecutar.

Registro de Instrucción:

Registro de 14 bits, en el cual se colocan todas las instrucciones para ser decodificadas por la CPU antes de ejecutarlas.

Registro W:

Registro de 8 bits que guarda resultados temporales de las operaciones realizadas por la ALU.

Registro STATUS: Registro de 8 bits, cada uno de sus bits (denominados Banderas) es un indicador del estado de la CPU o del resultado de la última operación como se indica en la siguiente figura:



➤ **Notación:** En adelante se usará lo siguiente:
R= Bit leíble W= Bit Escribe U= No implementado (se lee como 0)
-n= Valor después del Reset de encendido

Figura 7 Banderas del registro de status

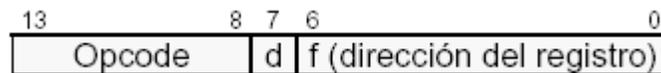
Donde:

- Z=1 para indicar que el resultado de la operación fue cero, de lo contrario se limpia.
- C.- Bit de acarreo/préstamo de la última operación aritmética.
- CD.- Acarreo/Préstamo proveniente del cuarto bit menos significativo. Igual que C.

CONJUNTO DE INSTRUCCIONES DE RANGO MEDIO.

Formato General de las Instrucciones: Cada instrucción en lenguaje de máquina (binario) del PIC contiene un código de operación (opcode) el cual puede ser de 3 a 4 o 6 bits, dependiendo del tipo de instrucción. A continuación se describe el formato para cada tipo de instrucción de los PIC de rango medio:

Operaciones con el archivo de registros orientados a bytes:



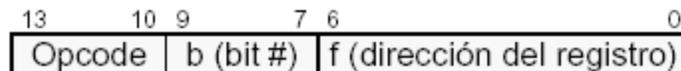
Donde:

El bit d especifica el destino del resultado de la operación: 0 destino W, 1 destino F

f = dirección de 7 bits del archivo de registros.

Donde:

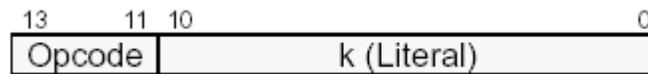
Operaciones con el archivo de registros orientados a bits:



b : Especificación en tres bits del bit sobre el que se va a operar

f = dirección de 7 bits del archivo de registros.

Operaciones con literales y de control: Formato para CALL y GOTO:



Donde:

k : Literal = Valor de un operando de 8 bits.

En la siguiente tabla se resumen las 35 instrucciones que reconoce la CPU de los PIC de rango medio, incluyendo su mnemónico, tiempo de ejecución, código de máquina y afectación de banderas:

Mnemónico	Descripción	Ciclos	Código de Máquina	Banderas afectadas
Operaciones con el archivo de registros orientadas a bytes				
ADDWF f,d	Suma f + W	1	00 0111 dfff ffff	C,DC,Z
ANDWF f,d	W AND f	1	00 0101 dfff ffff	Z
CLRF f	Limpia f	1	00 0001 1fff ffff	Z
CLRWF	Limpia W	1	00 0001 0xxx xxxx	Z
COMF f,d	Complementa los bits de f	1	00 1001 dfff ffff	Z
DECF f,d	Decrementa f en 1	1	00 0011 dfff ffff	Z
DECFSZ f,d	Decrementa f, escapa si 0	1(2)	00 1011 dfff ffff	
INCF f,d	Incrementa f en 1	1	00 1010 dfff ffff	Z
INCFSZ f,d	Incrementa f, escapa si 0	1(2)	00 1111 dfff ffff	
IORWF f,d	W OR f	1	00 0100 dfff ffff	Z
MOVF f,d	Copia el contenido de f	1	00 1000 dfff ffff	Z
MOVWF f	Copia contenido de W en f	1	00 0000 1fff ffff	
NOP	No operación	1	00 0000 0xx0 0000	
RLF f,d	Rota f a la izquierda	1	00 1101 dfff ffff	C
RRF f,d	Rota f a la derecha	1	00 1100 dfff ffff	C
SUBWF f,d	Resta f – W	1	00 0010 dfff ffff	C,DC,Z
SWAPF f,d	Intercambia nibbles de f	1	00 1110 dfff ffff	
XORWF f,d	W EXOR f	1	00 0110 dfff ffff	Z
Operaciones con el archivo de registros orientadas a bits				
BCF f,b	Limpia bit b en f	1	01 00bb bfff ffff	
BSF f,b	Pone bit b en f	1	01 01bb bfff ffff	
BTFSC f,b	Prueba bit b en f, escapa si 0	1(2)	01 10bb bfff ffff	
BTFSS f,b	Prueba bit b en f, escapa si 1	1(2)	01 11bb bfff ffff	
Operaciones con literales y de control				
ADDLW k	Suma literal k + W	1	11 111x kkkk kkkk	C,DC,Z
ANDLW k	k AND W	1	11 1001 kkkk kkkk	Z
CALL k	Llamado a subrutina	2	10 0kkk kkkk kkkk	
CLRWDT	Limpia timer del watchdog	1	00 0000 0110 0100	TO',PD'
GOTO k	Salto a la dirección k	2	10 1kkk kkkk kkkk	
IORLW k	k OR W	1	11 0000 kkkk kkkk	Z
MOVLW k	Copia literal a W	1	11 00xx kkkk kkkk	
RETFIE	Retorna de interrupción	2	00 0000 0000 1001	
RETLW k	Retorna con literal k en W	2	11 01xx kkkk kkkk	
RETURN	Retorna de subrutina	2	00 0000 0000 1000	
SLEEP	Activa Modo standby	1	00 0000 0110 0011	TO'PD'
SUBLW k	Resta k - W	1	11 110x kkkk kkkk	C,CD,Z
XORLW k	k EXOR W	1	11 1010 kkkk kkkk	Z

Tabla 5 Instrucciones que reconoce el PIC de rango medio.

ORGANIZACIÓN DE LA MEMORIA DEL PIC.

Los PIC tienen dos tipos de memoria: Memoria de Datos y Memoria de programa; cada bloque con su propio bus: Bus de datos y Bus de programa; por lo cual cada bloque puede ser accesado durante un mismo ciclo de oscilación.

La Memoria de datos a su vez se divide en:

- Memoria RAM de propósito general.
- Archivo de Registros Especiales (SFR.²⁹)

LA MEMORIA DE PROGRAMA

Los PIC de rango medio poseen un registro Contador del Programa (PC) de 13 bits, capaz de direccionar un espacio de 8K x 14. Todas las instrucciones son de 14 bits, esto significa un bloque de 8k instrucciones. El bloque total de 8K x 14 de memoria de programa está subdividido en 4 páginas de 2K x 14. La siguiente figura muestra la organización de la memoria de programa..

Dirección	
0000h	Vector de Reset
...	...
0004h	Vector de interrupción
0005h	Página 0
...	
07FFh	
0800h	Página 1
...	
0FFFh	
1000h	Página 2
...	
17FFh	
1800h	Página 3
...	
1FFFh	

Figura 8 Distribución de memoria del PIC16F877

²⁹ SFR: registros de funciones especiales

- Vector de Reset: Cuando ocurre un reset el contenido del PC es forzado a cero, ésta es la dirección donde la ejecución del programa continuará después del reset, por ello se le llama “dirección del vector de reset”.
- Vector de interrupción: Cuando la CPU acepta una solicitud de interrupción ejecuta un salto a la dirección 0004h, por lo cual a esta se le conoce como “dirección del vector de interrupción”. El registro PCLATH no es modificado en esta circunstancia, por lo cual habrá que tener cuidado al manipular el PC dentro de la Rutina de Atención a la Interrupción (Interrupt Service Routine (ISR)).

MANEJO DEL CONTADOR DEL PROGRAMA (PC):

El registro contador del programa (PC) especifica la dirección de la instrucción que la CPU buscará para ejecutarla. El PC consta de 13 bits, separados en dos partes: como se muestra en la figura siguiente:



Figura 9 Registro Contador de Programa

El byte de orden bajo es llamado el registro PCL, mientras que el byte de orden alto es llamado registro PCH. Este último contiene los bits PC<12:8> y no se puede leer o escribir directamente. Todas las actualizaciones al registro PCH deben ser hechas a través del registro PCLATH.

PAGINACIÓN:

Para saltar entre una página y otra, los bits más significativos del PC deberán ser modificados. Debido a que las instrucciones GOTO y CALL sólo pueden direccionar un bloque de 2K (pues usan una dirección de 11 bits) deben existir otros dos bits que completen los 13 bits del PC para moverse sobre los 8K de memoria de programa.

Estos dos bits extra se encuentran en un SFR denominado PCLATH (Program Counter Latch High) en sus bits PCLATH<4:3>. Por esto antes de un GOTO o un CALL el usuario deberá asegurarse que estos bits apunten a la página deseada. Si las instrucciones

se ejecutan secuencialmente el PC cruza libremente los límites de página sin necesidad de que el usuario escriba en el PCLATH.

MEMORIA DE STACK

La memoria de stack es una área de memoria completamente separada de la memoria de datos y la memoria de programa. El stack consta de 8 niveles de 13 bits cada uno. Esta memoria es usada por la CPU para almacenar las direcciones de retorno de subrutinas. El apuntador de stack no es ni legible ni escribible. Cuando se ejecuta una instrucción CALL, el PC es guardado en el stack y el apuntador de stack es incrementado en 1 para apuntar a la siguiente posición vacía. Lo contrario ocurre, cuando se ejecuta una instrucción RETURN, RETLW o RETFIE el contenido de la posición actual del stack es colocado en el PC.

- ❑ PCLATH no se modifica en ninguna de estas operaciones.
- ❑ Nota: Cuando el apuntador de stack ya está en la posición 8 y se ejecuta otro CALL se reinicia a la posición 1 sobrescribiendo en dicha posición. No existe ningún indicador que avise de esta situación. Así que el usuario deberá llevar el control para que esto no ocurra.

LA MEMORIA DE DATOS.

La memoria de datos consta de dos áreas mezcladas pero con funciones distintas:

- ❑ Registros de Propósito Especial (SFR)
- ❑ Registro de Propósito General (GPR)

Los SFR son localidades asociadas específicamente a los diferentes periféricos y funciones de configuración del PIC y tienen un nombre específico asociado con su función. Mientras que los GPR son memoria RAM de uso general.

Bancos de memoria.

Toda la memoria de datos está organizada en 4 bancos numerados 0, 1, 2 y 3. Para seleccionar un banco se debe hacer uso de los bits del registro STATUS<7:5> denominados RP1 y RP0.

La selección de bancos se basa en la siguiente tabla:

Direccionamiento Indirecto (IRP)	RP1:RP0	Banco
0	0 0	0
	0 1	1
1	1 0	2
	1 1	3

Tabla 6 Diferentes combinaciones para selección de banco específico.

Hay dos maneras de acceder a la memoria de datos:

- a) Direccionamiento directo.
- b) Direccionamiento indirecto.

Cada banco consta de 128 bytes (de 00h a 7Fh). En las posiciones más bajas de cada banco se encuentran los SFR, y arriba de éstos se encuentran los GPR. Toda la memoria de datos está implementada en RAM estática.

Direccionamiento Directo:

Para acceder una posición de memoria mediante direccionamiento directo, la CPU simplemente usa la dirección indicada en los 7 bits menos significativos del código de operación y la selección de banco de los bits RP1:RP0 en el registro STATUS.

Direccionamiento indirecto:

Este modo de direccionamiento, permite acceder una localidad de memoria de datos usando una dirección de memoria variable a diferencia del direccionamiento directo, en que la dirección es fija. Esto puede ser útil para el manejo de tablas de datos.

El Archivo de Registros.

Aunque el archivo de registros en RAM puede variar de un PIC a otro, la familia del PIC16F87x coincide casi en su totalidad. En la siguiente figura se muestra a detalle el mapa de este archivo de registros y su organización en los cuatro bancos.

En la siguiente figura, se muestra los diferentes bancos de memoria tipo RAM estática, en la cual se configura el microprocesador (SFR) y guarda datos generales (GPR)³⁰.

File Address	File Address	File Address	File Address
Indirect addr. ^(*) 00h	Indirect addr. ^(*) 80h	Indirect addr. ^(*) 100h	Indirect addr. ^(*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	105h	185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	107h	187h
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h	108h	188h
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h	109h	189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh	8Fh	EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h	90h	110h	190h
TMR2 11h	SSPCON2 91h	111h	191h
T2CON 12h	PR2 92h	112h	192h
SSPBUF 13h	SSPADD 93h	113h	193h
SSPCON 14h	SSPSTAT 94h	114h	194h
CCPR1L 15h	95h	115h	195h
CCPR1H 16h	96h	116h	196h
CCP1CON 17h	97h	General Purpose Register 117h	General Purpose Register 197h
RCSTA 18h	TXSTA 98h	16 Bytes 118h	16 Bytes 198h
TXREG 19h	SPBRG 99h	119h	199h
RCREG 1Ah	9Ah	11Ah	19Ah
CCPR2L 1Bh	9Bh	11Bh	19Bh
CCPR2H 1Ch	9Ch	11Ch	19Ch
CCP2CON 1Dh	9Dh	11Dh	19Dh
ADRESH 1Eh	ADRESL 9Eh	11Eh	19Eh
ADCON0 1Fh	ADCON1 9Fh	11Fh	19Fh
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
7Fh	EFh	16Fh	1EFh
Bank 0	accesses 70h-7Fh F0h	accesses 70h-7Fh 170h	accesses 70h - 7Fh 1F0h
	FFh	17Fh	1FFh
Bank 1	Bank 2	Bank 3	

■ Unimplemented data memory locations, read as '0'.

* Not a physical register.

³⁰ SFR: Special Function Registers GPR: General Purpose Registers

Figura 10 Mapa de memoria de registros especiales (SFR) y datos (GPR).

ANEXOS A.4

AMBIENTE DE MICROSOFT VISUAL BASIC 6.0

INTRODUCCIÓN A VISUAL BASIC 6.0

El lenguaje de programación BASIC (Beginner's All purpose Symbolic Instruction Code) nació en el año 1964 como una herramienta destinada a principiantes, buscando una forma sencilla para realizar programas, empleando un lenguaje casi igual al usado en la vida ordinaria (en inglés), y con instrucciones muy sencillas y escasas. Teniendo en cuenta el año de su nacimiento, este lenguaje cubría casi todas las necesidades para la ejecución de programas. Téngase en cuenta que las máquinas existentes en aquella época estaban estrenando los transistores como elementos de conmutación, los ciclos de trabajo llegaban a la impensable cifra de 10.000 por segundo y la memoria no pasaba de unos pocos k's en toroides de ferrita.

La evolución del BASIC por los años 70 fue escasa, dado el auge que tomaron en aquella época lenguajes de alto nivel como el FORTRAN y el COBOL. En 1978 se definió una norma para unificar los Basics existentes creándose la normativa BASIC STANDARD

Con la aparición de los primeros ordenadores personales, dedicados comercialmente al usuario particular, allá por la primera mitad de los ochenta, el BASIC resurgió como lenguaje de programación pensado para principiantes, y muchos de estos pequeños ordenadores domésticos lo usaban como único sistema operativo (Sinclair, Spectrum, Amstrad)

Con la popularización del PC, salieron varias versiones del BASIC que funcionaban en este tipo de ordenadores (Versiones BASICA, GW-BASIC), pero todas estas versiones del BASIC no hicieron otra cosa que terminar de rematar este lenguaje. Los programadores profesionales no llegaron a utilizarlo, debido a las desventajas de este lenguaje respecto a otras herramientas (PASCAL, C, CLIPPER). El BASIC con estas versiones para PC llegó incluso a perder crédito entre los profesionales de la informática.

Las razones para ello eran obvias:

- No era un lenguaje estructurado.
- No existían herramientas de compilación fiables.
- No disponía de herramientas de intercambio de información.

- No tenía librerías.
- No se podía acceder al interior de la máquina.
- Un largo etcétera de desventajas respecto a otros lenguajes de programación.

Tal fue ese abandono por parte de los usuarios, que la aparición del Quick-BASIC de Microsoft, una versión ya potente del BASIC, que corregía casi todos los defectos de las versiones pasó prácticamente inadvertida, a no ser porque las últimas versiones del sistema operativo MS-DOS incluían una versión de Quick-BASIC algo recortada (Q-Basic) como un producto más dentro de la amplia gama de ficheros ejecutables que acompañan al sistema operativo, y aprovecha de el editor de textos (Cada vez que se llama al EDIT estamos corriendo el editor del Q-Basic).

Esta versión del popular BASIC ya es un lenguaje estructurado, lo que permite crear programas modularmente, mediante subrutinas y módulos, capaz de crear programas ya competitivos con otros lenguajes de alto nivel. Sin embargo llegaba tarde, pues los entornos MS-DOS estaban ya superados por el entorno gráfico Windows.

Sin embargo algo había en el BASIC que tentaba a superarse: su gran sencillez de manejo. Si a esto se le añade el entorno gráfico Windows, el aprovechamiento al máximo de las posibilidades de Windows en cuanto a intercambio de información, de sus librerías, de sus drivers y controladores, manejo de bases de datos, etc. el producto resultante puede ser algo que satisfaga todas las necesidades de programación en el entorno Windows. La suma de todas estas cosas es VISUAL - BASIC. Esta herramienta conserva del BASIC de los años 80 únicamente su nombre y su sencillez, y tras su lanzamiento al mercado, la aceptación a nivel profesional hizo borrar por fin el "mal nombre" asociado a la palabra BASIC.

Actualmente se está comercializando la versión 6.0 de este producto. Desde su salida al mercado, cada versión supera y mejora la anterior. Dados los buenos resultados a nivel profesional de este producto, y el apoyo prestado por el fabricante para la formación de programadores, Visual-Basic se ha convertido en la primera herramienta de desarrollo de aplicaciones en entorno Windows.

Es obligado decir sin embargo, que sigue siendo BASIC. No se pueden comparar sus prestaciones con otros lenguajes cuando deseamos llegar al fondo de la máquina y controlar uno a uno sus registros. No es ese el fin perseguido con VB y si es necesario llegar a esas precisiones será necesario utilizar otro lenguaje que permita bajar el nivel de programación. (Visual-C). o realizar librerías (DLLs) que lo hagan. En la mayor parte de las aplicaciones, las herramientas

aportadas por VB son mas que suficiente para lograr un programa fácil de realizar y de altas prestaciones.

CARACTERÍSTICAS GENERALES DE VISUAL-BASIC

Visual-Basic es una herramienta de diseño de aplicaciones para Windows, en las que estas se desarrollan en una gran parte a partir del diseño de una interface gráfica. En una aplicación Visual - Basic, el programa está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interfase gráfica.

Es por tanto un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos. Combina ambas tendencias. Ya que no podemos decir que VB pertenezca por completo a uno de esos dos tipos de programación, debemos inventar una palabra que la defina: PROGRAMACIÓN VISUAL.

La creación de un programa bajo Visual Basic lleva los siguientes pasos:

- Creación de un interface de usuario. Este interface será la principal vía de comunicación hombre máquina, tanto para salida de datos como para entrada. Será necesario partir de una ventana - Formulario - a la que le iremos añadiendo los controles necesarios.
- Definición de las propiedades de los controles - Objetos - que hayamos colocado en ese formulario. Estas propiedades determinarán la forma estática de los controles, es decir, como son los controles y para qué sirven.
- Generación del código asociado a los eventos que ocurran a estos objetos. A la respuesta a estos eventos (click, doble click, una tecla pulsada, etc.) le llamamos Procedimiento, y deberá generarse de acuerdo a las necesidades del programa.
- Generación del código del programa. Un programa puede hacerse solamente con la programación de los distintos procedimientos que acompañan a cada objeto. Sin embargo, VB ofrece la posibilidad de establecer un código de programa separado de estos eventos. Este código puede introducirse en unos bloques llamados Módulos, en otros bloques llamados Funciones, y otros llamados Procedimientos. Estos Procedimientos no responden a un evento acaecido a un objeto, sino que responden a un evento producido durante la ejecución del programa. No es

necesario entender de momento lo anterior. Visual Basic introduce un concepto nuevo de programación, y es necesario cambiar hasta el argot³¹ del programador.

VISUAL BASIC

Tras esta pequeña introducción al lenguaje Basic ya estamos en disposición de encender el ordenador y comenzar a trabajar con Visual-Basic. Se supone que su PC tiene instalado el programa Visual BASIC la versión de 32 bits si dispone de Win95 o superior ó Windows NT.

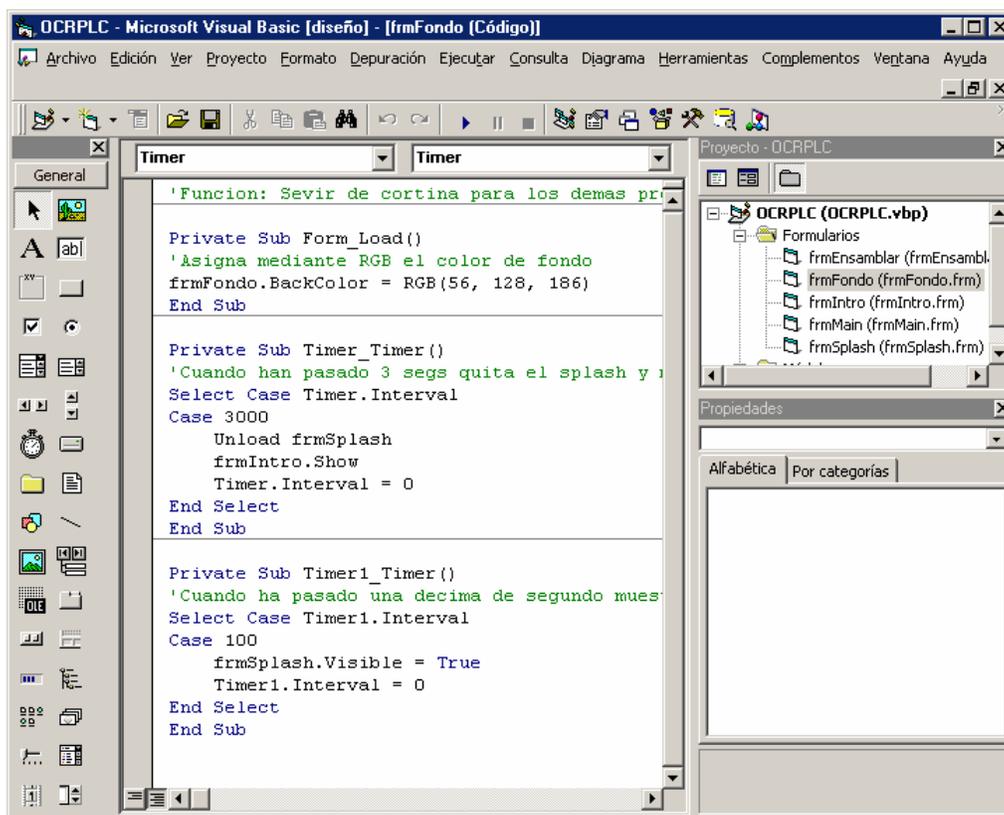


Figura 11: Pantalla principal de VB 6.0

Este es el comienzo del programa. Observe que en esta pantalla existen varias cosas. En la parte superior, la barra de título del programa Visual Basic, con el texto:

OCRPLC MICROSOFT VISUAL BASIC (DISEÑO)

Por debajo de esta barra de Título del VB, encontramos la barra de menú de VB, con las leyendas: Archivo, Edición, Ver, Insertar, Ejecutar, Herramientas, Complementos, Ayuda.

³¹ Se refiere al lenguaje en común que tienen grupos de personas

Por debajo de esta barra de menú tenemos la barra de herramientas, donde podrá ver varios iconos, que cada uno representa un determinada operación que Vd. puede realizar. Estas operaciones está todas en la línea de menú, y puede acceder a ellas abriendo los menús desplegables que existen en cada una de las palabras Archivo/ Edición/ Ver. en esta línea de menú. El hecho de colocar las operaciones más usuales en un icono en la barra de herramientas se hace para mayor comodidad del usuario.



Figura 12 Cuadro de Herramientas en VB 6.0A

la izquierda de la pantalla tiene una caja rectangular con tres columnas de iconos. Esa caja es laCaja de Herramientas (No la confunda con la barra de herramientas de la parte superior).

Esta caja de herramientas es, salvando las distancias, lo mismo que una caja de herramientas real que pueda tener un mecánico para realizar su trabajo. En ella están los iconos que representan a los controles con los que Vd. va a desarrollar su aplicación VB como se muestra en la figura 3-34. No están todos los que pueden estar, al igual que en una caja de herramientas de un mecánico no están todas las herramientas que puede usar. Están solamente las mas usuales. Si necesita alguna mas, puede cogerla de otra parte (barra de Menú, Herramientas, Controles

personalizados) y agregarlos a esa caja de herramientas, al igual que lo haría un mecánico con una herramienta especial, que la metería en su caja sólo cuando va a realizar un trabajo determinado para el que la necesite.

Posiblemente esta caja de herramientas no le aparezca tal y como la ve en estos apuntes. Eso depende si Ud. tiene la versión personal o la profesional, o si usa la versión de 16 ó 32 bits. Esa caja de herramientas puede personalizarla a su gusto. Para ello, deberá cargar un proyecto ya existente en los discos de distribución de Visual Basic, llamado AUTO32LD.VBP (versión 32 bits) ó AUTO16LD.VBP (Versión 16 bits) que se encontrará en el directorio donde haya cargado el programa Visual Basic. Este proyecto no tiene otra función que determinar las herramientas con las que quiere abrir el VB. Para introducir herramientas nuevas en la caja de herramientas, basta con desplegar el menú Herramientas de la barra de menú, e ir a Controles personalizados. Haciendo click en esta línea de controles personalizados, podemos seleccionar nuevos controles para añadir a la caja de herramientas, marcando la casilla que está a la izquierda del nombre del control que quiere introducir. Los controles comunes - CommandButton, Label, TextBox, etc.-, no se pueden eliminar de la caja de herramientas)

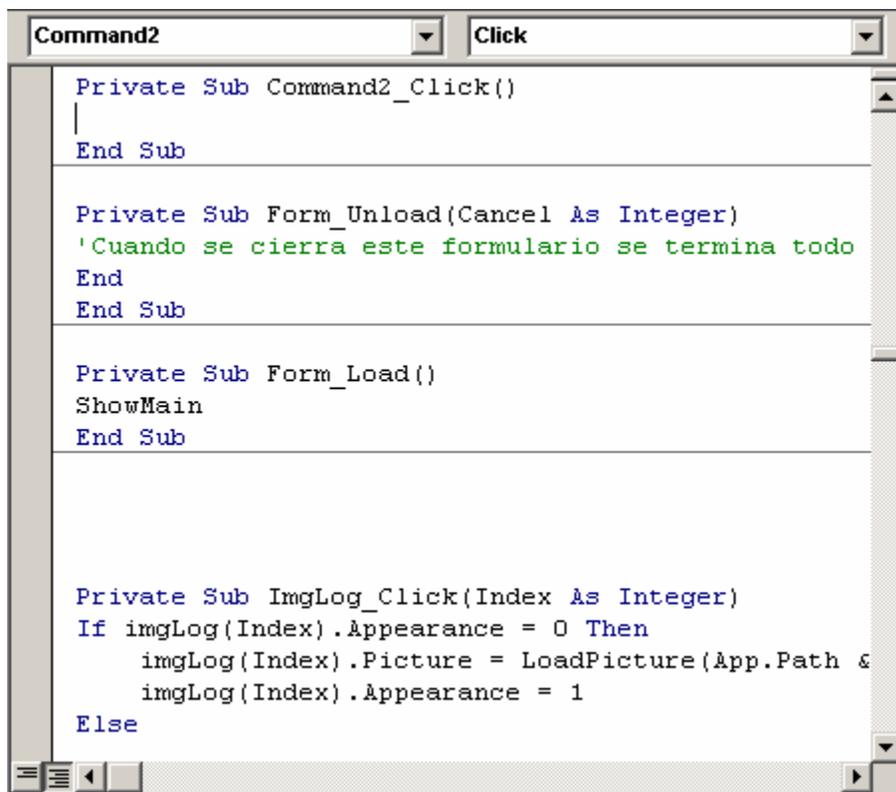
Para quitar controles de su caja de herramientas, debe proceder de forma análoga, cargando el programa AUTOxxLD.VBP, abriendo menú de herramientas, Controles personalizados, quitando la marca al cuadro situado en la parte izquierda del control a eliminar, y guardando el proyecto al salir.

En el centro, y ocupando casi toda la pantalla, tenemos el Formulario. Este Formulario es la interface gráfica de su aplicación, sobre el que podrá añadir los controles que necesite. Lo veremos mas adelante con todo detalle.

Observa dos ventanas, una denominada Ventana de Propiedades, donde puede ver las propiedades del formulario, Form1, y otra, denominada Ventana de Proyecto. Existe otra ventana, que no está en la figura anterior, la Ventana de Depuración. Por cada formulario y cada control que introduzca en el proyecto, le aparecerá otra ventana, denominada ventana de código, como se muestra en la figura 14.

No se extrañe de que esta presentación gráfica del Visual Basic coincida con otros sistemas de desarrollo (Delphi, p.e.). La lógica de desarrollo de una aplicación en Windows ha llevado a varios fabricantes de software a utilizar un entorno gráfico similar (diríamos idéntico). A Visual basic le queda el orgullo de ser el primero en utilizarlo.

Con lo descrito anteriormente ya tenemos, al menos, fijado el argot con el que expresarnos para comenzar a estudiar el VISUAL BASIC. Veamos con un poco mas detalle la Ventana de Código.



```
Private Sub Command2_Click()  
|  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
'Cuando se cierra este formulario se termina todo  
End  
End Sub  
  
Private Sub Form_Load()  
ShowMain  
End Sub  
  
Private Sub ImgLog_Click(Index As Integer)  
If imgLog(Index).Appearance = 0 Then  
    imgLog(Index).Picture = LoadPicture(App.Path &  
    imgLog(Index).Appearance = 1  
Else
```

Figura 13 Ventana de Código

Esta figura 13 se muestra un Formulario con su ventana de código. Cada objeto gráfico de VB tiene su propia ventana de código. Así, si en este formulario hubiésemos introducido un Label y dos CommandButton, todos ellos tendrían su propia ventana de código. La ventana de código la podemos ver haciendo doble click sobre cualquier objeto de nuestro proyecto. En este caso hemos hecho doble click sobre el único objeto que teníamos: el formulario.

Solamente nos queda por decir, para cerrar este capítulo, que es un procedimiento.

Para ello vamos a explicar lo que es un evento. Un Evento es algo que le puede ocurrir a un objeto. En una internase gráfica, lo que le puede ocurrir a un objeto es que se le haga click, doble click, que se pase el cursor del ratón por encima, etc. Este es el Evento. El Procedimiento es la respuesta por parte de ese objeto, al evento que le está sucediendo.

Esa respuesta, esa forma de Proceder del objeto al evento que le está sucediendo, debemos programarla según nuestras necesidades, es decir, debemos escribir el código que necesite

nuestra aplicación como respuesta al evento que acaba de ocurrir. Posiblemente, no queramos ninguna respuesta a muchos de los eventos que pueden acaecer a un objeto. Cada objeto tiene muchos eventos y solamente queremos aprovechar los que nos interesan. Para que un evento no produzca ningún efecto, basta con dejar sin código el procedimiento correspondiente a ese evento. En los eventos que queramos que realice alguna operación, le escribiremos en su procedimiento el código necesario para que esa operación se realice. Sin darnos cuenta, hemos comenzado a escribir el código de la aplicación.

Observará que el primer elemento del menú desplegable de la lista de objetos se denomina General. Este no es en realidad ningún objeto, sino un apartado existente en cada formulario, que, al desplegar su lista de procedimientos tiene la sección de declaraciones, donde debemos declarar las variables que queremos que afecten a todo el formulario y sus controles, y tendrá además, los nombres de todos los procedimientos que introduzcamos (véase un poco mas adelante). En este menú desplegable de la lista de procedimientos del General verá con frecuencia cosas que Vd. no puso allí. Cosas tales como `Command1_click`, y en la ventana un determinado código. Esto ocurre cuando se borra algún control que tenía escrito código en alguno de sus procedimientos. Visual Basic sabe lo mucho que cuesta escribir el código asociado a un control. Si borramos un control de nuestro formulario accidentalmente, después de haber introducido todo el código asociado a él, Visual Basic nos sorprende con que ese código no lo tira inmediatamente, sino que lo reserva como un procedimiento en ese apartado General del formulario. Si en realidad queríamos borrar el control y todo su código, debemos quitarlo de ese apartado General de nuestro formulario, pues en realidad, si no lo queremos, no hará otra cosa que estorbar. Para quitarlo basta con borrar todo el código que aparece en la ventana de código cuando hacemos click sobre el nombre del control eliminado. Debemos borrar todo el código, incluida la cabecera donde figura el nombre del control eliminado, y la parte final, que siempre termina con `EndSub`.

Una aplicación puede tener todo su código escrito en los sucesivos procedimientos del formulario y de los controles que tenga ese formulario.

Para insertar un procedimiento debe ir a la barra de menú, hacer click sobre Insertar, y en el menú que le desplegará, volver a hacer click sobre Procedimiento. VB le presentará un cuadro donde le pedirá el nombre, si quiere que sea un procedimiento, una función o una propiedad.

Escribiendo el código en los sucesivos procedimientos, bien en los propios de cada objeto, bien en los procedimientos que vayamos creando, es posible completar la aplicación. Pero en una aplicación larga esta forma de escribir el código no sería la más adecuada. Es más, posiblemente sería demasiado engorroso escribirla de esta forma, y muy probablemente deberíamos escribir el mismo código para varios procedimientos, lo que alargaría inútilmente el programa y el tiempo de desarrollo.

AMBIENTE LINUX: GTK+ 1.2V



³²GTK+ es la librería gráfica (toolkit) sobre la que se sustenta todo el interfaz gráfico de GNOME. Es una librería que contiene todo lo necesario para el desarrollo de interfaces gráficas, permitiendo la posibilidad de crear todo tipo de widgets, desde los más básicos, como botones, etiquetas, cajas de texto, hasta cosas más complejas como selectores de ficheros, colores, fuentes, cajas de texto multilínea con soporte para todo tipo de efectos sobre el texto, etc.

¿QUÉ ES UN WIDGET?

En términos de ingeniería del software, un widget es un componente software visible y personalizable. Visible porque está pensado para ser usado en los interfaces gráficos de los programas, y personalizable porque el programador puede cambiar muchas de sus propiedades. De esta forma se logra una gran reutilización del software, un objetivo prioritario en ingeniería del software. Los widgets se combinan para construir los interfaces gráficos de usuario. El programador los adapta según sus necesidades sin tener que escribir más código que el necesario para definir los nuevos valores de las propiedades de los widgets.

La librería GTK+ sigue el modelo de programación orientado a objetos. La jerarquía de objetos comienza en `GObject` de la librería Glib del que hereda `GtkObject`. Todos los widgets heredan

³² www.gtk.org

de la clase de objetos `GtkWidget`, que a su vez hereda directamente de `GtkObject`. La clase `GtkWidget` contiene las propiedades comunes a todos los widgets; cada widget particular le añade sus propias propiedades.

Los widgets se definen mediante punteros a una estructura `GtkWidget`. En GTK+, los widgets presentan una relación padre - hijo entre sí. Las aplicaciones suelen tener un widget "ventana" de nivel superior que no tiene padre, pero aparte de él, todos los widgets que se usen en una aplicación deberán tener un widget padre. Al widget padre se le denomina contenedor. El proceso de creación de un widget consta de dos pasos: el de creación propiamente y el de visualización. La función de creación de un widget tiene un nombre que sigue el esquema `gtk_nombre_new` donde "nombre" debe sustituirse por el nombre del widget que se desea crear. La función `gtk_widget_show` hará visible el widget creado.

La función de creación de un widget `gtk_nombre_new` devuelve un puntero a un objeto de tipo `GtkWidget` y no un puntero a un widget del tipo creado. Por ejemplo, la función `gtk_button_new` devuelve un puntero a un objeto de `GtkWidget` y no una referencia a un botón. Esta referencia puede convertirse a una referencia a un objeto `GtkButton` mediante la macro `GTK_BUTTON`, si se desea utilizar en lugares donde se requieran objetos botones. Aunque sería posible pasar en esos casos la referencia genérica, el compilador se quejará si se hace así posibilitando un control de tipos de objetos. Todo widget tiene una macro de conversión de una referencia genérica a una referencia al tipo propio. Eso sí, la macro únicamente funcionará correctamente si el widget referenciado fue creado con la función de creación del tipo apropiado, lo que incluye el propio tipo del widget o un descendiente.

El interfaz gráfico de una aplicación se construye combinando diferentes widgets (ventanas, cuadros combinados, cuadros de texto, botones, ...) y se establecen diversas retrollamadas (callbacks) para estos widgets, de esta forma se obtiene el procesamiento requerido por el programa a medida que se producen ciertas señales que a su vez provocan las retrollamadas. Las señales se producen por diversos sucesos como oprimir el botón de un ratón que se encuentra sobre un widget botón, pasar el cursor por encima de un widget u oprimir una tecla.

EJEMPLO: CONTENEDORES

GTK utiliza los contenedores para colocar los widgets de una forma determinada. Cuando se desarrolla una aplicación, normalmente se necesita colocar más de un widget dentro de una ventana.

En otros sistemas gráficos (como por ejemplo MS Windows), la colocación de los widgets dentro de las ventanas se hace por medio de coordenadas relativas. Esto hace necesario un nivel de detalle a la hora de diseñar las ventanas de las aplicaciones que lo hacen indeseable. En GTK+, al igual que en todos los toolkits gráficos provenientes del mundo UNIX (Motif, QT, GTK, AWT de Java), está basado en el modelo de contenedores, donde no es necesario el uso de coordenadas. Simplemente se crean distintos tipos de contenedores (cada uno de los cuales coloca los widgets dentro de si mismo de una forma determinada) para cada caso concreto, y simplemente se colocan widgets dentro de dichos contenedores. La forma en que se metan los widgets dentro del contenedor define cómo se comportarán dichos widgets cuando la ventana contenedora cambie de tamaño.

Usando estas funciones, GTK sabe en qué posición colocar los widgets y así poder cambiar el tamaño de los mismos automáticamente, cuando se cambia el tamaño del contenedor. Además cuentan con varias opciones para poder cambiar el estilo de colocación de los widgets. En la siguiente figura se muestran los 5 estilos diferentes de colocación.

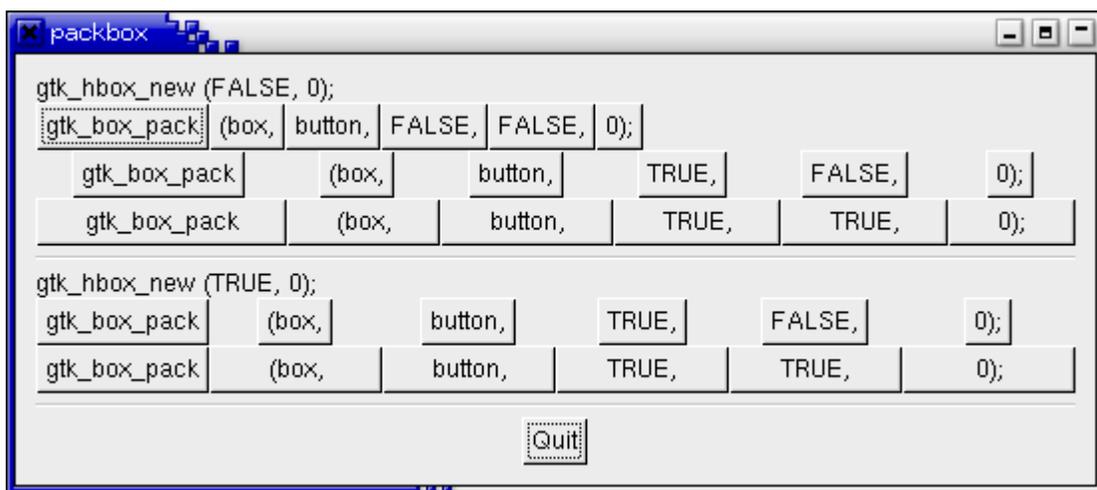


Figura 14: Ejemplo de Creación de una ventana

ANEXO 5

“ESTUDIO DE FACTIBILIDAD”

METODOLOGÍA GENERAL DEL ESTUDIO DE FACTIBILIDAD.

A continuación se presenta la metodología que se empleará en el Estudio de Factibilidad en una forma general, por lo que a su vez dicha metodología constará de una serie de etapas globales, con las cuales se busca obtener los resultados más óptimos.

PARTES	PASOS	TÉCNICA
Estudio de Mercado	Estudio de Mercado de Consumo. Estudio de Mercado Competidor. Estudio de Mercado de Abastecimiento.	Investigación de posibles usuarios del producto. Investigación de productos existentes. Investigación de los posibles proveedores de materiales directos , así como equipo especial necesario para la elaboración del producto.
Estudio Técnico	Ingeniería del proyecto	Investigar las características de los PLC's de la competencia. Investigar las tecnologías necesarias en la construcción del prototipo. Determinar las características de control y configuración que tendrá el prototipo.
Estudio Económico	Establecer cuales son los costos en los que se incurrirá en la elaboración del prototipo.	Cotizar precios de materiales directos a utilizar. Elaborar un informe de gastos que se incurrirán en la elaboración del prototipo, tales como compra o arrendamiento de equipo especial, otros.

Tabla 7 Metodología general para el desarrollo de los estudios a realizar

MERCADO DE CONSUMO

ANÁLISIS DE LA DEMANDA.

Debido a que no se cuenta con datos estadísticos previos sobre el tema del consumo de las empresas de los PLCs, el análisis de la demanda se basó únicamente en información obtenida a partir de fuentes primarias (encuestas) de productos actuales de la competencia que realizan la misma función y otras más avanzadas que el proyecto propuesto.

FUENTES PRIMARIAS: ENCUESTA.

Para la determinación del tamaño de la muestra para la investigación se realizó de la siguiente manera:

Utilizando la fórmula³³ :

$$n = \left[\frac{\sigma \times Z}{E} \right]^2$$

Ecuación 1

Donde : σ es la desviación estándar
 Z es el número de errores
 E es el error máximo permitido

El nivel de confianza deseado es del 95%, así como en la mayoría de las investigaciones. El valor de Z es entonces llamado número de errores estándar asociado con el nivel de confianza. Su valor se obtiene de la tabla de probabilidades de una distribución normal.

Para un nivel de confianza de 95% $Z = 1.96$

El error máximo permitido se interpreta como la mayor diferencia permitida entre la media de la muestra y la media de la población.

El investigador de mercado determina cuán alejada está la media poblacional de la media muestral para este caso consideramos un error del 25%.

³³Andrey & Richard p. Runyon, Estadística General

El valor de σ se determina por criterio, por referencia de otros estudios ó mediante una prueba piloto.

Para este análisis el valor de σ , se calculo por medio de una encuesta piloto, cuya finalidad fue sondear el numero de dispositivos que estaría dispuesto a comprar la persona o empresa sujeta de estudio.

Se encuestaron 10 personas, como se detalla a continuación:³⁴

Tabulación de los datos:

Número de compras	Cantidad de encuestas
0	2
1	7
2	1
Total:	10

Análisis Estadísticos de los datos:

Número de datos: $n = 10$

$$\text{Media : } \bar{X} = \frac{\sum x}{n} = 0.9 \quad \text{Ecuación}^{35} 2$$

$$\text{Desviación estándar: } \sigma_{\bar{X}} = \sqrt{\frac{\sum (X^2 - \bar{X})^2}{N(N-1)}} = 0.65 \quad \text{Ecuación 3}$$

Sustituyendo en la Ecuación 1, para la determinación del tamaño de la muestra

$$\text{obtenemos: } n = \frac{\sigma^2 \times Z^2}{E^2} = \frac{(0.65)^2 \times (1.96)^2}{(0.25)^2} = 27$$

Para este estudio se realizaron 25 encuestas.

³⁴Consultar Anexo A.1 Prueba Piloto.

³⁵ Andrey & Richard p. Runyon, Estadística General

LISTADO DE EMPRESAS ENCUESTADAS

Empresa	Zona
INCO	Final 1ª Av Norte Soyapango
PROMEDI	Barrio Lourdes, San Salvador
Friccionado Hernández	Frente Bodega Ofertas ADOC, Soyapango
Guzmán Hermanos	Avenida Peralta Frente ITEXSAL
MAG	Urbanización Bosques del Matazano Soyapango
Imprenta FIA	FIA, UES
Friccionado Atlacatl, Ramón Viera	San Miguelito, San Salvador
Súper Alto Clucth Andrés García R.	Apopa, San Salvador
Electro Técnicos Asociados	3ª Calle 6ta Avenida Norte San Salvador
Multi frenos Servicio	Boulevard Venezuela, San Salvador
Juan Carlos Alberto Rivera	Col. Sn. Antonio #102 San Salvador
Ingeniería Activa S.A.	Avenida las Amapolas Villas de San Francisco 2 pje #6 San Salvador
Planta de Torrefacción de Café	Boulevard del Ejercito, Soyapango
TELEFONICA	San Antonio Abad
Aseguradora Agrícola Salvadoreña S.A.	San Salvador
Energy International	Soyapango
GALVANISSA	San Salvador
Nejapa Power Planta	Apopa
AVX Corporation	Zona Franca Ilopango
Duke Energy International	Soyapango
ETESAL	Soyapango
Baterías de Q Salvador	Opico
Fabrica El Dorado (Unilever de C.A., S.A.)	Boulevard del Ejercito San Salvador
BIMBO	Ciudad Merliot Santa Tecla
MOLSA	Boulevard del Ejercito San Salvador

Tabla 8 Listado de las empresas encuestadas.

ANÁLISIS DE LOS RESULTADOS DE LA ENCUESTA:

Para nuestro este se realizaron 25 encuestas³⁶.

1. ¿En sus procesos de fabricación, que tipos de equipos tienen?

Resultados de la encuesta:

a) Resultados generales del los equipos:

Tipo de equipos	Encuestas:
Solo Lámparas	2
Motores – Lámparas	5
Motores – Lámparas –Compresores	11
Motores – Lámparas – compresores -Hornos	7
Total general	25

Tabla 9 Tipo de equipos

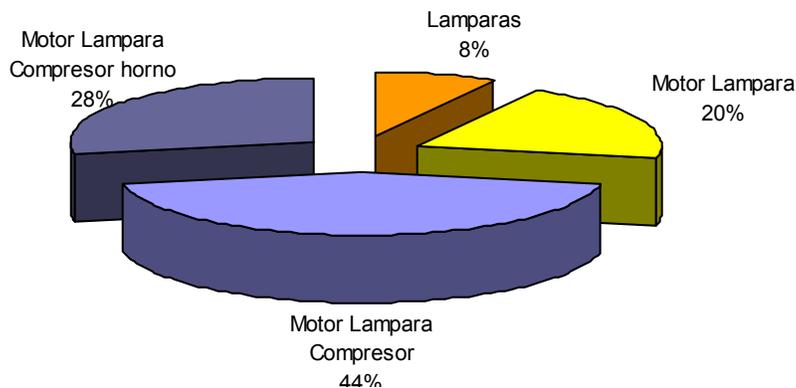


Figura 15 Resultados en porcentajes por tipo de quipos.

Descripción:

En la mayoría de las empresas de la industria, el 92% de ellas usan en sus procesos motores, lámparas, compresores y hornos, por lo que son un mercado potencial para los PLCs.

³⁶ Consultar Anexo A.2 Formato de Encuesta.

b) Resultados por su Alimentación, tipo de sistema y cantidad.

Resultados de la encuesta:

Variables		Resultados
Alimentación	AC	19
	DC	0
	Ambos	6
Sistema	Trifásico	8
	Monofásico	8
	Ambos	9
Cantidad	1-5	7
	6-15	7
	Mas	11
Total general		75

Tabla 10 Resultados por alimentación, sistema, cantidad

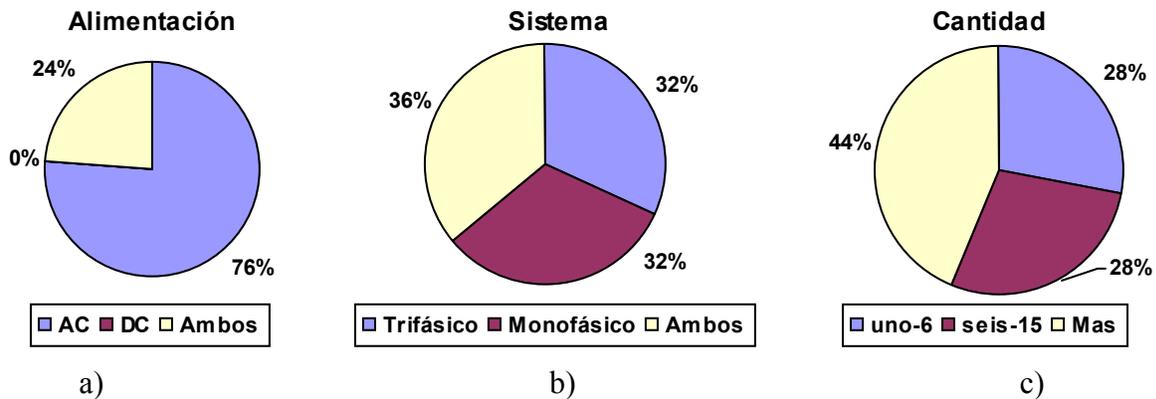


Figura 16 Tipo de alimentación, sistema y cantidad.

Descripción:

Alimentación: se puede decir que en un 80% el sistema es AC por lo que los PLC's deben de estar capacitados para estos niveles 120, 240, 480VAC.

Sistema: se observa que en la industria se usa comúnmente el sistema trifásico para reducción de pérdidas y aumentar la eficiencia de los motores.

Cantidad: En las empresas que ya se dedican a procesos grandes, el número de motores aumenta, pero también aumentan los costos provocados por: arranque mal sincronizado de motores, pérdidas por calentamiento, lámparas mal utilizadas en galeras grandes, por lo que un PLC que controle estas variables, puede ahorrar mucho dinero

2. ¿Qué tipo de control usan en esta empresa en los procesos de fabricación?

Resultados de la encuesta:

TIPOS DE CONTROL	Numero de encuestas
Manual	4
Semiautomático	3
Automático PLC	3
Manual-semiautomático	5
Manual-Automático PLC	1
Semiautomático-Automático PLC	2
Manual-Semiautomático-Automático PLC	7
Total general	25

Tabla 11 tipos de control frecuentemente usados por las empresas

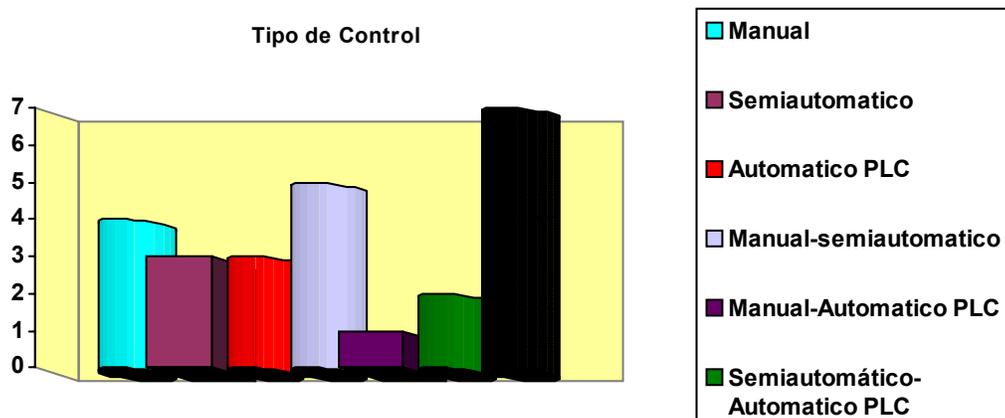


Figura 17 Resultados de los tipos de control mas usados

Descripción: Se logra observar que de las empresas encuestadas, por lo menos el 28% tienen todos los tipos de control encuestados; el 20% son manual y semiautomático y 16% son manuales, lo que muestra un mercado potencial para incorporar el PLC.

3. ¿Cuántos PLC's posee?

Resultados de la encuesta:

Cantidad de PLC's	Número de encuestas
Ninguno	13
1 a 2	4
3 a 5	2
Mas	6
Total general	25

Tabla 12 Numero de PLC que frecuentemente poseen las empresas

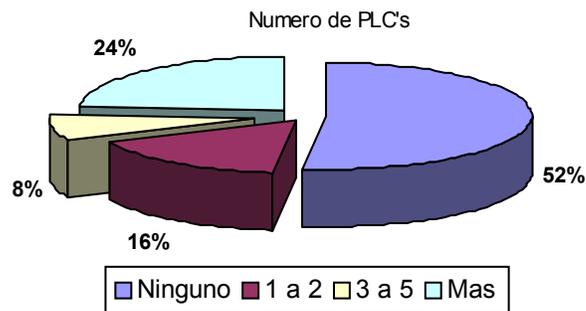


Figura 18 Resultados encontrados

Descripción: En el siguiente gráfico, se observa que el 52% de las empresas encuestadas no tienen un PLC para el control de sus procesos, por lo que los procesos son manuales y por ende no son óptimos, no pueden competir en un mercado exigente; por lo que el caso contrario en empresas un poco grandes tienen mas de 5 PLCs y las hacen de mayor capital y competitivas tanto nacional como internacional

4. ¿Qué marcas de PLCs tiene en su proceso de producción?

Resultados de la encuesta:

Marcas de PLC's	Cantidad de encuestas
SIEMENS	6
ABB	1
OMRON	1
ALLEN BRADLEY	2
CIRCUITOR	1
SIEMENS-OMRON	1
Total general	12

Tabla 13 Marcas de PLC's usados

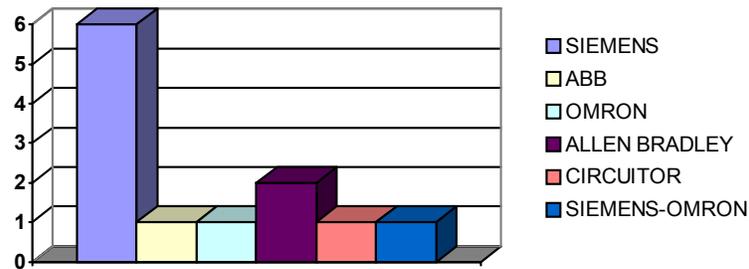


Figura 19 Resultados de los PLC's mas utilizados

Descripción

Observamos que entre las empresas que tienen PLC's, la marca mas difundida es SIEMENS con un 50%; también las marcas que se pueden destacar son OMRON y ALLEN BRAYLEY con un 25% de presencia en el mercado.

5. ¿Cuál es la forma de programación de su(s) PLC(s) ?

Resultados de la encuesta:

Forma de programación del PLC's	Número de encuestas
Manual	8
Pc.	4
Manual-Pc	1
Total general	13

Tabla 14 Formas de Programación de sus PLC's

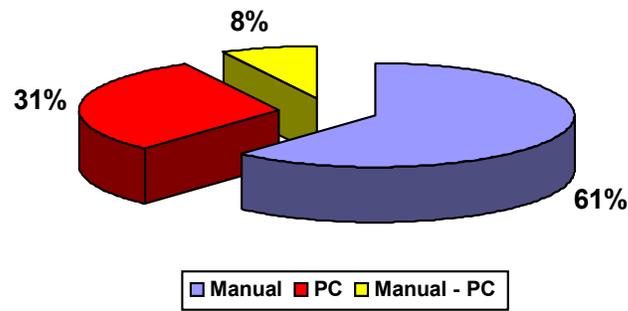


Figura 20 Resultados de los PLC's mas utilizados (en %)

Descripción: Se puede observar que en un 61% de los PLCs se programan de forma manual o teclado externo, lo cual le llaman “Insitu”, estos PLC's son los mas difundidos, porque son económicos; el 31% lo programan con PC y son mas complejos en su capacidad y comunicación y existen otros que tienen la ventaja de poder ser programados por PC y manualmente (8%).

6. ¿Qué Sistema Operativo utiliza para programar el PLC?

Descripción:

Según los datos anteriores en el mercado existe una mayor difusión del ambiente Windows de Microsoft, en cuanto a otros sistemas operativos, existe poca utilización.

7. Con relación al tamaño de su PLC, ¿Cuántas entradas y salidas tiene?

Descripción: Las empresas que tienen PLC's en sus procesos productivos, utilizan PLC de acuerdo al la función que desempeñen, de aquí depende el tamaño de este.

8. ¿Qué función(es) realiza(n) el(los) PLC's?

Resultados de la encuesta:

Función del PLC's	Número de encuestas
Control de temperatura	2
Control de motores	7
Temperatura-motores	1
Luz-temperatura-motores	2
Total general	12

Tabla 15 Aplicaciones mas comunes de los PLC's encontrados

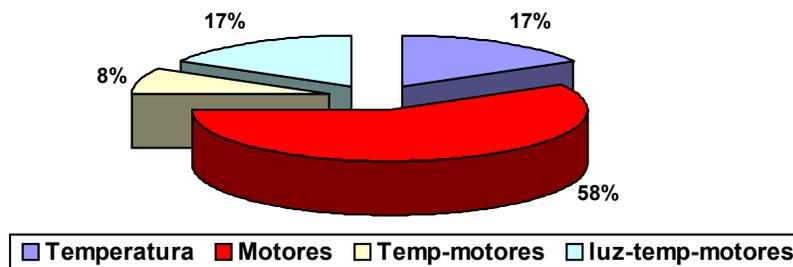


Figura 21 Aplicaciones mas comunes de los PLC's encontrados

Descripción: En la industria, la aplicación más utilizada es la que se refiere al control de motores, este dato se respalda con un 58%, las funciones van desde simple controles de arranque hasta complejos controles de velocidades y movimientos precisos; también, otras aplicaciones utilizadas son el control temperatura, presión, luz, etc. que en consolidado forman 34%. De las aplicaciones totales.

9. ¿En que forma recibe(n) el mantenimiento el(los) PLC's?

Descripción: Las empresas cuando compran los PLC's, en su mayoría el mantenimiento es interno para no pagar a la empresa distribuidora, esta práctica encarecería el proceso; otra practica es olvidarse del mantenimiento, y repararlo hasta que falle.

10. ¿Qué empresa le sirve el mantenimiento?

Resultados de la encuesta: no hay registro de nombre y direcciones de distribuidores.

11. ¿Qué lo motivó para comprar el(los) PLC's?

Los motivos son comunes, el cual es optimizar los procesos productivos para disminuir costos, por ende obtener mayor rentabilidad y calidad en los productos.

12. ¿A que empresa le compró el(los) PLC's?

Resultados de la encuesta:

EMPRESA	ENCUESTAS
SERVIELECTRIC	2
SIEMENS	3
INGENIERIA ELECTRONICA	1
MAK	1
WARTSILA DIESEL	1
GENERAL ELECTRIC	1
Total general	9

Tabla 16 empresa proveedora de el PLC

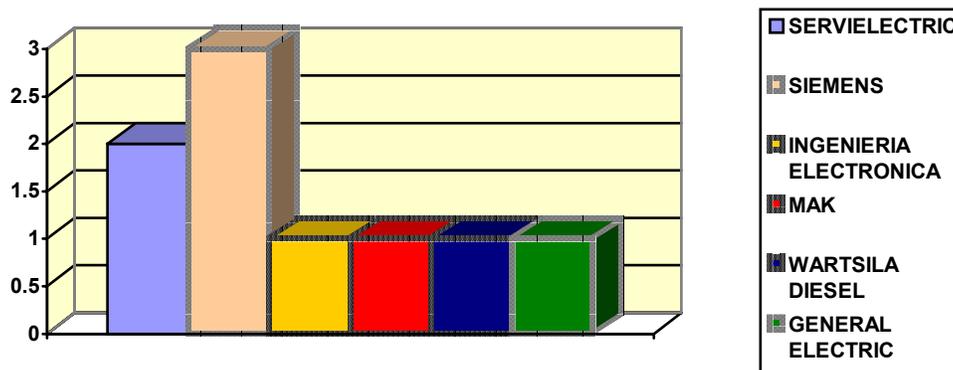


Figura 22 empresa proveedora de el PLC

Descripción: Uno de los mayores distribuidores en El Salvador, es SIEMENS; otro distribuidor importante es SERVIELECTRIC con el 20%.

13. ¿A través de que medio se enteró de la existencia de éste producto?

Descripción: Las empresas que desean ampliar y entrar en la era de la alta tecnología dedican un tiempo importante en asesorías técnicas, y otro en exposiciones y seminarios.

14. ¿El PLC que compró le presenta algún tipo de inconveniente?

Descripción: A las empresas les gusta comprar PLC's de buena calidad y garantías, por lo que la probabilidad de falla se reduce, mostrando eficiencia en los equipos adquiridos.

15. ¿Le interesaría controlar en forma automática sus procesos?

Resultados de la encuesta: El 100% sí desean controlar de forma automática los procesos.

16. ¿Que función desea que controle el PLC?

Resultados de la encuesta:

Función	Encuestas
Control de luces	1
Control de temperatura	2
Control de motores	10
Combinaciones entre ellas	12
Total general	25

Tabla 17 funciones que se desea que tenga el PLC

Descripción: La mayoría de empresas no le interesa el control de luces, lo ven como una pérdida de dinero por lo que solo el 4% los ocuparía para eso; la mayoría los emplearía para control de motores 40% que es donde esta la fuerza del proceso productivo y los mayores ahorros en automatización.

17. ¿Que precio estaría dispuesto a pagar por un PLC básico?

Resultados de la encuesta:

Rango de precios	Encuestas
\$200-400	12
\$401-600	8
\$601-800	5
Mas	1
Total general	26

Tabla 18 precio dispuesto a pagar por un PLC

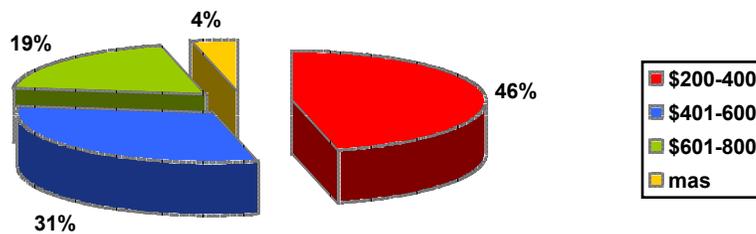


Figura 23 precio dispuesto a pagar por un PLC

Descripción

Según los datos anteriores se puede deducir que: un 46% quieren el más barato; por lo que otros buscan calidad y saben que eso implica pagar un poco más por eso son el 54%. Después se demostrará a como se debe de vender el PLC.

18. ¿Cuántos PLC's compraría?

Descripción:

Resultados de la encuesta: en la gran mayoría solo esta en capacidad de comprar un número limitado de 1 a 2 PLC's.

19. ¿Compraría un PLC básico ?

Descripción: Las empresas se muestran desconfiadas cuando se les habla de productos de alta tecnología fabricados por manos salvadoreñas, y no creen que sea de buena calidad, si no mas bien un producto que no reúne los estándares o normas internacionales, y el comprarlo corren el riesgo de incurrir en pérdidas.

MERCADO COMPETIDOR

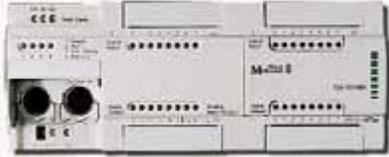
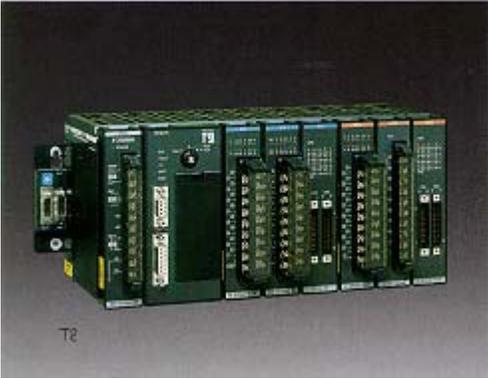
ANTECEDENTES:

En el país, la totalidad de los productos de alta tecnología para la industria de automatización son importados desde países como: EE.UU., México, Alemania, España, Japón, etc.. Las empresas productoras de este tipo de artículos tienen diversas líneas de productos, algunos abarcando también los accesorios, por ejemplo, la empresa SIEMENS produce desde sensores hasta los PLC de mas alta tecnología, abarcando así un mayor mercado. En el caso de los PLC's las empresas producen desde el mas barato y con funcionalidades básicas hasta los más caros pero con características avanzadas de automatización, lo cual invaden el mercado con novedades, pero ésta tecnología no está al alcance de todos, solo a los que la pueden pagar y darle el mantenimiento necesario.

Según el mercado a través de la wide world web, algunos productos que se pueden comprar son:

Fabricantes	Autómatas / Series	Características
ABB	<p style="text-align: center;">Advant Controller 31 - Series 90 Procontic</p> 	<p>Central unit 07 KT 95 y 07 KT 96 CS 31 - Field bus Program memory 480 kByte Flash EPROM and RAM Plug-in memory Smart Media Card (flash) Number of inputs/outputs Integral DI/DO 12 DI and 8 DO 24 DI and 14 DO DI/DO max. 1012 in total 1032 in total User program protection Password Data memory (bit) 8192, Words (16 bit) 8192, Double words (32 bit) 1024 Sequencers (of 16 sequences) 256, Timers, Counters, Fast counters 1 to 2 Special functions Protocols MODBUS® PI and PID controllers, ASCII communication, 32 bit arithmetic, Floating Point Arithmetic</p>

Fabricantes	Autómatas / Series	Características
Mitsubishi	<p>Serie FX</p> 	<p>La serie FX y FXo se configura desde 14 hasta 256 puntos de E/S. Alimentación: 120/240 V AC a 50-60 Hz. o 24 V DC, entradas de 24 V DC y 120 V AC, salidas tipo relevador, transistor y triac, módulos analógicos, memoria de 8K palabras.puede ser programado desde PC. Reloj de tiempo real</p>
Matsushita	 <p>FP0</p>	<p>Configuración Flexible desde 10 I/O hasta 128 I/O CPU ultra rápido con 0.9µs por ciclo de instrucción Gran capacidad con tamaño de memoria 2.7k, 5k y 10k Usa memoria para programa EEPROM or RAM Módulos analógicos con 8 canales de entrada o 2 entradas y 1 salida1. Puerto RS 232 para aplicaciones telemetricas</p>
Omron	<p>SYSMAC – C20H/C28H/C40H/C60H</p> 	<p>PLC con interface RS-232C incorporado y recursos configurables. Hasta 32 autómatas tipo H conectables directamente a un solo ordenador a través del interface RS-232C (entorno SYSWAY). El reloj en tiempo real Potencia e inteligencia en un formato pequeño dotado con 135 instrucciones y configurable a 240 puntos de E/S.</p>
Siemens	<p>LOGO! 230R \$120.00</p> 	<p>Requiere una alimentación de 115/120 ó 230/240 VCA. Frecuencia: 47 y 63 Hz Consumo: - típico 40 mA para 115/120 VCA. - típico 26 mA para 230/240 VCA. Potencia disipada: - típico 2,5 W para 115/120 VCA. - típico 3 W para 230/240 VCA. -posee 6 entradas digitales posee 4 salidas de relé de hasta 240 V -leva integradas las siguientes funciones básicas: Y, O, Y-NEGADA, INVERSO, O-NEGADO, O-EXCLUSIVO -lleva integradas las siguientes funciones especiales: - Retardo de activación - Retardo de desactivación - Relé de impulsos de corriente - Relé de automantenimiento - Generador de reloj - Retardo de activación almacenable - Contador progresivo/regresivo</p>

Fabricantes	Autómatas / Series	Características
GE-Fanuc	<p>GFK-0159J serie 90-70</p> 	<p>CPU de solo un slot para conexión. 512 entradas y/o salidas. Up to 8K analog I/O. 0.4 microsegundos por función. CPU 80C186. 12 MHz Soporta IC660/IC661) y productos IC697 I/O Programado en MS-DOS o Windows.</p>
Hitachi	<p>EH-150</p> 	<p>Comunicación óptica CPU's: 104, 208, 308, 316, 448 Convertidor 12bit y adaptable PT100/1000, 1024 I/O's, 2 puertos de comunicación serie ISO 9001 en ISO 14001 32 Bit-RISC-Microprocesador</p>
Moeller	<p>PS 4 Compact PLCs – Compact Control and Regulation</p> 	<p>Type PS 4-101-DD1 I/O81/60 Interfaces Suconet K1 Memory 3,6 KByte Ciclo de trabajo 5 µs Expandable 3 EM4</p>
Toshiba	<p>T2 PROGRAMMABLE CONTROLLER</p> 	<p>SFC y programación en escalera Procesador multirreas (programa principal y de interrupciones) Se conecta en red (a CPU's, I/O remotas, otros) Instrucciones parecidas a las de lenguaje de computadoras (Llamadas a Subrutinas, saltos, lazos For-Next, etc.) Protocolo de acceso a datos internos desde internet (FTP)</p>

Fabricantes	Autómatas / Series	Características
Unitronics	<p>M90 Micro – OPLCs \$400.00</p> 	<p>Manejo de entradas y salidas discretas clásicas de todo PLC. Manejo de entradas de alta velocidad de hasta 10 kHz. Manejo de salidas de alta velocidad (solo en algunos modelos) de hasta 5 KHz para control PWM. Instalable en el frente del Tablero en formatos estándar como el M90 que viene en formato de 1/4 DIN Reloj de tiempo real soportado por batería de litio. Uno o dos puertos seriales de entrada RS-232. Posibilidad de integrarlos en redes de hasta 63 OPLC's via CANBUS Drive DDE y OPC incluido en el CD del software para el desarrollo de interfaces a PC en Excell, Access o Visual Basic / C. Maneja entradas y salidas analógicas (-10 a 10 V, 0-5 V, 0/4-20 mA) Disponibles modelos con entrada para Termopares. Expandibles hasta 64 o 128 I/O según la familia. Incluye 4 u 8 lazos PID por OPLC según el modelo. Firmware actualizable para incorporar los futuros desarrollos del fabricante a los nuevos modelos. Software instalable en castellano windows. Series con posibilidad de configurar 2do. puerto serial a MODBUS. Acceso remoto vía modem externo convencional o via modem inalámbrico celular GSM pudiendo intercambiar mensajes en protocolo estándar SMS.</p>
Allen Bradley	<p>SLC 500 1746 \$350.00</p> 	<p>La 1746 plataforma es un hardware modular diseñado para usar con bajos costos y espacio reducido, con los que se pueden agregar módulos al sistema, pueden ser diseñados 4,8,16 y 32 puntos con interfaces AC, DC, y TTL. El software de programación RSLogix 500, puede ser ejecutado en las siguientes plataformas: Microsoft® 32-bit Windows 95® and Windows NT. Siguiendo sistemas de comunicación: Ethernet ControlNet DeviceNet DH+ DH 485 RS-232-C</p>

Fuente web: <http://eya.swin.net/>

Tabla 19 Algunos PLCs existentes en el mercado.

MERCADO DE ABASTECIMIENTO

PROVEEDORES DE MATERIAS PRIMAS Y PRECIOS:

Algunos de los proveedores de los elementos electrónicos comunes:

- Casa Rivas
- Electrónica 2001

Algunos de los proveedores de los elementos electrónicos especiales:

- Electrónicas J&R
- Thecnical Electronic Tools

La información referente a las condiciones de compra, períodos de entrega es la siguiente:

Proveedor	Localización	Condiciones	Período de entrega
Casa Rivas	Metrosur y Centro Comercial las Terrazas 29 calle oriente y 10 calle norte	20 % de descuento por compras > ¢ 50.00 No créditos	Cada 6 meses para grandes cantidades de materia prima
Electrónica 2001	11 av. Sur y calle Arce	20% de descuento por compras > ¢ 200	Información no disponible
Electrónicas J&R	Cumbres de Cuscatlan 3 ^a Etapa, Pje 1, casa #3 Antigua Cuscatlan, La Libertad	Pago Contra entrega, no descuentos	2 Semanas
Thenical Electronic Tools	Col. Flor Blanca 43 Av. Sur, y 4ta Calle Poniente #205 local 10	Pago Contra entrega, no descuentos	2 Semanas

Tabla 20 Cotización de los elementos y proveedores principales en el mercado

Cotización de los elementos más relevantes en las distribuidoras anteriormente mencionadas para construcción del prototipo del PLC. Usando las dos clases de tecnología usadas.

TECNOLOGÍA PIC16F8XX DE MICROCHIP

Componentes	Casa Rivas	Electrónica 2000	Electrónicas J&R	Thenical Electronic Tools	Cantidad	Precio menor cotizado
PIC16F877	NA	NA	19.5	27	1	19.5
STA401A		16.5	14.6	15.25	1	14.6
Dip Switch	1.6	1.5	1.85	7	2	3
LED	0.3	0.3	0.35	0.8	16	4.8
SN7405N	1.05	1.15	0.78	2.2	1	0.78
Tableta Cobreada	NA	NA	6.5	10	3	19.5
DB-25	NA	NA	1.5	1.6	1	1.5
Fuente Interna	25	30	30	NA	1	25
Switch on-off	0.75	0.75	0.8	0.8	2	1.5
FAN	3.35	2.5	5	14.5	1	2.5
Chasis	8.25	NA	NA	NA	1	8.25
DG201ACJ	3.25	5	NA	NA	1	3.25
74LS245	1.25	1.5	2	2	2	2.5
CD4010	1.8	2	2.5	2	1	1.8
CD40109	1.8	2	2.5	2	1	1.8
BC549C	1.5	1.5	2	2.25	3	4.5
BYV10	0.5	0.5	0.5	0.50	1	0.5
Conectores de Bus	NA	NA	6	NA	10	60
buzzer	0.6	0.6	1	1	1	0.6
R2.2K	0.11	0.11	0.11	0.11	12	1.32
R10K	0.11	0.11	0.11	0.11	24	2.64
BC557	1.5	1.5	2	2.25	3	4.5
Articulos varios						10
Total estimado del prototipo:						\$194.34

Tabla 21 Insumos necesario utilizando la tecnología PIC16F877 de Microchip.

Nota: Todos los precios US\$

TECNOLOGÍA CPLD DE XILINX.

Componentes	Casa Rivas	Electrónica 2000	Electrónicas J&R	Thenical Electronic Tools	Cantidad	Precio menor cotizado
XC9572	NA	NA	25	29.5	1	25
STA401A		16.5	14.6	15.25	1	14.6
Dip Switch	1.6	1.5	1.85	7	2	3
LED	0.3	0.3	0.35	0.8	16	4.8
TDA-8764ATS	10.5	13	15	13	1	10.5
Tableta Cobreada	NA	NA	6.5	10	3	19.5
DB-25	NA	NA	1.5	1.6	1	1.5
Fuente Interna	25	30	30	NA	1	25
Switch on-off	0.75	0.75	0.8	0.8	2	1.5
FAN	3.35	2.5	5	14.5	1	2.5
Chasis	8.25	NA	NA	NA	1	8.25
DG201ACJ	3.25	5	NA	NA	2	6.5
R300	0.11	0.11	0.11	0.11	15	1.65
CD4010	1.8	2	2.5	2	1	1.8
CD40109	1.8	2	2.5	2	1	1.8
74HC175	2.5	3	2.75	2.00	2	4
1N5817	0.34	0.4	0.46	0.46	4	1.36
Conectores de Bus	NA	NA	6	NA	10	60
buzzer	0.6	0.6	1	1	1	0.6
R2.2K	0.11	0.11	0.11	0.11	12	1.32
R10K	0.11	0.11	0.11	0.11	24	2.64
Articulos varios						10
Total estimado del prototipo:						\$207.82

Tabla 22 insumos necesario utilizando la tecnología CPLD DE XILINX.

Nota: Todos los precios US\$.

De acuerdo con lo anterior, al evaluar los costos del material directo se dice que la construcción del prototipo para el PLC, es mas viable usando la TECNOLOGÍA PIC16F8XX DE MICROCHIP. Existiendo una diferencia neta de \$15.48, de acuerdo a cotización, con respecto al uso de la tecnología CPLD DE XILINX.

INGENIERÍA DEL PROYECTO

DISEÑO TÉCNICO DEL PLC

En la tabla 23, se realiza una comparación entre el CPLD de Xilinx y el PIC de Microchip, para determinar con cual dispositivo se debe de construir el PLC.

CASO	TECNOLOGÍA	
	PIC de Microchip	CPLD de Xilinx
Precio	US\$194.34	US\$207.82
Interfaz de programación	De fácil construcción (DIP)	Montaje complejo (PLCC)
Facilidad de construcción	Dispositivos existentes en el mercado nacional	No existentes localmente
Periféricos adicionales	No necesita periféricos adicionales	Se debe de colocar uno o varios ADC's, temporizadores externos.
Software de programación	Gratis	Con licencia
Lenguaje de programación	Solamente 35 instrucciones	Gran variedad de instrucciones
Manejo del tiempo real	Maneja fácilmente tiempo real	No maneja tiempo real porque no atiende interrupciones

Tabla 23 Tabla de comparación de CPLD y PIC

Según la tabla 23, el mas optimo para nuestro PLC construido artesanalmente es la tecnología de Microchip.

ESTUDIO ECONÓMICO

INVERSIONES DEL PROYECTO

En éste estudio se limita, a proponer los costo de construcción del prototipo, que involucra: costo de materia prima, mano de obra y gastos indirectos de fabricación, obviando la depreciación del equipo a utilizar que es propiedad de la Escuela de Ingeniería Eléctrica de la UES, se basá en un sistemas de costeo por ordenes, (debido a que para éste caso solo se elaboró un prototipo de PLC), más, sin embargo si se proyectara la idea de elaborarlos en serie , se debe introducir el sistema de costeo de procesos , que disminuiría los costos de producción porque el estándar ya se tendría establecido.

CONSIDERACIONES NECESARIAS:

Para ensamblar un circuito prototipo, se tomaron las siguientes consideraciones:

- ❑ Los costos de materias primas son los evaluados en las distribuidoras que tienen los precios más bajos del mercado.
- ❑ No se tomará como costo el rubro que involucra el diseño y elaboración de los diferentes estudios de factibilidad.
- ❑ Se supondrá que se poseen los estándares del hardware y el software correspondiente, aunque este ultimo podrá ser sujeto a modificación por parte de los desarrolladores.
- ❑ Se define como inversión a los costos directos de materia prima y equipo, mano de obra directa y CIF (variables y fijos).

a) Equipo para producción

Equipo para producción			
Descripción	Cantidad	P.U	Total
Pinza	2	\$4	8
Succionadora manual de estaño	1	\$17.14	\$17.14
Cautín	2	\$17.14	34.28
Cortadora	2	\$8.57	17.14
Multímetro	1	\$85.71	85.71
		Total	\$162.27

Tabla 24 Equipo necesario como inversión de Activo Fijo

b) Materias primas propuestas

CUADRO COMPARATIVO MATERIAL DIRECTO

TECNOLOGÍA PIC16F8XX DE MICROCHIP		TECNOLOGÍA CPLD DE XILINX.	
Componentes	Precio menor cotizado	Componentes	Precio menor cotizado
PIC16F877	19.5	XC9572	25
STA401A	14.6	STA401A	14.6
Dip Switch	3	Dip Switch	3
LED	4.8	LED	4.8
SN7405N	0.78	TDA-8764ATS	10.5
Tableta Cobreada	19.5	Tableta Cobreada	19.5
DB-25	1.5	DB-25	1.5
Fuente Interna	25	Fuente Interna	25
Switch on-off	1.5	Switch on-off	1.5
FAN	2.5	FAN	2.5
Chasis	8.25	Chasis	8.25
DG201ACJ	3.25	DG201ACJ	6.5
74LS245	2.5	R300	1.65
CD4010	1.8	CD4010	1.8
CD40109	1.8	CD40109	1.8
BC549C	4.5	74HC175	4
BYV10	0.5	1N5817	1.36
Conectores de Bus	60	Conectores de Bus	60
buzzer	0.6	Buzzer	0.6
R2.2K	1.32	R2.2K	1.32
R10K	2.64	R10K	2.64
BC557	4.5	Articulos varios	10
Articulos varios	10	Total estimado del prototipo: \$207.82	
Total estimado del prototipo: \$194.34			

*Todas las cantidades se encuentran en US\$

Tabla 25 Comparación de costos de insumos , utilizando las dos tecnologías

c) Mano de obra directa: Se proyecta para la elaboración del PLC , 40 horas semanales a un precio de \$1.25 , haciendo un total de \$50 dólares semanales.

d) Costos Indirectos de Fabricación: incluimos gastos de operación y venta para la elaboración del PLC.

Costos Variables:

- Gastos de energía Eléctrica No aplica
- Papelería \$5 por Orden
- Gastos de envió \$3 Por Orden
- Viáticos y otros \$5 Por Orden.

HOJAS DE COSTOS ESTÁNDAR POR UNIDAD.

TECNOLOGÍA PIC16F8XX DE MICROCHIP		TECNOLOGÍA CPLD DE XILINX.	
MATERIAL DIRECTO	\$194.34	MATERIAL DIRECTO	\$207.82
MANO DE OBRA DIRECTA 40*\$ 1.25 =	\$50.00	MANO DE OBRA DIRECTA	\$50.00
GASTOS INDIRECTOS DE FABRICACIÓN	\$13.00	GASTOS INDIRECTOS DE FABRICACIÓN	\$13.00
INVERSIÓN EN MAQUINARIA Y EQUIPO	\$162.27		\$ 162.27
TOTALES	\$419.61	TOTALES	\$433.09

Tabla 26 Hoja de costo estándar

Al evaluar los costos generales para la elaboración de un PLC, usando dos clases de tecnologías, podemos determinar que el mismo es más accesible si se construye con base en el PIC16F8XX DE MICROCHIP, en comparación con el CPLD DE XILINX, aunque habría que evaluar las ventajas funcionales derivados de la utilización de cada uno de ellos.