

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

CONSTRUCCIÓN DE UN CONTROLADOR LÓGICO PROGRAMABLE BÁSICO, CON INTERFACE USB UTILIZANDO EL PIC 18F4550

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTRÓNICA Y TELECOMUNICACIONES
ELECTROMECAÁNICA

ROCHA SALDAÑA CAROL ELIZABETH
k_ro501@hotmail.com

TAYUPANTA TIGASI JORGE DAVID
jorgepearce_64@hotmail.com

DIRECTOR: ING. CARLOS ORLANDO ROMO HERRERA
carlos.romo@epn.edu.ec

Quito, julio 2016

DECLARACIÓN

Nosotros, Carol Elizabeth Rocha Saldaña y Jorge David Tayupanta Tigasi, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Carol Elizabeth Rocha Saldaña

Jorge David Tayupanta Tigasi

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Carol Elizabeth Rocha Saldaña y Jorge David Tayupanta Tigasi, bajo mi supervisión.

Ing. Carlos Romo
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

A Dios porque sin él no se puede realizar nada, por brindarme salud y fuerza en este largo camino.

A mis padres Ricardo y Gladys, que me dieron el empuje necesario para seguir adelante cuando más lo necesitaba. Gracias por su amor, comprensión y por saber guiarme en cada paso de mi vida.

A mi hermana María Belén, quien siempre me ha apoyado y es mi inspiración. Gracias Puppy.

Por último a mi mejor amiga Gaby Gavilánez, que también me ha apoyado mucho en los momentos más difíciles.

Carol Rocha

Le agradezco a Dios por haberme guiado a lo largo de mi carrera, por haberme dado fuerzas en los momentos de debilidad y por darme una vida llena de experiencias y aprendizajes.

Le doy gracias a mis padres David y Luz María por todo el apoyo brindado, los buenos valores que me han enseñado y por haberme dado la oportunidad de tener una buena educación durante toda mi vida; han sido un gran ejemplo de lucha y esfuerzo.

A Santiago, mi mejor amigo, que siempre me estuvo dando fuerza y ánimo en todo momento a lo largo de mi carrera.

Al Ingeniero Carlos Romo por creer en Carol y en mí, y habernos brindado la oportunidad de desarrollar nuestra tesis profesional.

Al Ingeniero Carlos Miranda quien supo apoyarme y darme palabras de aliento para poder culminar la tesis. Por darme la oportunidad de crecer profesionalmente y aprender cosas nuevas.

A Carol por haber sido una excelente compañera de tesis y una gran amiga, por haberme tenido la paciencia necesaria y motivarme a seguir adelante en todo momento.

Jorge Tayupanta

DEDICATORIA

Todo el esfuerzo, trabajo y lucha por realizar este proyecto se lo dedico a toda mi familia

A mi papá Ricardo Rocha porque siempre creyó en mí y me apoya en todas las decisiones de mi vida como ningún papá lo hace.

A mi mamá Gladys Saldaña, porque sin su empuje y determinación esta tesis no existiría. Gracias por darme ánimos y fuerzas en todo lo que hago.

A mi hermana María Belén, quien es mi mejor amiga. Tú más que nadie sabe lo que significa esto, haz estado junto a mí dándome apoyo y ganas de seguir adelante. Gracias por ser la razón de que yo quiera ser alguien mejor.

Carol Rocha

Esta tesis se la dedico a mi Dios quién supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban.

A mis padres por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles, y por ayudarme con los recursos necesarios para estudiar. Me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos.

A mi hermana por estar siempre presente, en cada momento de mi vida estudiantil. A mi compañera de tesis Carol Rocha quien ha sido un gran apoyo durante la realización de este proyecto y me demostrado que podemos ser grandes amigos y compañeros de trabajo a la vez.

Jorge Tayupanta

CONTENIDO

INTRODUCCIÓN.....	x
RESUMEN	xi
CAPÍTULO I	1
CONTROLADORES LÓGICOS PROGRAMABLES	1
1.1 HISTORIA DEL PLC.....	1
1.1.1 PLC'S LOGO.....	2
1.1.1.1 Software de programación de LOGO	3
1.1.1.2 Funciones básicas LOGO	4
1.1.1.3 Funciones especiales LOGO.....	4
1.2 PRINCIPIOS BÁSICOS DEL PLC.....	5
1.2.1 AUTOMATIZACIÓN	5
1.2.1.1 Sistema de control.....	6
1.2.1.1.1 Parte operativa.....	6
1.2.1.1.2 Parte de mando.....	7
1.2.1.2 Funciones básicas del PLC	7
1.2.1.2.1 Detección	7
1.2.1.2.2 Mando	7
1.2.1.2.3 Dialogo hombre - máquina	8
1.2.1.2.4 Programación	8
1.2.1.2.5 Redes de comunicación.....	8
1.2.1.2.6 Sistemas de supervisión	8
1.2.1.2.7 Entradas/salidas distribuidas	8
1.2.1.2.8 Buses de campo	8
1.2.1.3 Campos de aplicación.....	8
1.2.2 DESCRIPCIÓN Y FUNCIONAMIENTO DEL PLC	9
1.2.2.1 Descripción del PLC	9
1.2.2.2 Funcionamiento del PLC.....	9
1.2.2.2.1 Ciclo de funcionamiento	10
1.2.2.2.2 Tiempo de ejecución en tiempo real.....	11
1.2.3 ARQUITECTURA INTERNA DEL PLC.....	11
1.2.3.1 CPU	12

1.2.3.2 Módulos entrada/salida	12
1.2.3.3 Módulos entrada digital	13
1.2.3.3.1 Módulos entrada analógica.....	13
1.2.3.3.2 Módulos salida digital	13
1.2.3.3.3 Módulos salida analógica	13
1.2.3.4 Fuente de alimentación.....	14
1.2.3.5 Memorias	14
1.2.3.6 Consola de programación	15
1.2.3.7 Dispositivos periféricos.	15
1.3 CLASIFICACIÓN DEL PLC	15
1.3.1 PLC NANO.....	15
1.3.2 PLC COMPACTO.....	16
1.3.3 PLC MODULAR.....	17
1.4 VENTAJAS Y DESVENTAJAS DEL PLC	17
1.4.1 VENTAJAS	17
1.4.2 DESVENTAJAS.....	18
1.5 LENGUAJES DE PROGRAMACIÓN PARA EL PLC.....	18
1.5.1 CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN	19
1.6 ESPECIFICACIONES TÉCNICAS.....	22
1.6.1 ESPECIFICACIONES MECÁNICAS.....	22
1.6.1 ESPECIFICACIONES ELECTRICAS.....	22
CAPÍTULO II	24
MICROCONTROLADOR E INTERFAZ USB	24
2.1 MICROCONTROLADOR.....	24
2.1.1 INTRODUCCIÓN	24
2.1.2 CARACTERÍSTICAS.....	25
2.1.2.1 Capacidad de operación en paralelo.....	25
2.1.2.2 Capacidad de memoria	25
2.1.2.2.1 Memoria de datos.....	25
2.1.2.2.2 Memoria de instrucciones.....	25
2.1.2.2.3 ICSP.....	26
2.1.2.3 Puertos de E/S	26
2.1.2.4 Juego de instrucciones	27

2.1.2.5	Bajo consumo	27
2.1.2.6	Protección de los programas frente a copias.....	27
2.1.3	ARQUITECTURA DE LOS MICROCONTROLADORES	28
2.1.3.1	Arquitectura Von Neumann	28
2.1.3.1	Arquitectura Harvard	29
2.2	MICROCONTROLADORES PIC	29
2.2.1	CLASIFICACIÓN DE LOS MICROCONTROLADORES PIC	30
2.2.2	CARACTERÍSTICAS DE LOS MICROCONTROLADORES PIC.....	30
2.2.3	ARQUITECTURA INTERNA DEL PIC.....	31
2.2.3.1	Recursos fundamentales	31
2.2.3.1.1	Procesador	31
2.2.3.1.2	Memorias	32
2.2.3.1.3	Reloj principal.....	33
2.2.3.2	Recursos especiales.....	35
2.2.3.2.1	Temporizadores	35
2.2.3.2.2	Perro guardián.....	35
2.2.3.2.3	Estado de reposo o bajo consumo	35
2.2.3.2.4	Convertidor A/D y D/A	36
2.2.3.2.5	Comparadores analógicos	36
2.2.3.2.6	Puertos digitales de E/S.....	36
2.3	HERRAMIENTAS PARA EL DESARROLLO DE LAS APLICACIONES.....	37
2.3.1	HERRAMIENTAS DE HARDWARE	37
2.3.1.1	Placas de prototipos.....	37
2.3.1.2	Grabadores o programadores.....	38
2.3.1.3	Depuradores en el sistema.....	38
2.3.2	HERRAMIENTAS DE SOFTWARE	38
2.3.2.1	Editores y compiladores.....	38
2.3.2.2	Simuladores.....	39
2.4	INTERFACES	39
2.4.1	PUERTO PARALELO.....	39
2.4.2	PUERTO SERIE.....	39
2.4.3	BUS UNIVERSAL EN SERIE.....	39
2.4.3.1	Características de transmisión	40

2.4.3.2 Velocidad de transmisión.....	41
2.4.3.3 Tipos de conectores	41
CAPÍTULO III	43
IMPLEMENTACIÓN DEL CONTROLADOR LÓGICO PROGRAMABLE	43
3.1 INTRODUCCIÓN	43
3.2 ESPECIFICACIONES DE LOS REQUISITOS PARA LA CONSTRUCCIÓN DEL PLC.....	43
3.2.1 REQUERIMIENTOS DEL SISTEMA.....	43
3.3 DIAGRAMA DE BLOQUES	44
3.4 SELECCIÓN DE LOS COMPONENTES.....	44
3.4.1 MICROCONTROLADOR 18F4550	44
3.4.1.1 Diagrama de Pines.....	46
3.4.1.2 Memorias	47
3.4.2 TFT LCD.....	48
3.4.2.1 TFT Arduino	48
3.4.3 RELÉ	49
3.4.4 OPTOACOPLADORES	51
3.4.5 CONVERTIDOR USB UART RS232RL.....	52
3.4.6 FUENTE DE ALIMENTACIÓN	54
3.4.6.1 Transformador de entrada.....	54
3.4.6.2 Rectificador de diodos.....	54
3.4.6.3 Filtro	55
3.4.6.4 Regulador de voltaje	55
3.5 DISEÑO Y CONSTRUCCIÓN DEL PLC.....	56
3.5.1 DISEÑO DEL PLC.....	56
3.5.2 CONSTRUCCIÓN DEL PLC.....	57
3.6 DISEÑO DEL SOFTWARE	62
3.6.1 DESCRIPCIÓN DEL SOFTWARE.....	62
3.6.1.1 Parte de control.....	63
3.6.1.2 Parte de comunicación.....	64
3.6.2 DIAGRAMA DE FLUJO DEL SOFTWARE	66
3.6.3 INSTRUCCIONES Y VARIABLES	67
3.6.3.1 Variables.....	67
3.6.3.2 Instrucciones	67

CAPÍTULO IV	68
PRUEBAS EXPERIMENTALES	68
4.1 DESCRIPCIÓN FÍSICA.....	68
4.2 PRUEBAS EXPERIMENTALES.....	68
4.2.1 PRUEBAS DE LAS FUNCIONES LÓGICAS.....	68
4.2.1.1 Función and.....	68
4.2.1.2 Función or	70
4.2.1.3 Función not	71
4.2.1.3 Función xor.....	73
4.2.2 PRUEBA GENERAL DEL SISTEMA	75
4.2.2.1 Armado.....	75
4.2.2.2 Programación	76
4.3 ANÁLISIS TÉCNICO Y ECONÓMICO	77
4.4 ALCANCES Y LIMITACIONES.....	78
4.4.1 ALCANCES.....	78
4.4.2 LIMITACIONES	79
CONCLUSIONES Y RECOMENDACIONES.....	80
CONCLUSIONES	80
RECOMENDACIONES	80
REFERENCIA BIBLIOGRÁFICAS	81
ANEXOS	84
ANEXO 1:	85
SOFTWARE DE PIC 18F4550	85
ANEXO 2:	94
DIAGRAMA DE FLUJO PIC 18F4550	94
ANEXO 3:	96
SOFTWARE INTERFAZ USUARIO	96
ANEXO 4:	111
SOFTWARE TFT.....	111
ANEXO 5:	120
DISEÑO DEL CIRCUITO.....	120
ANEXO 6:	122
PLACA PLC	122

ANEXO 7: 124
DATA SHEETS..... 124

ÍNDICE DE FIGURAS

Figura 1.1 MODICON 084	2
Figura 1.2 Logo Siemens.....	3
Figura 1.3 Esquema del sistema de control	7
Figura 1.4 Arquitectura interna PLC.....	12
Figura 1.5 PLC Tipo Nano.....	16
Figura 1.6 Estructura interna PLC.....	16
Figura 1.7 PLC Tipo Compacto.....	17
Figura 1.8 PLC Tipo Modular	17
Figura 1.9 Lenguaje de programación	20
Figura 2.1 Microcontrolador	24
Figura 2.2 Arquitectura Von Neuman	28
Figura 2.3 Arquitectura Harvard	29
Figura 2.4 Arquitectura interna del PIC.....	31
Figura 2.5 Oscilador RC.....	33
Figura 2.6 Oscilador XT	34
Figura 2.7 SiDePIC USB	37
Figura 2.8 Conector USB tipo A	41
Figura 2.9 Conector USB tipo B	42
Figura 2.10 Conector mini USB.....	42
Figura 2.11 Conector micro USB	42
Figura 3.1 Diagrama de funcionamiento.....	44
Figura 3.2 Diagrama pines PIC 18f4550.....	46
Figura 3.3 Diagrama pines TFT Arduino	49
Figura 3.4 Simbología del relé	50
Figura 3.5 Diagrama interno optoacoplador	51
Figura 3.6 Diagrama esquemático del integrado FT232	53
Figura 3.7 Diagrama fuente de alimentación.....	55
Figura 3.8 Diagrama de flujo PLC	56
Figura 3.9 Diagrama entradas digitales.....	57
Figura 3.10 Diagrama salidas digitales	58
Figura 3.11 Diagrama comunicación USB.....	60
Figura 3.12 Placa PLC.....	61
Figura 3.13 Funciones especiales	632
Figura 3.14 Cuadro de control.....	63
Figura 3.15 Cuadro de comunicación.....	65
Figura 3.16 Interfaz del PLC.....	65
Figura 3.17 Diagrama de flujo del software	66
Figura 4.1 Programación función and	68
Figura 4.2 Resultado de la función and	69
Figura 4.3 Programación función or	70
Figura 4.4 Resultado función or	71
Figura 4.5 Programación función not.....	72
Figura 4.6 Resultado de la función not.....	73

Figura 4.7 Programación función xor	73
Figura 4.8 Resultado función xor.....	74
Figura 4.9 Circuito armado semáforo.....	76
Figura 4.10 Programación	76

ÍNDICE DE TABLAS

Tabla 1.1 Lenguajes de programación PLC.....	19
Tabla 1.2 Ejemplo lenguaje booleano	20
Tabla 1.3 Especificaciones mecánicas PLC	22
Tabla 1.4 Especificaciones eléctricas PLC.....	23
Tabla 2.1 Tabla de cristal oscilador y capacitores	34
Tabla 2.2 Características de transmisión USB	40
Tabla 3.1 Características PIC 18f4550 y PIC 16f887	45
Tabla 3.2 Características eléctricas PLC	61
Tabla 4.1 Tabla lógica función and.....	69
Tabla 4.2 Tabla lógica función or	71
Tabla 4.3 Tabla lógica función not	72
Tabla 4.4 Tabla lógica función xor.....	74
Tabla 4.5 Costos PLC.....	78

INTRODUCCIÓN

Actualmente existen microempresas que llevan realizando procesos de producción o manufactura de forma manual. Por ello se genera diferentes problemas como pérdida de tiempo, aumento de costos en la producción, deficiente atención a los clientes y en consecuencia reducción de utilidades.

Esto se debe a los costos elevados de los distintos equipos de control y automatización, los cuales ayudarían a que sus diferentes procesos sean más eficientes, dichas empresas no pueden acceder a los mismos.

El presente proyecto se centra en la construcción de un controlador lógico programable con interface USB a base de un Microcontrolador de fácil programación y de bajo costo. Con la finalidad de obtener un PLC con reducidas dimensiones, facilidad de implementación, almacenamiento de programas, los cuales podrán ser modificados a través del interfaz USB para su posterior y rápida ejecución.

Lo mencionado anteriormente, permite resumir, que la integración de este tipo de equipos al ámbito laboral como estudiantil, sea una solución a los diferentes problemas que se dan en el día a día.

El PLC será programado a través de una aplicación de Windows Forms, que se encuentra instalada en el computador. Con ello cualquier persona desde un estudiante hasta un trabajador que tenga conocimientos de automatización, podrá utilizar el Controlador Lógico Programable.

RESUMEN

El presente proyecto está formado por cinco capítulos, los cuales detallan el proceso para la construcción del Controlador Lógico Programable.

El Capítulo I trata acerca de los fundamentos teóricos que tiene un PLC, estos son: Principios básicos del PLC, su clasificación, tipos de lenguaje de programación, especificaciones eléctricas/mecánicas/funcionales. Aplicando estos conceptos en el proyecto se logrará obtener un PLC de acuerdo a nuestros objetivos.

El Capítulo II está dedicado al microcontrolador e interfaz USB. A través de la teoría se obtendrá la selección de un PIC que cumpla con las características de este proyecto, y que mejor que tener en conjunto al PIC con una interfaz USB incorporada.

En el Capítulo III se realiza la implementación del Controlador Lógico Programable y la selección de los elementos que lo van a conformar, el diseño de la placa y el diseño del software que permite la programación del PLC desde el computador.

En el Capítulo IV se muestra el resultado de las pruebas a las que fue sometido el PLC para comprobar su correcto funcionamiento. Se detalla las conclusiones obtenidas durante la elaboración de este proyecto, además de recomendaciones para evitar problemas tanto en la construcción como en el uso del PLC.

Finalmente, en los anexos se encuentra el código de programación del PLC y del software que permite la programación del mismo, diseño de la placa del circuito del PLC y las hojas de datos de los dispositivos electrónicos utilizados.

CAPÍTULO I

CONTROLADORES LÓGICOS PROGRAMABLES

1.1 HISTORIA DEL PLC

Su historia se remonta a finales de 1960 cuando la industria buscó en las nuevas tecnologías electrónicas una solución más eficiente para reemplazar los sistemas de control basados en circuitos eléctricos con relés, interruptores y otros componentes utilizados.

La razón principal de tal hecho fue la necesidad de eliminar el gran costo que se producía al reemplazar el complejo sistema de control basado en relés y contactores.

En 1969 la Bedford Associates propuso algo denominado controlador digital modular (MODICON, Modular Digital CONTROLER) para la División Hydramatic de la General Motors, la cual instaló el primer PLC para reemplazar los sistemas inflexibles alambrados usados en sus líneas de producción. (Unicrom)

Para 1971, los PLC se extendían a otras industrias por lo que nace el MODICON 084, primer PLC del mundo en ser producido comercialmente.

En los años ochenta se produjo un intento de estandarización de las comunicaciones con el protocolo MAP (Manufacturing Automation Protocol) de General Motors. Además, se redujeron las dimensiones del PLC y se pasó a realizar los programas a través de ordenadores personales con programación simbólica en lugar de los clásicos terminales de programación.

En los años noventa, aparecieron los microprocesadores de 32 bits con posibilidad de operaciones matemáticas complejas, y de comunicaciones entre PLC de diferentes marcas y computadoras, estos nos brindaron la posibilidad de tener fábricas completamente automatizadas. Su programación es más sencilla ya que pueden ser programados en diagramas de bloques, lista de instrucciones C y texto estructurado al mismo tiempo.

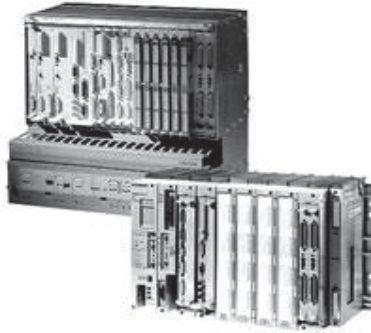


Figura 1.1 MODICON 084
Fuente: (Unicrom)

Actualmente la mayor parte de los procesos de fabricación son automatizados, ya que la combinación de la inteligencia de los autómatas programables con los accionares industriales permiten que se automatice un mayor número de procesos, liberando al ser humano tanto de tareas de gran compilación intelectual como de esfuerzo físico.

La potencia de los PLC ha aumentado considerablemente dejando de ser unos “lentos elementos de control que solo pueden sustituir a los cuadros de contactores” (Moreno, 2004) a ejecutar complicadas aplicaciones de control con un tiempo de ejecución mínimo y con alta fiabilidad.

1.1.1 PLC'S LOGO

LOGO es el módulo lógico universal de Siemens. Permite solucionar las aplicaciones cotidianas de una manera sencilla y disminuyendo costos. Puede resolver diferentes tareas desde el alumbrado de escaleras, luz exterior, toldos, persianas, hasta realizar el control de puertas, instalaciones de ventilación, bombas de agua no potable, etc. (Siemens)



Figura 1.2 Logo Siemens

Fuente: (Querevalu. 2013)

Toda la programación se realiza, de forma sencilla, con las 6 teclas que están situadas en la parte frontal del mismo. La visualización del programa, estado de entradas y salidas, parámetros, etc. se realiza en una pequeña pantalla LCD de forma gráfica.

Las funciones básicas (and, or, nand, nor, etc.) son idénticas en todos los modelos. Mientras que las funciones especiales, como relojes, temporizadores, etc. están limitados en algunos de los modelos de gama baja.

1.1.1.1 Software de programación de LOGO

El software de programación para PC es el LogoSoft Comfort el cual realiza funciones como una interface de comunicación entre el usuario y el sistema de control. A continuación, se mencionan las funciones que posee el software de programación de LOGO (Siemens)

- Elaboración de programas para su aplicación, el lenguaje de programación es de bloques.
- Simulación y visualización del circuito (o programa) en la PC, se pueden hacer pruebas previas con el programa antes de cargarlo a la memoria del controlador.
- Codificación y manipulación de los programas de usuario, se pueden realizar las modificaciones que necesite en algún momento dado el sistema.
- Generación e impresión de un diagrama del circuito.

- Protección de los datos del programa en un disco duro u otro medio, se pueden guardar diferentes programas que pueden ser usados y cargados al controlador posteriormente.
- Transferencia del programa de la PC a LOGO y de LOGO a la PC mediante una interface.
- Ajuste de horario de verano e invierno.

1.1.1.2 Funciones básicas LOGO

Las funciones básicas son elementos lógicos sencillos del álgebra de Boole. A continuación, se prevén las siguientes funciones básicas:

- AND
- NAND
- OR
- NOR
- XOR
- NOT

1.1.1.3 Funciones especiales LOGO

Las funciones especiales con las que cuenta LOGO son:

- Retardo a la conexión.
- Retardo a la desconexión.
- Retardo a la conexión/reconexión.
- Retardo a la conexión memorizada.
- Relé auto enclavado.
- Relé de impulsos.
- Relé de barrido (salida de impulsos).
- Relé disipador activado por flanco.
- Temporizador semanal.
- Temporizador anual.
- Contador avance/retroceso.
- Contador de horas de funcionamiento.
- Reloj simétrico.

- Generador de impulsos asíncronos.
- Generador aleatorio.
- Selector de umbral para frecuencias.
- Comparador analógico.
- Interruptor de alumbrado para escalera.
- Interruptor confortable.
- Textos de mensaje.
- Interruptor de software.

NOTA: Se debe tener en cuenta que Siemens cuenta con varios modelos de controlador LOGO y cada uno tiene distintas características tanto en las entradas y salidas como en las funciones especiales con las que cuenta. (Siemens)

1.2 PRINCIPIOS BÁSICOS DEL PLC

1.2.1 AUTOMATIZACIÓN

La automatización no es más que realizar las actividades de operación y funcionamiento en forma autónoma, es decir, por cuenta propia. (Quiroz, 2004)

La Automatización Industrial permite realizar mejoras al producto y a su proceso de fabricación. Asociado al producto se tienen parámetros como cantidad, calidad, mercado, métodos de producción, gestión y planificación de la producción, economía de producción y otros.

En el proceso de la automatización se utiliza sistemas de control, instrumentación industrial, tecnología informática como aplicaciones de software en tiempo real para supervisar, controlar las operaciones de plantas industriales. Con ello, se logra reducir la necesidad de la intervención humana en un proceso.

En síntesis, la automatización se puede entender como la facultad de autonomía o acción de operar por sí solo que poseen los procesos industriales y donde las actividades de producción son realizadas a través de acciones autónomas, la participación de fuerza física humana es mínima y la de inteligencia artificial, máxima.

1.2.1.1 Sistema de control

La Automatización Industrial se hace posible mediante los Sistemas de Control, que son conjuntos de equipos e instrumentos, que combinados con procedimientos mentales o algorítmicos trabajan en torno a propósitos previamente establecidos.

Generalmente se los divide en dos:

1.2.1.1.1 Parte operativa

Actúa directamente sobre la máquina. Son los elementos que hacen que la máquina se mueva y realice la operación deseada. Los elementos que forman la parte operativa son:

- **Captadores.** - Son elementos que informan al órgano de mando el estado del sistema, captan las señales necesarias para conocer el estado del proceso y decidir su desarrollo futuro. Detectan posición, presión temperatura, caudal, velocidad, etc. (Moreno, 2004) Estos se clasifican en:
 - **Captadores Pasivos.** - Son aquellos que cambian su estado lógico, activo/no activo, por medio de una acción mecánica. Estos son los interruptores, pulsadores, finales de carrera, etc.
 - **Captadores Activos.** - Son dispositivos electrónicos que necesitan ser alimentados por una tensión para que varíen su estado lógico. Este es el caso de los diferentes tipos de detectores (Inductivos, Capacitivos, Fotoeléctricos). Muchos de estos aparatos pueden ser alimentados por la propia fuente de alimentación del autómeta.
- **Accionadores.** - El accionador es el elemento final de control, actúa sobre la variable o elemento final del proceso. Los accionadores pueden ser clasificados en eléctricos, neumáticos e hidráulicos. Los más utilizados en la industria son los siguientes: cilindros, motores de corriente alterna, motores de corriente continua, etc.
 - **Pre-accionadores.** - Parte de mando o de control que se encarga de conmutar la conexión eléctrica, hidráulica o neumática entre los cables o conductores del circuito de potencia. Permiten activar los accionadores y pueden ser contactores, variadores de potencia, electroválvulas, etc.

1.2.1.1.2 Parte de mando

En un sistema automatizado el PLC está en el centro del sistema, ya que debe ser capaz de comunicarse con todos los constituyentes del sistema de control.

Los PLC's están especializados en sincronizar el flujo de entradas de sensores y eventos con el flujo de salidas a los actuadores y eventos. La Interfaz Hombre-Máquina (HMI) o interfaces hombre computadora, se suelen utilizar para comunicarse con los PLC's y otros equipos. (Moreno, 2004)

El personal de servicio se encarga del seguimiento y control del proceso a través de HMI. En donde no solo puede visualizar el estado actual proceso sino también hacer modificaciones a variables críticas del proceso.

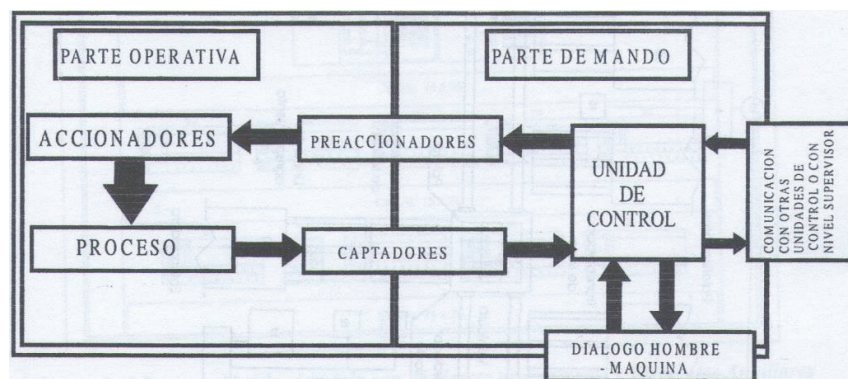


Figura 1.3 Esquema del sistema de control

Fuente: (Moreno, 2004)

1.2.1.2 Funciones básicas del PLC

1.2.1.2.1 Detección

Lectura de la señal de pulsadores, sensores, finales de carrera, etc. Que están distribuidos en un proceso de fabricación. (Moreno, 2004)

1.2.1.2.2 Mando

A través de los accionadores y pre-accionadores se elabora y envía las acciones al proceso. (Moreno, 2004)

1.2.1.2.3 Dialogo hombre - máquina

Se debe establecer un dialogo constante con los operarios de producción, obedeciendo sus consignas e informando del estado del proceso. (Moreno, 2004)

1.2.1.2.4 Programación

Permite el ingreso, elaboración y modificación de una aplicación para el PLC. (Moreno, 2004)

1.2.1.2.5 Redes de comunicación

Las redes permiten la comunicación y el intercambio de datos entre autómatas en tiempo real. En unos cuantos milisegundos pueden enviarse datos e intercambiar tablas de memoria compartida. (Moreno, 2004)

1.2.1.2.6 Sistemas de supervisión

Los PLC's permiten comunicarse con ordenadores provistos de programas de supervisión industrial. Esta comunicación se realiza por una red industrial o por medio de una conexión por el puerto serie del ordenador. (Moreno, 2004)

1.2.1.2.7 Entradas/salidas distribuidas

Los módulos de entrada / salida pueden estar distribuidos por la instalación, es decir, no es necesario que se encuentren en el mismo armario del autómata. Se comunican con la unidad central del PLC mediante un cable de red. (Moreno, 2004)

1.2.1.2.8 Buses de campo

Mediante un solo cable de comunicación se pueden conectar al bus, captadores y accionadores, reemplazando al cableado tradicional. El PLC consulta cíclicamente el estado de los captadores y actualiza el estado de los accionadores. (Moreno, 2004)

1.2.1.3 Campos de aplicación

El PLC por sus especiales características de diseño y por los avances tecnológicos en su hardware y software tiene un campo de aplicación muy extenso.

Su utilización se da fundamentalmente en aquellas instalaciones en donde es necesario un proceso de maniobra, control, señalización, etc. (Maser) Se lo aplica principalmente en procesos de fabricación industrial de cualquier tipo como transformaciones industriales, control de instalaciones, etc.

1.2.2 DESCRIPCIÓN Y FUNCIONAMIENTO DEL PLC

1.2.2.1 Descripción del PLC

El PLC (Controlador Lógico Programable) es un equipo electrónico programable diseñado para controlar en tiempo real y en un medio industrial procesos secuenciales de control. (Maser)

Posee las herramientas necesarias, tanto de software como de hardware, para controlar dispositivos externos, recibir señales de sensores y tomar decisiones de acuerdo al programa que el usuario elabore según el esquema del proceso a controlar.

Los PLC permiten realizar el control de cualquier tipo de máquina a diferencia de otros controladores automáticos que solamente pueden controlar un tipo específico de aparato.

Son conocidos también como Autómatas Programables ya que como su nombre lo indica, son automáticos, es decir son dispositivos que comparan las señales emitidas por la máquina controlada y toman decisiones en base a las instrucciones programadas, para mantener estable la operación de dicha máquina. Se puede modificar las instrucciones almacenadas en la memoria, además de monitorizarlas.

1.2.2.2 Funcionamiento del PLC

El PLC en sus entradas recibe información de una variedad de sensores, captadores, etc. Estas señales pueden ser de dos tipos: digitales o analógicas, las mismas son transformadas internamente en señales compatibles con los microprocesadores y demás circuitos integrados de procesamiento interno.

El microprocesador procede a ejecutar las instrucciones indicadas en el programa de usuario, almacenado en su memoria, generando órdenes o señales de mando a partir de las señales de entrada. (Maser)

Las entradas leídas se guardan en una memoria temporal conocida también como Imagen de entradas. A esta acude la CPU en la ejecución del programa, y según se va obteniendo las salidas, se guardan en otra memoria temporal o Imagen de salida.

Una vez ejecutado el programa completo, estas imágenes de salida se transfieren todas a la vez al módulo de salida. El PLC altera sus salidas, inicialmente con señales de formato digital y posteriormente a otro formato de acuerdo a los actuadores que se vayan a utilizar, ya sean digitales o analógicos.

A fin de optimizar el tiempo, la lectura y escritura de las señales se realiza a la vez para todas las entradas y salidas.

1.2.2.2.1 Ciclo de funcionamiento

El ciclo de funcionamiento se divide en dos partes:

- Proceso inicial

Antes de entrar en el ciclo de operación el PLC realiza una serie de acciones para inicializar los estados del mismo y chequear el hardware. Estas rutinas de chequeo comprueban:

- Elementos y equipos eléctricos.
- El nivel de la batería, si esta existe.
- La conexión de las memorias internas del sistema.
- El módulo de memoria exterior conectado, si existe.

Transcurrido el Proceso Inicial y si no han aparecido errores el PLC entra en el Ciclo de Operación.

- Ciclo de operación

Este ciclo se divide en tres bloques:

- Proceso Común. - Se realizan los chequeos cíclicos de conexiones y de memoria de programa protegiendo al sistema contra:
 - Errores de hardware
 - Errores de sintaxis (programa imposible de ejecutar).

El chequeo cíclico de conexiones comprueba los siguientes puntos:

- Niveles de tensión de alimentación.
- Estado de la batería si existe.
- Buses de conexión con las interfaces.

El chequeo de la memoria de programa comprueba la integridad de la misma y los posibles errores de sintaxis y gramática:

- Mantenimiento de los datos.
- Existencia de la instrucción END de fin de programa
- Códigos de instrucciones correctas
- Ejecución del programa. - Se consultan los estados de las entradas y de las salidas. Se elaboran las órdenes de mando o de salida para los mismos.
- Servicio a Periféricos.- En el caso de existir algún intercambio de información, el CPU le dedica un tiempo de 1 a 2 ms, en el intercambio de datos. (Maser)

1.2.2.2.2 Tiempo de ejecución en tiempo real

El tiempo total que el Controlador Lógico Programable emplea para realizar un ciclo de operación se llama tiempo de ejecución de ciclo de operación o Scan time.

Dicho tiempo depende de:

- El número de E/S involucradas.
- La longitud del programa usuario.
- El número y tipo de periféricos conectados al autómata.

1.2.3 ARQUITECTURA INTERNA DEL PLC

La arquitectura interna del PLC está formada por los siguientes elementos básicos:

- CPU
- Módulos de Entradas
- Módulos de Salida

Para que el PLC sea operativo son necesarios otros elementos tales como:

- Fuente de alimentación.
- Memorias.
- La unidad o consola de programación.

- Los dispositivos periféricos.

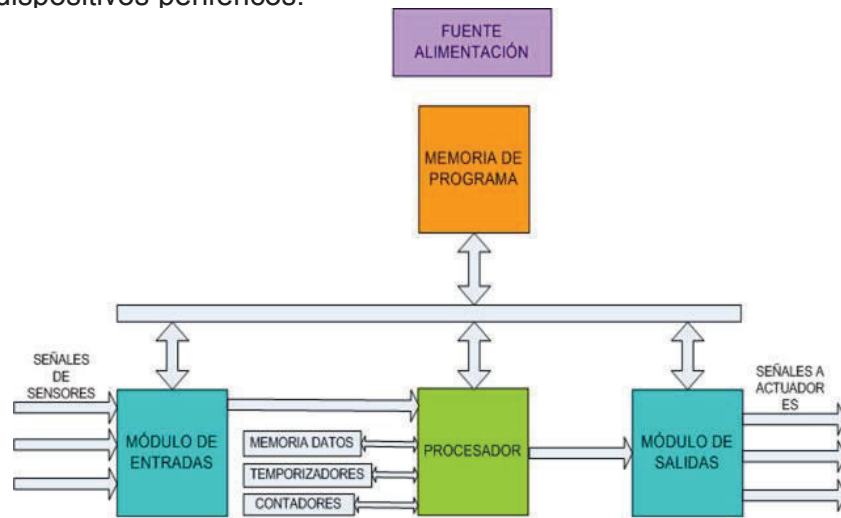


Figura 1.4 Arquitectura interna PLC

Fuente: (adrformación)

1.2.3.1 CPU

La Unidad Central de Proceso es el cerebro del PLC, se encarga de la toma de decisiones. Interpreta las instrucciones del programa que se haya diseñado para el control del proceso y consulta el estado de las entradas. Dependiendo de dichos estados y del programa, ordena la activación de las salidas deseadas. (Molinari, 2012)

La CPU está conformada por el microprocesador que realiza una gran cantidad de operaciones como operaciones de tipo lógico, aritmético, de control de la transferencia de la información dentro del automático.

La memoria monitor del sistema es una memoria de tipo ROM que contiene el sistema operativo del automático y las siguientes rutinas incluidas por el fabricante.

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

1.2.3.2 Módulos entrada/salida

Realizan la intercomunicación entre los dispositivos externos al PLC como son los captadores, accionadores y todos los circuitos electrónicos de baja potencia que

comprenden a la unidad central de proceso del PLC, es decir dichos módulos tienen la misión de proteger y aislar la etapa de control que está conformada por el microcontrolador, de todos los elementos que se encuentran fuera de la unidad central de proceso.

Dependiendo del tipo de señal (digital o analógica) los módulos de entrada/salida son los siguientes:

1.2.3.3 Módulos entrada digital

Los módulos de entrada digitales permiten conectar al autómeta, captadores como finales de carrera, pulsadores, etc. Los módulos de entrada digitales trabajan con voltajes estándar hasta 24 voltios C.C. Si a una línea llegan 24 voltios se interpreta como un 1L y cuando llegan cero voltios se interpreta como un 0L. (Maser)

1.2.3.3.1 Módulos entrada analógica

Estos módulos son utilizados para leer valores analógicos como temperatura, presión, intensidad, etc. Convierten una magnitud analógica en un número que se deposita en una variable interna del PLC. Posteriormente se realiza una conversión A/D, con una precisión o resolución determinada (número de bits) y cada cierto intervalo de tiempo (periodo muestreo) ya que el PLC solo trabaja con señales digitales. (Maser)

1.2.3.3.2 Módulos salida digital

Los módulos de salida digital permiten al autómeta programable actuar sobre los pre-accionadores y accionadores.

El valor binario de las salidas digitales se convierte en la apertura o cierre de un relé interno en el caso de módulos de salidas a relé y en los módulos estáticos los elementos que conmutan son los componentes electrónicos como transistores o triacs. (Maser)

1.2.3.3.3 Módulos salida analógica

Permiten transmitir hacia los actuadores analógicos señales de tensión o intensidad que varían continuamente.

Se realiza una conversión D/A con una precisión o resolución determinada (número de bits) y cada cierto intervalo de tiempo (periodo muestreo).

Esta tensión o intensidad puede servir de referencia de mando para actuadores que admitan mando analógico como pueden ser los variadores de velocidad, las etapas de los tiristores de los hornos, reguladores de temperatura, etc. Permitiendo al autómatas realizar funciones de regulación y control de procesos continuos.

1.2.3.4 Fuente de alimentación

La fuente de alimentación proporciona las tensiones necesarias para el funcionamiento de los distintos módulos que se conecten al PLC, así como a la unidad central de proceso y los diferentes circuitos del sistema.

Su función es reducir y adaptar la tensión de la red de 120V o 220V A.C a baja tensión de D.C normalmente 24 V, siendo ésta la tensión de trabajo en los circuitos electrónicos que forma el Autómata.

1.2.3.5 Memorias

La memoria es el lugar donde el Controlador Lógico Programable guarda todo cuanto necesita para ejecutar la tarea de control como datos de proceso, control.

- Datos de Proceso:
 - Señales de planta, entradas y salidas.
 - Variables internas, de bit y de palabra.
 - Datos alfanuméricos y constantes.
- Datos de Control:
 - Instrucciones del programa
 - Configuración Controlador Lógico Programable (modo de funcionamiento, número de E/S conectadas. (Maser)

Existen varios tipos de memorias:

- RAM. - Se utiliza principalmente como memoria interna y como memoria de programa en el caso de que pueda asegurarse el mantenimiento de los datos con una batería exterior.
- ROM.- En esta memoria se almacena el programa monitor del sistema.

- EEPROM. - Almacena el programa de usuario, una vez que ha sido convenientemente depurada. En la mayoría de PLC's se utiliza la combinación de una memoria RAM con EEPROM como protección de la información. En el caso de que exista falta de alimentación eléctrica el contenido de la memoria RAM pasa a la EEPROM y una vez reanudada la alimentación, el contenido de la EEPROM se transfiere hacia la RAM. Las soluciones de este tipo están sustituyendo a las clásicas RAM + batería puesto que presentan menos problemas. (Maser)

1.2.3.6 Consola de programación

Es un terminal a modo de ordenador que proporciona una forma más cómoda de realizar el programa de usuario y observar parámetros internos del PLC.

La consola de programación en la actualidad ha sido reemplazada por la PC que nos permite programar desde un ordenador personal estándar, con todo lo que ello supone: herramientas más potentes, posibilidad de almacenamiento en soporte magnético, impresión, transferencia de datos, monitorización mediante software SCADA, etc.

1.2.3.7 Dispositivos periféricos.

El autómata programable, en la mayoría de los casos, puede ser ampliable. Las ampliaciones abarcan varias posibilidades, que van desde módulos auxiliares de E/S, memoria adicional hasta la conexión con otros autómatas del mismo modelo. Cada fabricante facilita las posibilidades de ampliación de sus modelos, los cuales pueden variar incluso entre modelos de la misma serie.

1.3 CLASIFICACIÓN DEL PLC

Los PLC se clasifican de la siguiente manera:

1.3.1 PLC NANO

Los PLC tipo Nano se caracterizan por presentar en un solo bloque todos sus elementos fuente de alimentación, CPU, memorias, entradas/salidas, etc.

Su potencia de proceso es limitada dedicándose a controlar máquinas muy pequeñas o cuadros de mando. Gracias al avance tecnológico han ido aumentando su potencia con la posibilidad de incorporar módulos de ampliación de entradas/salidas, analógicas y digitales.

Un ejemplo de un PLC Nano es el LOGO de Siemens.



Figura 1.6 PLC Tipo Nano
Fuente: (AliExpress)

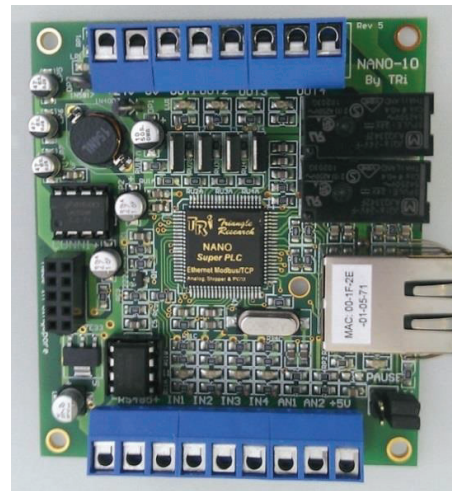


Figura 1.5 Estructura interna PLC
Fuente: (COMPARE)

1.3.2 PLC COMPACTO

Se caracteriza por separar las E/S del resto del PLC, y en un bloque compacto están reunidas las CPU, memoria de usuario o de programa y fuente de alimentación.

Su tamaño es superior a los PLC tipo Nano y soportan una gran variedad de módulos especiales, tales como:

- Entradas y salidas analógicas
- Módulos contadores rápidos
- Módulos de comunicaciones
- Interfaces de operador
- Expansiones de entrada y salida



Figura 1.7 PLC Tipo Compacto
Fuente: (Ayala, 2012)

1.3.3 PLC MODULAR

Estos PLC tienen como característica principal que para cada elemento que lo componen existe un módulo como, por ejemplo: módulos entrada/salida, fuente de alimentación, CPU, E/S, etc.



Figura 1.8 PLC Tipo Modular
Fuente: (Automatización)

1.4 VENTAJAS Y DESVENTAJAS DEL PLC

1.4.1 VENTAJAS

El uso del PLC en el proceso de automatización industrial frente al control manual, brinda ciertas ventajas y beneficios de orden económico y tecnológico como:

- Menor costo de mano de obra.
- Empleo de poco espacio.
- Fácil instalación.
- Menos consumo de energía.
- Control más preciso.
- Mayor rapidez de respuesta.
- Flexibilidad Control de procesos complejos.
- Facilidad de programación.
- Seguridad en el proceso.
- Posibilidad de añadir modificaciones sin elevar costos.
- Menor costo de instalación, operación y mantenimiento.

1.4.2 DESVENTAJAS

Las desventajas de un PLC son:

- Mano de obra especializada.
- Centraliza el proceso.
- Condiciones ambientales apropiadas.
- Mayor costo para controlar tareas muy pequeñas o sencillas.

1.5 LENGUAJES DE PROGRAMACIÓN PARA EL PLC

Los lenguajes de programación ayudan a la comunicación entre el usuario y el PLC. La interacción que tiene el usuario con el PLC se la realiza por medio de un computador, el cual tiene cargado el software que enviará las instrucciones al PLC. En procesos grandes o en ambientes industriales el PLC recibe el nombre de API (Autómata Programable Industrial) y utiliza como interface para el usuario pantallas de plasma o pantallas táctiles.

Los fabricantes de PLC han desarrollado varios tipos de lenguajes de programación en mayoría de los casos siguiendo normas internacionales, con el fin de facilitar la interacción de los programadores. En la tabla 1.1 se presentan lenguajes de uso común. (Molina, 2005)

Tabla 1.1 Lenguajes de programación PLC

Lenguaje	Características	Ejemplos*	Tipo	Nivel
Listas	Lista de Instrucciones	IL AWL STL IL/ST	Escrito	Bajo
Plano	Diagrama Eléctrico	LADDER LD KOP	Visual	Alto
Diagrama de Bloques Funcionales	Diagrama Lógico	FBD FBS FUD		
Organigrama de Bloques Secuenciales	Diagrama Algorítmico	AS SFC PETRI GRAFSET		
Otros	Lenguajes Usados en Otras Áreas de la computación	BASIC C	Escrito	

1.5.1 CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación para PLC son de dos tipos, visuales y escritos.

Los lenguajes visuales permiten estructurar el programa por medio de símbolos gráficos, entre estos los más destacados tenemos:

- LADDER. - El cual representa el funcionamiento deseado, como en un circuito de contactores el cual es fácil de entender y utilizar para usuarios con experiencia en diagramas unifilares.
- FBD. - Conocido también como diagrama de bloque funcional el cual utiliza los diagramas lógicos de la electrónica digital.

Los lenguajes escritos son listados de sentencias que describen las funciones a ejecutar, instrucción por instrucción.

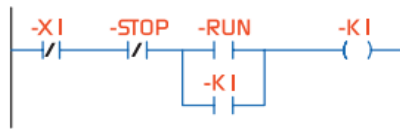


Figura 1.9 Lenguaje de programación
Fuente: (Molina, 2005)

En la figura 1.9 se puede observar una representación gráfica del circuito de control donde: XI y STOP son contactos normalmente cerrados, RUN y KI son contactos normalmente abiertos, mientras que KI que se encuentra sobre un paréntesis () representa la salida.

El circuito se pone en marcha al accionar RUN, que puede ser un pulsador, lo que activará la salida KI () y a la vez accionará el contacto KI, con lo que el circuito se mantendrá trabajando indefinidamente hasta que se presione el pulsador de paro (STOP) o la protección (XI).

- LENGUAJE BOOLEANO. - El lenguaje Booleano utiliza la sintaxis del Álgebra de Boole para ingresar y explicar la lógica de control. Consiste en elaborar una lista de instrucciones, haciendo uso de operadores Booleanos (AND, OR, NOT, etc.) y otras instrucciones, para implementar el circuito de control. Ejemplo de programación Booleana:

Función	In/Out	Operación
A	I	2.3
A	I	4.1
O	I	3.2
=	Q	1.6

Tabla 1.2 Ejemplo lenguaje booleano

En la tabla 1.2 (Molina, 2005) se observan tres columnas: la primera representa la función a cumplir, A (AND), O (OR), = (resultado).

La segunda columna indica si se utiliza una entrada (I) o una salida (Q). Mientras que en la tercera columna muestra un número que representa una función específica entre cada una de las entradas.

- LENGUAJE DE TEXTO ESTRUCTURADO. - Es un lenguaje de alto nivel que permite que muchas tareas complejas puedan ser divididas en unidades más pequeñas. ST se parece mucho a los lenguajes de computadoras BASIC o PASCAL, que usa subrutinas para llevar a cabo diferentes partes de las funciones de control. Al igual que LD, FBD e IL, el lenguaje de texto estructurado utiliza la definición de variables para identificar entradas y salidas de dispositivos de campo y cualquier otra variable creada internamente. La programación en Texto Estructurado (ST) es el mejor lenguaje para la implementación de aplicaciones de inteligencia artificial, lógica difusa, toma de decisiones, etc.

Ejemplo de texto estructurado:

```
IF Manual AND Alarm THEN
Level = Manual_Level;
Mixer = Start AND NOT Reset
ELSE IF    Other_Mode THEN
Level = Max_level;
ELSE      Level = (Level_Indic X100)/Scale;
END IF;
```

El ejemplo nos indica la programación de un tanque que recibe un tipo de líquido y lo mezcla por lo que mediante las instrucciones ingresadas el funcionamiento será el siguiente:

Si tenemos que el selector está en manual y con alarma, entonces el nivel se podrá seleccionar de forma manual y la mezcla comenzará y no se detendrá; pero si se encuentra en otro modo, entonces el nivel llegará al máximo, caso contrario el nivel estará dado por la fórmula descrita.

1.6 ESPECIFICACIONES TÉCNICAS

Para cada tipo de PLC se presenta una serie de especificaciones técnicas donde se puede observar; sus dimensiones, su peso, el tipo de montaje; así como también especificaciones eléctricas que se mostrarán a continuación.

1.6.1 ESPECIFICACIONES MECÁNICAS.

En la tabla 1.3 se detallan las especificaciones mecánicas de un de PLC LOGO. (Siemens)

Tabla 1.3 Especificaciones mecánicas PLC

Criterio	Valores
Dimensiones (AxAxP)	72 x 90 x 55 mm
Peso	aprox. 190 g
Montaje	En perfil soporte de 35 mm 4 unidades de distribución de ancho o montaje en la pared
Temperatura ambiente	
montaje horizontal	0 ... 55 °C
montaje vertical	0 ... 55 °C
Almacenaje/transporte	- 40 °C ... +70 °C
Humedad relativa	Del 10 al 95% sin condensación
Presión atmosférica	795 ... 1080 hPa (hectoPascales)

1.6.2 ESPECIFICACIONES ELÉCTRICAS.

En la siguiente tabla se detallan, los valores de voltajes para energizar el PLC, voltajes en las entradas y salidas, valores de corriente, etc. (Siemens)

Tabla 1.4 Especificaciones eléctricas PLC

	LOGO! 24RC LOGO! 24RCo	LOGO! DM8 24R
Fuente de alimentación		
Tensión de entrada	24 V CA/CC	24 V CA/CC
Rango admisible	20,4 ... 26,4 V CA 20,4 ... 28,8 V CC	20,4 ... 26,4 V CA 20,4 ... 28,8 V CC
Frecuencia de red admisible	47 ... 63 Hz	47 ... 63 Hz
Consumo de corriente 24 V AC 24 V DC	40 ... 110 mA 20 ... 75 mA	40 ... 110 mA 20 ... 75 mA
Compensación de fallos de tensión	Típ. 5 ms	Típ. 5 ms
Potencia disipada en caso de 24 V	0,5 ... 2,9 W (CA)	0,5 ... 2,9 W (CA)
Exactitud del reloj de tiempo real	Máx. 5 s / día	Máx. 5 s / día
Entradas digitales		
Cantidad	8, alternativo entre conexión P y N	4, alternativo entre conexión P y N
Separación galvánica	no	no
Tensión de entrada señal 0 señal 1	L <5 V c.a. >12 V c.a.	L <5 V c.a. >12 V c.a.
Salidas digitales		
Cantidad	4	4
Tipo de las salidas	Salidas a relé	Salidas a relé
Separación galvánica	sí	sí
En grupos de	1	1
Activación de una entrada digital	sí	sí
Corriente constante I_{th}	Máx. 10 A por relé	Máx. 5 A por relé
Carga de lámparas incandescentes	1.000 W	1.000 W

CAPÍTULO II

MICROCONTROLADOR E INTERFAZ USB

2.1 MICROCONTROLADOR

2.1.1 INTRODUCCIÓN

Un microcontrolador es un circuito integrado que fue diseñado para incluir en su interior las tres unidades fundamentales de una computadora: CPU, Memorias y Unidades de E/S.

Su completa funcionalidad se lleva a cabo a través de un programa que puede desarrollarse en distintos lenguajes de programación como C++, assembler entre otros. (Costales, 2012) Es almacenado en su memoria y con ello se puede realizar cualquier tipo de aplicaciones como:

- Equipos de Comunicaciones y de Telefonía.
- Aplicaciones industriales de automatización.
- Equipos médicos e industriales.
- Sistemas aeroespaciales.

Se emplea la notación μC o las siglas MCU (por microcontroller unit) para referirse a los microcontroladores.

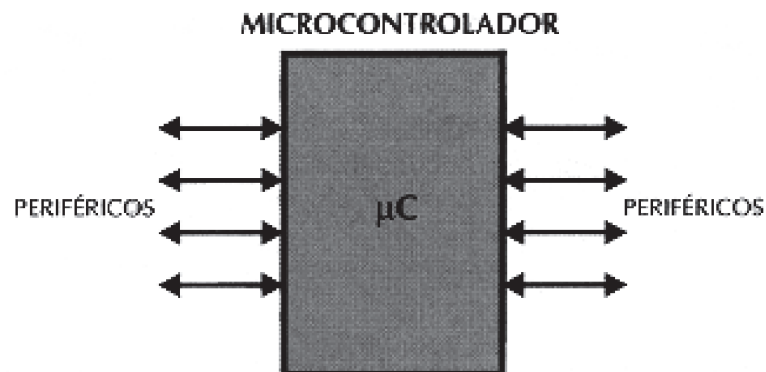


Figura 2.1 Microcontrolador
Fuente: (WordPress.com, 2011)

2.1.2 CARACTERÍSTICAS

Debido a las diferentes aplicaciones que se pueden desarrollar a través de los microprocesadores los fabricantes han desarrollado el concepto de familia que se define como un conjunto de microcontroladores que poseen una misma arquitectura interna, juego de instrucciones y que se diferencian en una o más de las siguientes características (Pérez, 2007)

2.1.2.1 Capacidad de operación en paralelo

Es la capacidad de operación de un microcontrolador como el número de bits de las combinaciones binarias con las que se realiza operaciones aritméticas o lógicas en un solo ciclo de instrucción. Los microcontroladores más utilizados son de 8 bits y para aplicaciones en las que se trabaja con informaciones binarias de un número elevado de bits se utiliza los de 16 y 32 bits

2.1.2.2 Capacidad de memoria

En los microcontroladores existe una memoria en la que se almacena los datos y otra las instrucciones. A continuación, se presenta los dos tipos de memoria.

2.1.2.2.1 Memoria de datos

Se caracteriza porque en ella se puede leer y escribir durante la ejecución del programa y pueden ser de dos tipos:

- Memorias volátiles. - RAM y NVRAM.
- Memorias no volátiles. - EPROM, EEPROM.

2.1.2.2.2 Memoria de instrucciones

Esta memoria solamente se lee durante la ejecución del programa y no puede ser volátil. Por ello se utilizan las memorias de acceso aleatorio no volátiles ROM o conocidas también como memorias pasivas.

- Memoria acceso aleatorio ROM.- Las memorias tipo ROM se caracterizan porque su contenido es establecido por el fabricante y no puede ser modificado por el usuario.

- Memoria de acceso aleatorio PROM. - Las memorias PROM (Programmable Read Only Memories) pueden ser programadas solo una vez. También son conocidas como OTP (One Time Programmable).
- Memoria de acceso aleatorio R PROM. - Aunque el termino R PROM no es muy reconocido permite identificar a todas las memorias de acceso aleatorio no volátiles cuyo contenido se puede escribir mediante impulsos eléctricos como EPROM, EEPROM, Flash. Las memorias EEPROM y Flash son memorias que pueden ser programadas en el propio sistema en el que están instaladas, a este proceso se conoce como Programación Serial En Circuito. (Pérez, 2007)

2.1.2.2.3 ICSP

ICSP (Programación Serial En Circuito) es una tecnología incluida en todos los microcontroladores PIC de Microchip que posibilita la reprogramación de los mismos sin que sea necesario removerlos de su circuito de aplicación.

Esta característica es de gran utilidad como se presenta a continuación.

- En la etapa de desarrollo ahorra tiempo y trabajo, porque nos permite reprogramar los microcontroladores en sus respectivas placas, ya sea para corregir errores de software o para la mejora del sistema con la implementación de nuevas funciones.
- Permite realizar diferentes actualizaciones al sistema.
- Calibra el sistema durante la fabricación o en la placa.

2.1.2.3 Puertos de E/S

Los puertos de Entrada y Salida (E/S) permiten comunicar al microcontrolador con el mundo exterior, a través de interfaces, o con otros dispositivos. Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

Los puertos también conocidos como pórtricos pueden ser configurados de acuerdo a la necesidad de la aplicación.

2.1.2.4 Juego de instrucciones

El juego de instrucciones es un conjunto de operaciones de transferencia aritméticas, lógicas y movimientos entre el acumulador o registro de trabajo y los registros de la memoria RAM del microcontrolador. (Pérez, 2007)

De acuerdo con el juego de instrucciones los microcontroladores se pueden clasificar en:

- RISC. - Proviene de las siglas Conjunto de Instrucciones Reducidas para Computador poseen un número reducido de instrucciones, unas 35 instrucciones, y un número elevado de registros de aplicación en general.
- CISC. - Conjunto de Instrucciones complejas para Computador poseen un número elevado de instrucciones, más de 80 instrucciones, y un potente juego de modos de direccionamiento o formas de seleccionar los operandos y las instrucciones.

2.1.2.5 Bajo consumo

Debido a que en varias aplicaciones se desea utilizar baterías como fuente de alimentación, es de gran necesidad que el microcontrolador consuma poca energía cuando se ejecuta el programa como cuando no está realizando ninguna acción. Para ello es conveniente paralizar total o parcialmente al PIC poniéndolo a dormir hasta que ocurra la acción esperada.

2.1.2.6 Protección de los programas frente a copias

Se trata de proteger la información almacenada en la memoria, es decir, el programa de la aplicación contra lecturas furtivas de la memoria del Microcontrolador.

Los microcontroladores disponen de recurso para proteger el programa y estos son:

- La configuración del dispositivo para que la memoria del programa no se pueda leer una vez que haya sido programada.
- En microcontroladores con arquitectura de memoria abierta lo que se hace es encriptar la información que se intercambia con la memoria externa.

2.1.3 ARQUITECTURA DE LOS MICROCONTROLADORES

Existen dos tipos de arquitecturas presentes en los microcontroladores: Von Neumann y Harvard. Ambas se diferencian en la forma de conexión de la memoria al procesador y en los buses que cada una necesita.

2.1.3.1 Arquitectura Von Neumann

La arquitectura Von Neumann es la arquitectura tradicional en computadoras y microprocesadores. Como se puede observar en la figura 2.2 la Unidad Central de Proceso (CPU), está conectada a una memoria principal única (casi siempre solo RAM) donde se guardan las instrucciones del programa y los datos. A dicha memoria se accede a través de un sistema de buses único (control, direcciones y datos). (Costales, 2012)

Las principales limitaciones que nos encontramos con la arquitectura Von Neumann son:

- La limitación de la longitud de las instrucciones por el bus de datos, que hace que el microprocesador tenga que realizar varios accesos a memoria para buscar instrucciones complejas.
- La limitación de la velocidad de operación a causa del bus único para datos e instrucciones que no deja acceder simultáneamente a unos y otras, lo cual impide superponer ambos tiempos de acceso.

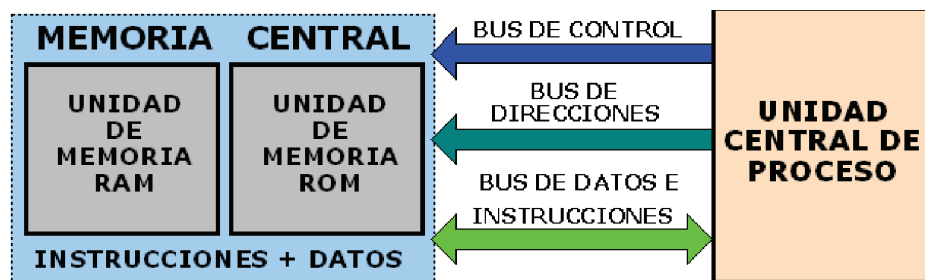


Figura 2.2 Arquitectura Von Neuman
Fuente: (Camacho, 2012)

2.1.3.1 Arquitectura Harvard

La arquitectura Harvard que está representada en la figura 2.3, tiene la Unidad Central de Proceso (CPU) conectada a dos memorias una de ellas con las instrucciones del programa y otra almacena datos.

Se comunican a través de buses que son totalmente independientes lo que permite que la CPU pueda acceder de forma independiente y simultánea a la memoria de datos y a la de instrucciones. De esta manera la arquitectura Harvard permite:

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

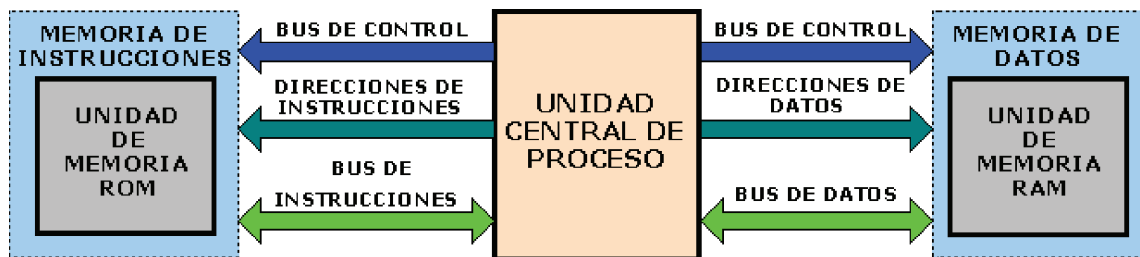


Figura 2.3 Arquitectura Harvard
Fuente: (Camacho, 2012)

2.2 MICROCONTROLADORES PIC

Son los microcontroladores más utilizados en el mercado actual. Pertenecen a la familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc.

Se los identifica por su nombre completo Peripheral Interface Controller (controlador de interfaz periférico).

El éxito de los PIC's de Microchip se basa en su utilización. Cuando se aprende a manejar uno de ellos, conociendo su arquitectura y juego de instrucciones. Es mucho más fácil utilizar otro modelo.

2.2.1 CLASIFICACIÓN DE LOS MICROCONTROLADORES PIC

Los microcontroladores PIC se clasifican según el tamaño de sus instrucciones en tres grupos o gamas:

- Gama Baja. - Su característica principal es su reducido tamaño. Tienen un repertorio de 33 instrucciones de 12 bits cada una. La memoria de programa tiene una capacidad de hasta 2048 (2K) palabras de 12 bits. La memoria de datos está formada por registros de 8 bits y se organizan en bancos de hasta 32 registros cada uno. (Pérez, 2007)
- Gama Media. - Es la gama más variada y completa. Con un repertorio de 35 instrucciones de 14 bits cada una. La memoria de programa tiene una capacidad de hasta 8192 (8K) palabras de 14 bits. La memoria de datos está formada por registros de 8 bits y se organizan en bancos de hasta 120 registros cada uno. Son compatibles con los PIC de gama baja además permiten interrupciones, poseen comparadores de magnitudes analógicas, convertidores A/D, puerto serie y diversos temporizadores. (Pérez, 2007)
- Gama Alta. - Con 58 instrucciones de 16 bits cada una. Estos PIC se caracterizan por tener arquitectura abierta, es decir, permite la ampliación de las memorias de programa y de datos. El número de dispositivos E/S es bastante amplio a comparación de las gamas mencionadas anteriormente.

2.2.2 CARACTERÍSTICAS DE LOS MICROCONTROLADORES PIC

Las principales características de los microcontroladores PIC son las siguientes:

- Tienen facilidad de desarrollo de aplicaciones debido a su reducido repertorio de instrucciones. Ya que los microcontroladores PIC son de tipo RISC. El número de instrucciones es diferente para cada gama varía entre 33 y 77.
- Utilizan la arquitectura Harvard que se caracteriza porque la memoria de programa genera la dirección de la memoria de datos. La memoria de datos está estructurada en palabras de 8 bits y la memoria de programa en palabras de 12,14 o 16 bits.
- Al momento de ejecutar las instrucciones se aplica la segmentación (pipeline) o búsqueda y ejecución simultanea de instrucciones, es decir, que al mismo tiempo

se realiza la ejecución de una instrucción y la búsqueda de la siguiente esto hace que se ejecuten siempre en un ciclo de instrucción con excepción de las instrucciones de salto.

- Poseen una memoria RAM de pequeña capacidad comprendida entre 25 y 3840 octetos dependiendo del modelo y direccionada por la memoria no volátil EPROM, OTP, FLASH, etc. Que contiene el programa.
- Los PIC's cuentan con una amplia gama de dispositivos de entrada y salida. Disponen de puertos paralelos de 8 bits, temporizadores, puertos serie sincrónicos, convertidores A/D, moduladores de ancho de pulso, etc.

2.2.3 ARQUITECTURA INTERNA DEL PIC

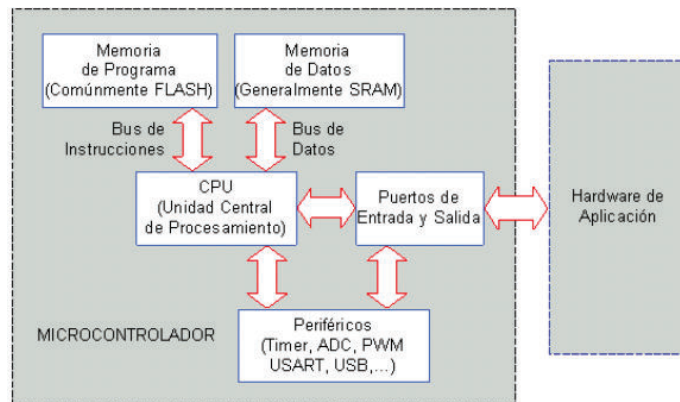


Figura 2.4 Arquitectura interna del PIC
Fuente: (Arduino)

Los elementos principales que conforman la arquitectura interna del PIC son:

2.2.3.1 Recursos fundamentales

2.2.3.1.1 Procesador

Es el elemento más importante del microcontrolador se encarga del procesamiento de las instrucciones, así como la búsqueda de los operandos y el almacenamiento del resultado.

Por la necesidad de conseguir elevados rendimientos en el procesamiento de las instrucciones se generalizó el empleo de los procesadores de arquitectura Harvard frente a los tradicionales que seguían la arquitectura de von Neumann. El procesador de los modernos microcontroladores responde a la arquitectura RISC.

Internamente el procesador está formado por:

- Registros. - Son un espacio de memoria muy reducido de aquí se toman los datos para varias operaciones que deben realizar el resto de circuitos del procesador. Los registros almacenan los resultados de la ejecución de instrucciones, cargan datos desde la memoria externa o los almacenan en ella.
- Unidad de control. - En ella recae la lógica necesaria para la decodificación y ejecución de las instrucciones, el control de los registros, la ALU, los buses, etc. La unidad de control es uno de los elementos fundamentales que determinan las prestaciones del procesador, ya que determina parámetros tales como el tipo de conjunto de instrucciones, velocidad de ejecución, tiempo del ciclo de máquina, tipo de buses que puede tener el sistema, manejo de interrupciones. La unidad de control se divide en: unidad de decodificación, unidad de ejecución, controladores de memoria cache, controladores de buses, controlador de interrupciones, pipelines, entre otros elementos, dependiendo siempre del tipo de procesador.
- Unidad de aritmética lógica. - En esta unidad básicamente se realiza operaciones lógicas y matemáticas como: sumas, restas, y operaciones lógicas típicas del álgebra de Boole. Este tipo de unidades han evolucionado mucho y los procesadores tienen varias ALU, especializadas en la realización de operaciones complejas como las operaciones en coma flotante

2.2.3.1.2 Memorias

En el microcontrolador la memoria de instrucciones y de datos se encuentran en un mismo chip.

- Memoria de instrucciones. - Conocida también como memoria de programa, aquí se almacenan todas las instrucciones del programa de control y no hay posibilidad de utilizar memorias externas de ampliación. Existen algunos tipos de memoria adecuados para realizar lo que se mencionó anteriormente estas son:
 - ROM
 - EPROM
 - EEPROM

- Memoria de datos. - La memoria RAM está destinada al almacenamiento de información temporal que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. El tipo de memoria utilizada en las memorias RAM de los microcontroladores es SRAM, lo que evita tener que implementar sistemas de refrescamiento.
- Buses. - Son el medio de comunicación que utilizan los diferentes componentes del procesador para intercambiar información. Existen tres tipos de buses:
 - Bus de dirección. - Se utiliza para seleccionar al dispositivo con el cual se quiere trabajar o en el caso de las memorias, seleccionar el dato que se desea leer o escribir.
 - Bus de datos. - Conjunto de líneas de un bit cada una realizan la transmisión de datos del sistema.
 - Bus de control. - Se utiliza para gestionar los distintos procesos de escritura lectura y controlar la operación de los dispositivos del sistema.

2.2.3.1.3 Reloj principal

Es el encargado de indicar al microcontrolador la velocidad en la que debe trabajar. En algunos casos el circuito de reloj está incorporado en el mismo y en otros se le adiciona un circuito externo conocido como oscilador de frecuencia, entre los cuales se encuentran:

- Oscilador RC. - Este tipo de oscilador es de bajo costo, se lo utiliza cuando la precisión y la estabilidad de la frecuencia de trabajo no son esenciales.

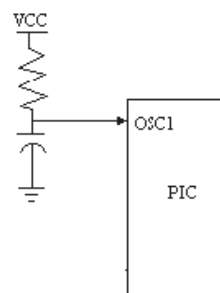


Figura 2.5 Oscilador RC
Fuente: (Frino)

- Oscilador XT. - Está conformado por dos capacitores y un cristal de cuarzo. La condición básica importante para que este oscilador funcione es que los condensadores C1 y C2 deberán ser iguales. Trabaja a frecuencias medias entre 100 KHz a 4Mhz.

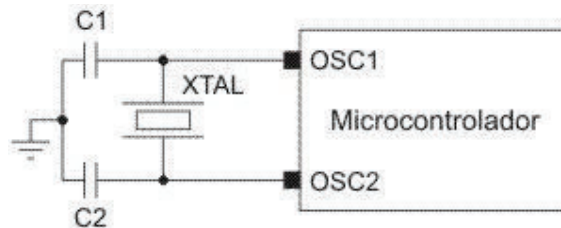


Figura 2.6 Oscilador XT

- Oscilador HS.- Es un oscilador de cristal de cuarzo resonador cerámico trabaja con aplicaciones de alta frecuencia entre 8Mhz a 20Mhz.
- Oscilador LP. - Cristal para baja frecuencia y bajo consumo de potencia, su rango de trabajo es en un rango de frecuencia de 32Khz a 200Khz aproximadamente.

Cabe mencionar que los osciladores tipo LP, XT y HS utilizan un oscilador externo como una fuente de reloj cuya frecuencia está determinada por un cristal de cuarzo o por resonadores cerámicos como se explica en la tabla 2.1 (Angelica, 2010)

Tabla 2.1 Tabla de cristal oscilador y capacitores

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

2.2.3.2 Recursos especiales

2.2.3.2.1 Temporizadores

Los temporizadores son circuitos sincrónicos para el conteo de los pulsos que llegan a su entrada de reloj. Los temporizadores son empleados para controlar periodos de tiempos, mientras que los contadores o timers llevan la cuenta de acontecimientos que suceden en el exterior.

Para la medida de tiempos se carga un registro con un valor adecuado y a continuación dicho valor se va aumentando o disminuyendo al ritmo del pulso del reloj hasta que se desborde y llegue a cero, momento en el que se produce un aviso.

Si la fuente de conteo es el oscilador interno del microcontrolador es común que no tengan un pin asociado, y en este caso trabajan como temporizadores. Por otra parte, cuando la fuente de conteo es externa, entonces tienen asociado un pin configurado como entrada, este es el modo contador. (S, 2004)

2.2.3.2.2 Perro guardián

Debido a que el microcontrolador funciona sin el control de un supervisor y de forma continua. El Perro Guardián o conocido también como Watch Dog Timer consiste en un contador que, cuando llega al máximo, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que resetee al perro guardián de vez en cuando antes de que provoque el reset. Si falla el programa o se bloquea (si cae en bucle infinito), no se refrescará al perro guardián y, al completar su temporización, provocará el reset del sistema. (S, 2004)

2.2.3.2.3 Estado de reposo o bajo consumo

Permite que el microcontrolador entre a un estado pasivo donde consume muy poca potencia. Ya que en ocasiones el microcontrolador debe esperar sin hacer nada a que algún proceso lo vuelva a entrar en funcionamiento.

2.2.3.2.4 *Convertidor A/D y D/A*

Los microcontroladores que incorporan un conversor A/D procesan señales analógicas. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde los pines del circuito integrado.

Al contrario de los convertidores A/D, los conversores D/A realizan el proceso contrario transformando los datos digitales en datos analógicos. (S, 2004)

2.2.3.2.5 *Comparadores analógicos*

Algunos modelos de microcontroladores disponen internamente de un amplificador operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por uno de los pines. La salida del comparador proporciona un nivel lógico 1 o 0 según si una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores. (S, 2004)

2.2.3.2.6 *Puertos digitales de E/S*

Todos los microcontroladores destinan parte de sus pines a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando puertos. Las líneas digitales de los puertos pueden configurarse como Entrada o como Salida cargando un 1 o un 0 en el bit correspondiente de un registro destinado a su configuración.

Además, el microcontrolador por tener la posibilidad de comunicarse con otros dispositivos externos, buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo normas y protocolos disponen de recursos que permiten esta tarea, estos son:

- UART. - Adaptador de comunicación serie asíncrona. Puerto Serie
- USART. - Adaptador de comunicación serie síncrona y asíncrona.
- Puerto paralelo esclavo. - Para poder conectarse con los buses de otros microprocesadores.
- USB. - Universal Serial Bus que es un moderno bus serie para los PC.

- Bus I2C.- Es un interfaz serie de dos hilos desarrollado por Philips.

2.3 HERRAMIENTAS PARA EL DESARROLLO DE LAS APLICACIONES

Las herramientas para el desarrollo de las diferentes aplicaciones que se pueden hacer mediante el microcontrolador son un conjunto de herramientas de hardware y software utilizados para:

- Desarrollar el programa.
- Transferirlo a la memoria del Microcontrolador.
- Verificar su funcionamiento.

2.3.1 HERRAMIENTAS DE HARDWARE

Facilitan la prueba real de los programas y el desarrollo de prototipos en aquellos casos en los que no es imprescindible minimizar la complejidad del hardware.

2.3.1.1 Placas de prototipos

Conocidos también como entrenadores son sistemas electrónicos de los que forma parte el microcontrolador. Con un conjunto de periféricos adecuados se puede realizar operaciones de entrada/salida. Un ejemplo de la placa de prototipos es SiDePIC- US. (Pérez, 2007)

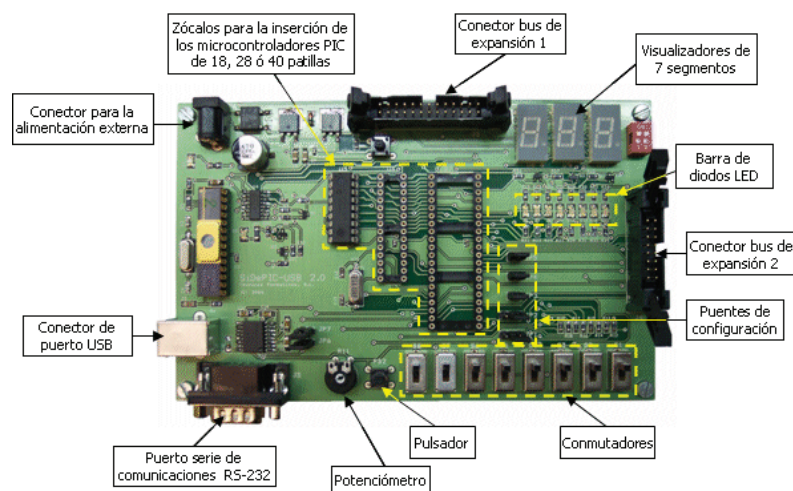


Figura 2.7 SiDePIC USB

Fuente: (Pérez, 2007)

2.3.1.2 Grabadores o programadores

Son sistemas electrónicos que transfieren el programa a la memoria de instrucciones del microcontrolador. Suelen estar asociados a un programa grabador que se ejecuta en un computador personal. Se debe tener en cuenta las siguientes características:

- Cuando la memoria de instrucciones es tipo EPROM son sistemas independientes que poseen un zócalo en el que se inserta el microcontrolador.
- Cuando la memoria de instrucciones es de tipo FLASH, el grabador puede ser también independiente o estar incorporado en el sistema de desarrollo.

2.3.1.3 Depuradores en el sistema

Son sistemas que facilitan la prueba real de un sistema digital basado en un microcontrolador.

Con los depuradores en el sistema se puede grabar y borrar el programa del microcontrolador innumerables veces y probar de esta forma diferentes programas inmediatamente. Las dos herramientas de hardware descritas anteriormente también nos sirven como depuradores. (Pérez, 2007)

2.3.2 HERRAMIENTAS DE SOFTWARE

Las herramientas de software facilitan el diseño de programa, la verificación de su funcionamiento y su transferencia a la memoria del PIC.

2.3.2.1 Editores y compiladores

Para el desarrollo del programa del microcontrolador se utilizan programas editores que constituyen herramientas que facilitan la utilización de los lenguajes de programación.

Los compiladores son programas que se encargan de traducir el programa escrito en un lenguaje de alto nivel a lenguaje máquina. (Pérez, 2007)

2.3.2.2 Simuladores

Son programas capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores brindan la información sobre la ejecución de un programa y son ideales para la depuración de los mismos. (Pérez, 2007)

2.4 INTERFACES

Las Interfaces permiten la comunicación entre el microcontrolador y los dispositivos externos.

2.4.1 PUERTO PARALELO

Los puertos paralelos pueden ser de entrada o de salida se los emplea para lectura de datos y control de dispositivos. Transmiten la información byte por byte, es decir que los 8 bits de datos que forman un byte viajan juntos.

Un puerto de entrada en paralelo es un circuito electrónico que contiene los elementos necesarios para que los datos provenientes del periférico de entrada se transfieran en paralelo a la memoria del procesador digital.

Un puerto de salida en paralelo es un circuito electrónico que contiene los elementos necesarios para que los resultados de proceso llevado a cabo por el procesador digital se transfieran en paralelo a los periféricos de salida.

2.4.2 PUERTO SERIE

Este periférico está presente en casi cualquier microcontrolador, normalmente en forma de UART o USART. El destino común de este periférico es la comunicación con otro microcontrolador o con una PC en donde la información es transmitida bit a bit enviando un solo bit a la vez.

En la actualidad los puertos series son muy utilizados debido a su alta velocidad, menor cableado y son más baratos ya que usan la técnica del par trenzado.

2.4.3 BUS UNIVERSAL EN SERIE

Conocido como USB es una interfaz que provee un estándar de bus serie para conectar dispositivos a un PC, con una distancia de hasta 5 metros.

Fue creado en 1996 por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC. Fue diseñado para estandarizar la conexión de periféricos como mouse, teclados, memorias, cámaras digitales, etc. Este incluye la transmisión de energía eléctrica al dispositivo conectado.

Además, con USB se elimina la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y mejorar las capacidades plug and play permitiendo a los dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar.

2.4.3.1 Características de transmisión

En la tabla 2.2 se presenta la distribución de pines del USB. (Castro J.)

Tabla 2.2 Características de transmisión USB

Pin	Nombre	Color del cable	Descripción
1	VCC	Rojo	+5v
2	D-	Blanco	Data -
3	D+	Verde	Data +
4	GND	Negro	Masa

Las señales del USB se transmiten en un cable de par trenzado con impedancia característica de $90 \Omega \pm 15\%$, cuyos hilos se denominan D+ y D-. Éstos, colectivamente, utilizan señalización diferencial en half dúplex excepto el USB 3.0 que utiliza un segundo par de hilos para realizar una comunicación en full dúplex.

La razón por la cual se realiza la comunicación en modo diferencial es que reduce el efecto del ruido electromagnético en enlaces largos. D+ y D- suelen operar en conjunto y no son conexiones simples.

Este puerto sólo admite la conexión de dispositivos de bajo consumo, es decir, que tengan un consumo máximo de 100 mA por cada puerto. Con la primera fabricación de un PC con USB 3.0 tenemos 1 A por puerto, lo cual da 5 W en lugar de 0,5 A (500 mA, 2,5W) como máximo. (Fonquernie, 2012)

2.4.3.2 Velocidad de transmisión

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- Baja velocidad 1.0.- Tasa de transferencia de hasta 1,5 Mbit/s (192 Kb/s). Utilizado en su mayor parte por dispositivos como teclados, mouse, cámaras web, etc.
- Velocidad completa 1.1.- Tasa de transferencia de hasta 12 Mbit/s (1,5 Mb/s), esta fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad.
- Alta velocidad 2.0.- Tasa de transferencia de hasta 480 Mbit/s (60 MB/s). El cable USB 2.0 dispone de cuatro líneas, un par para datos, y otro par de alimentación.
- Súper alta velocidad 3.0.- Tiene una tasa de transferencia de hasta 4,8 Gbit/s (600 MB/s). La velocidad del bus es diez veces más rápida que la del USB 2.0. En octubre de 2009 la compañía taiwanesa ASUS lanzó la primera placa base que incluía puertos USB 3.0, tras ella muchas otras le han seguido y actualmente se ve cada vez más en placas base y portátiles nuevos, conviviendo junto con el USB 2.0. (Fonquernie, 2012)

2.4.3.3 Tipos de conectores

Existen cuatro tipos de conectores USB como los siguientes:

- Tipo A.- Es el estándar USB que se utiliza normalmente, se trata de un conector plano de forma rectangular. Se lo conecta generalmente en el puerto de un computador o de un hub.

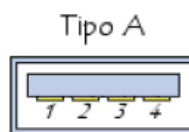


Figura 2.8 Conector USB tipo A

Fuente: (CCM)

- Tipo B.- Los conectores tipo B tienen forma cuadrada y se utilizan en dispositivos que poseen alta velocidad como discos duros externos también se lo utiliza en scanner, impresoras, etc.



Figura 2.9 Conector USB tipo B

Fuente: (CCM)

En la figura 2.9 se muestra los pines de conexión que se mencionó anteriormente en la tabla de 2.2

- Mini USB. - Son versiones pequeñas de los conectores tipo A y B. Ambos son comunes en las cámaras digitales y reproductores de MP3.



Figura 2.10 Conector mini USB

Fuente: (Culturación)

- Micro USB. - Este tipo de conectores son más pequeños que el mini USB. Es usado en teléfonos móviles, cámaras, etc.



Figura 2.11 Conector micro USB

Fuente: (SuFiiAn, 2010)

CAPÍTULO III

IMPLEMENTACIÓN DEL CONTROLADOR LÓGICO PROGRAMABLE

3.1 INTRODUCCIÓN

Con el avance tecnológico que se presenta en el día a día, es de suma importancia mantenernos a la vanguardia de la misma. Los Controladores Lógicos Programables no son la excepción, estos han ido mejorando de acuerdo a las necesidades existentes en el mundo.

Los PLC tienen varias utilidades principalmente en lo que se refiere a la industria, es decir, a través de los mismos se puede automatizar desde un pequeño negocio hasta grandes empresas, pero para lograr su máximo potencial es necesario llevar a cabo el estudio de las diferentes características de hardware y software, sus funciones y posibles aplicaciones.

3.2 ESPECIFICACIONES DE LOS REQUISITOS PARA LA CONSTRUCCIÓN DEL PLC

Para que el PLC a desarrollarse tenga una capacidad similar a los existentes en el mercado como es el caso de Logo que es fabricado por Siemens, debe cumplir con los siguientes requisitos mínimos:

3.2.1 REQUERIMIENTOS DEL SISTEMA

El PLC está diseñado para cumplir con lo siguiente:

- 8 Entradas digitales.
- 4 Salidas digitales.
- Temporizadores on delay.
- Comunicación USB.
- Software de Programación.
- Operaciones Lógicas

3.3 DIAGRAMA DE BLOQUES

De manera general el funcionamiento de un PLC es el siguiente:

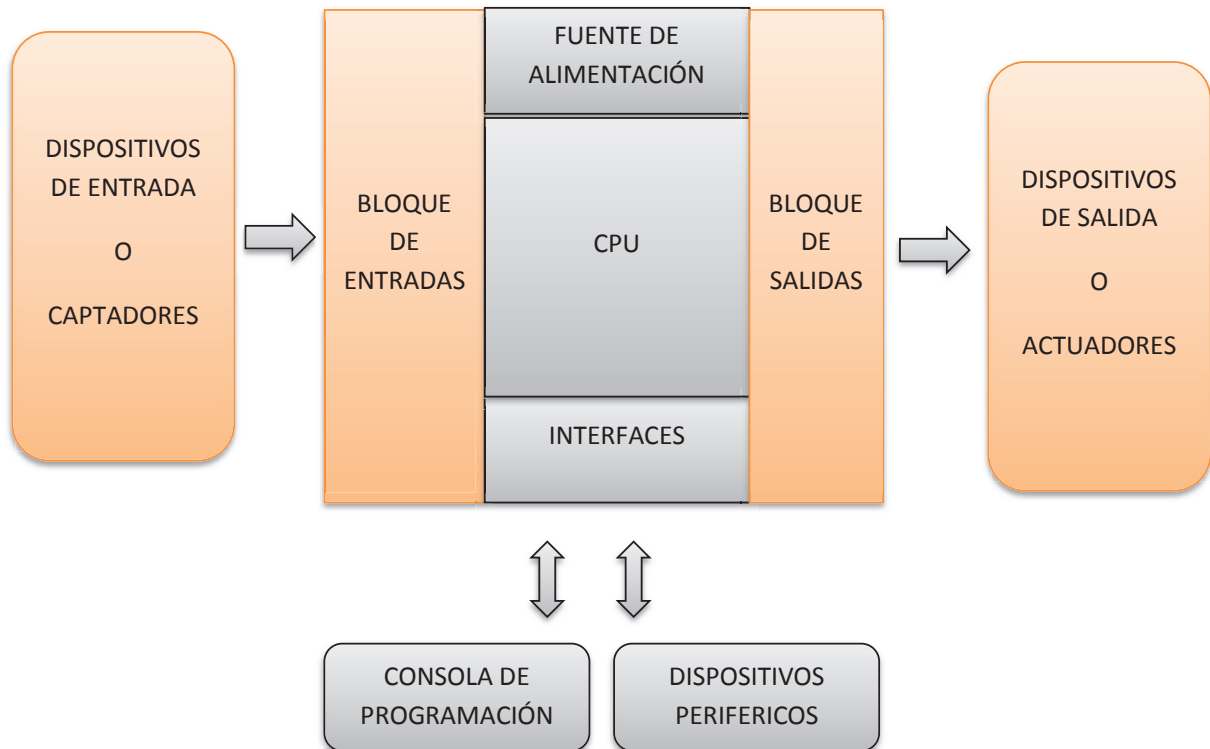


Figura 3.1 Diagrama de funcionamiento

3.4 SELECCIÓN DE LOS COMPONENTES

A continuación, se presenta los distintos dispositivos que conforman el PLC, con sus respectivas características, distribución de pines y su funcionalidad en el circuito.

3.4.1 MICROCONTROLADOR 18F4550

El PIC 18F4550 de Microchip pertenece a la gama mejorada de microcontroladores, con arquitectura RISC avanzada Harvard de 16 bits con 8 bits de datos, cuenta con set de 77 instrucciones, y un número de pines desde 18 a 80. Además, posee una memoria RAM de hasta 3968 bytes y 1KBytes de EEPROM, con una pila de 32 niveles que permite el anidamiento de subrutinas, como también tiene periféricos de comunicación CAN y USB. El PIC18F4550 dispone buses diferentes para el acceso a memoria de programa y memoria de datos (arquitectura Harvard).

Sus características más importantes se presentan en la tabla 3.1 (Inc, 2009) (MIKROELEKTRONIKA)

Tabla 3.1 Características PIC 18f4550 y PIC 16f887

CARACTERÍSTICAS	PIC 18F4550	PIC 16F887
Frecuencia de operación	Hasta 48 MHz	0-20 MHz
Memoria de programa (bytes)	32.768	Flash 8K
Memoria RAM de datos (bytes)	3968	368
Memoria EEPROM datos (bytes)	1Kb	256
Interrupciones	20	11
Líneas de E/S	35	35
Temporizadores	4	3
Módulos de comparación/PWM (CCP)	1	1
Módulos de comparación/PWM mejorado (ECCP)	1	1
Canales de comunicación serie	MSSP, EUSART	USART, MSSP
Canal USB	1	-
Puerto paralelo de transmisión de datos (SPP)	1	-
Canales de conversión A/D de 10 bits	13 canales	14 canales
Comparadores analógicos	2	2
Juego de instrucciones	75 (83 ext.)	35
Encapsulados	PDIP 40 pines QFN 40 pines TQFP 40 pines	DIP 40 pines

A diferencia de los microcontroladores de la familia 16FXXX como se observa en la tabla 3.1, presentan características inferiores tanto en frecuencia de operación como en memoria RAM, Flash y EEPROM. El microcontrolador 18F4550 permite una mayor velocidad de operación, memoria EEPROM con 1'000.000 de ciclos de lectura/escritura y retención de datos de hasta 40 años, además tiene como principal característica la comunicación USB en versión 2.0.

3.4.1.1 Diagrama de Pines

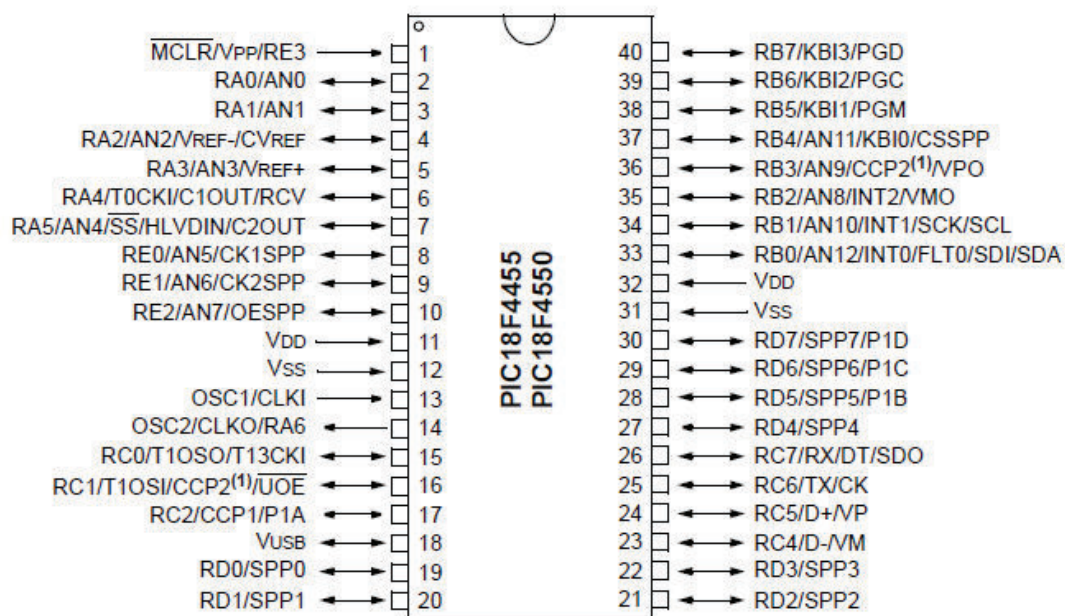


Figura 3.2 Diagrama pines PIC 18f4550

Fuente: (Inc, 2009)

En la figura 3.2 se presenta la distribución de pines del PIC 18F4550, el puerto A y puerto B se pueden utilizar como entradas o salidas digitales. Para la programación del PLC se utilizó el puerto A para las salidas digitales desde RA0 a RA3, mientras que para las entradas digitales se utilizó el puerto B desde RB0 a RB7.

El puerto C RC6 y RC7 son los pines que permiten la comunicación con la pantalla TFT. (Véase Anexo 5)

Para que se efectúe la programación ICSP se tiene las siguientes conexiones con el microcontrolador: (Véase Anexo 5)

- ICSPDAT o PGD. - Datos de Programación es una línea de datos bidireccional sincrónico serial. Se encuentra conectado al pin 40.
- ICSPCLK o PGC. - Reloj de Programación es una línea unidireccional sincrónica serial de reloj que va desde el programador hasta el microcontrolador. Pin 39
- VPP. - Voltaje de Programación cuando es aplicado, el microcontrolador entra en el modo Programación. Configurado en el pin 1.
- VDD. - Suministro de voltaje positivo.
- VSS. - Suministro de voltaje negativo.

3.4.1.2 Memorias

El PIC18F4550 dispone de las siguientes memorias:

- Memoria de programa: memoria flash interna de 32.768 bytes
 - Almacena instrucciones y constantes/datos
 - Puede ser escrita/leída mediante un programador externo o durante la ejecución del programa mediante unos punteros.
- Memoria RAM de datos: memoria SRAM interna de 3968 bytes en la que están incluidos los registros de función especial.
 - Almacena datos de forma temporal durante la ejecución del programa
 - Puede ser escrita/leída en tiempo de ejecución mediante diversas instrucciones.
- Memoria EEPROM de datos: memoria no volátil de 1Kb.
 - Almacena datos que se deben conservar aun en ausencia de tensión de alimentación.
 - Puede ser escrita/leída en tiempo de ejecución a través de registros
- Pila: bloque de 31 palabras de 21 bits.
 - Almacena la dirección de la instrucción que debe ser ejecutada después de una interrupción o subrutina.
- Memoria de configuración: memoria en la que se incluyen los bits de configuración (12 bytes de memoria flash) y los registros de identificación (2 bytes de memoria de solo lectura).

El PIC18F4550 en el presente proyecto es el dispositivo fundamental para el funcionamiento del PLC, ya que realiza lo siguiente:

- Recibe los estados de las entradas.
- Permite la comunicación USB entre el PLC y el computador.
- Almacena el programa del usuario en la memoria EEPROM.
- Refleja el estado de cada entrada y salida en la pantalla TFT.
- Cambia el estado de las salidas de acuerdo al programa de usuario.

En lo que respecta a la programación del PIC 18F4550, esta fue realizada en el compilador PIC Basic PRO utilizando MicroCode Studio, en esta consta la configuración de los pines y las operaciones lógicas and, or, not, xor. (Véase en el Anexo 1 y Anexo 2).

3.4.2 TFT LCD

TFT-LCD (Thin Film Transistor-Liquid Crystal Display / Pantalla de cristal líquido de transistores de película fina) es una variante de la pantalla de cristal líquido (LCD) que usa tecnología de transistor de película delgada (TFT). Tienen una matriz extra de transistores (un transistor por cada color de cada pixel) de esta manera se logra eliminar los problemas de pureza de color, contraste y la velocidad de respuesta a la renovación de las imágenes.

Este tipo de pantallas son utilizadas en televisores, visualizadores de pantalla plana y proyectores.

3.4.2.1 TFT Arduino

La pantalla TFT de Arduino es una pantalla LCD retro iluminada con encabezados. Es capaz de dibujar texto, imágenes y diferentes formas en la pantalla con la ayuda de la biblioteca de TFT. Gracias a la integración de una ranura para tarjetas micro-SD en la parte posterior de la pantalla, se puede almacenar las imágenes de mapa de bits.

Su pantalla es de 10" de diagonal, con una resolución de 800 x 480 píxeles. La pantalla funciona con 5 VDC.

A continuación, en la figura 3.3 se presenta la distribución de pines de la pantalla TFT.



Figura 3.3 Diagrama pines TFT Arduino
Fuente: (Arduino)

La función de la pantalla TFT en el PLC es visualizar el estado de las distintas entradas y salidas. Su programación se encuentra en el Anexo 4.

La TFT cuenta con una memoria microSD, en la cual se encuentra almacenada las imágenes que se muestran en la pantalla.

3.4.3 RELÉ

El relé es un dispositivo electromagnético. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

La gran ventaja de los relés electromagnéticos es la separación eléctrica entre la corriente de accionamiento, la que circula por la bobina del electroimán, y los circuitos controlados por los contactos, lo que hace que se puedan manejar altos voltajes o elevadas potencias con pequeñas tensiones de control. También ofrecen la posibilidad de control de un dispositivo a distancia mediante el uso de pequeñas señales de control.

En el presente caso se tendrá un grupo de relés que serán controlados mediante un PIC que permite crear funciones de temporización, junto con otras funciones lógicas, actuando como el más básico de los PLC's.

Con estos modernos sistemas los relés pueden actuar de forma programada e independiente lo que supone grandes ventajas en su aplicación aumentando su uso

dentro del área de control y automatización. Se puede encender por ejemplo una bombilla o motor y al encenderlo se apaga el otro motor o bombilla, esto gracias a que cuenta con dos tipos de salidas, una normalmente abierta (NA) y otra normalmente cerrada (NC).

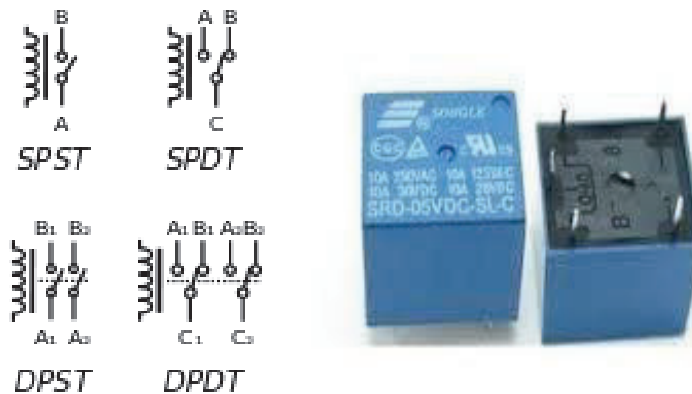


Figura 3.4 Simbología del relé

Fuente: (electronics)

En este caso se seleccionó un relé tipo 05vdc-sl-c que tiene las siguientes características:

- Capacidad de conmutación disponible 10A pesar de su diseño de tamaño pequeño
- Selección de material plástico para alta temperatura y mejor rendimiento de la solución química.
- Capacidad de contacto 7A 28VDC
- Carga resistiva ($\cos = 1$) 125VAC 10A, 240VAC 7A
- Carga inductiva ($\cos = 0,4$ L / R = 7msec) 120VAC 3A, 3A 28VDC
- Max. Tensión Admisible 250VAC / 110VDC
- Max. Potencia admisible 800VAC / 240W
- Max. Conmutación mecánica 300 operaciones / min
- Max. Conmutación eléctrica 30 operaciones / min

Además, estos relés son de bajo costo, fácil implementación, presentan aislamiento entre el circuito de mando y los contactos de ejecución y tienen reducida caída de tensión en contactores cerrados.

3.4.4 OPTOACOPADORES

Un optoacoplador, o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo LED que satura un componente opto electrónico, normalmente en forma de fototransistor o fototriac. De este modo se combinan en un solo dispositivo semiconductor, un foto emisor y un foto receptor cuya conexión entre ambos es óptica. Se suelen utilizar para aislar eléctricamente a dispositivos muy sensibles.

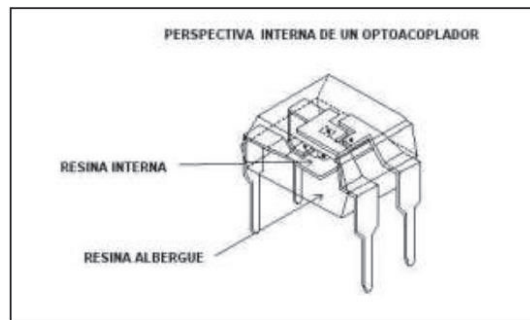


Figura 3.5 Diagrama interno optoacoplador

Fuente: (Castro J. M., 2007)

La corriente de salida I_C (corriente de colector del fototransistor) es proporcional a la corriente de entrada I_F (corriente en el diodo LED). La relación entre estas dos corrientes se llama razón de transferencia de corriente (CTR) y depende de la temperatura ambiente

La ventaja fundamental de un optoacoplador es el aislamiento eléctrico entre los circuitos de entrada y salida. Mediante el optoacoplador, el único contacto entre ambos circuitos es un haz de luz. Esto se traduce en una resistencia de aislamiento entre los dos circuitos.

En el presente proyecto se utilizó un optoacoplador tipo PC817 el cual permite hacer un aislamiento entre dos etapas del circuito. Este fue seleccionado por tener las siguientes características. (Véase Anexo 7)

- Voltaje de Aislamiento: 5000Vrms
- V_{ceo} (voltaje colector – emisor): 80V
- I_f (corriente directa): 50mA
- I_c (corriente de colector): 50mA

- Temperatura de funcionamiento: -30 a +100 °C
- Canales: 1
- Entrada: tipo DC
- Salida: Tipo Transistor

Además de las características ya mencionadas, su bajo costo y su tamaño fueron relevantes en el momento de hacer la selección del dispositivo.

3.4.5 CONVERTIDOR USB UART RS232RL

El estándar RS-232 o TIA/EIA-232, nació debido a la necesidad de poner un alto a la cantidad de protocolos que cada empresa hacía para comunicar sus máquinas, pues esto imposibilitaba el flujo de información. Sus principales características son:

- Los voltajes para un "1" lógico deben estar entre -3V a -15V y para un "0" lógico deben ser de entre +3V a +15V.
- Por lo regular 25 pines de señal.
- Los voltajes más usados están entre $\pm 12V$ y $\pm 9V$.
- Cubre distancias de hasta 15m, dependiendo de la velocidad de transmisión.
- Por lo general, las velocidades de transmisión están desde los 300 Kbps hasta los 9600Kbps.

El FT232 es un circuito integrado que básicamente cumple la función de pasar del protocolo USB al serial, diseñado con la finalidad de minimizar los componentes externos necesarios para este mismo objetivo, además de eliminar la parte de programación, ya que simplemente requiere polarizarlo, agregar ciertos componentes para su correcto funcionamiento y listo. Según su hoja de datos cuenta con las siguientes características: (Véase Anexo 7)

- Hace la conversión de protocolos directamente de USB a UART, para la transmisión de datos.
- El protocolo USB viene integrado sin la necesidad de ser programado o actualizado.
- Incluye una memoria EEPROM de un 1 Mb con la información necesaria para ser configurada e identificada el dispositivo.
- No requiere cristal externo debido a que tiene incluido un reloj interno.

- Tiene una tasa de transmisión de datos desde 300 baudios hasta 3 Mbaudios en niveles TTL.
- Posee 256 bytes de buffer de recepción y 128 bytes de buffer de transmisión que utiliza para mejorar el rendimiento en el instante de transferir una importante cantidad de datos.
- Los drivers de este circuito no tienen derechos de autor, con el objetivo de permitir el desarrollo de proyectos con USB sin restricción alguna.
- Se pueden conectar LED indicadores para verificar el estado de la transmisión.
- El dispositivo tiene pre programado un único número serial USB.
- Capaz de trabajar con la alimentación brindada del puerto USB o con una fuente independiente.
- Para compatibilidad, pasa valores de voltaje de +3.3V a niveles USB.
- Permite invertir la señal del UART si se desea.
- Consume poco ancho de banda del USB.
- Capaz de trabajar en bajo consumo.
- Compatible con el protocolo USB 2.0.
- No tiene dificultad alguna de operar en temperaturas desde -40°C hasta 85°C .

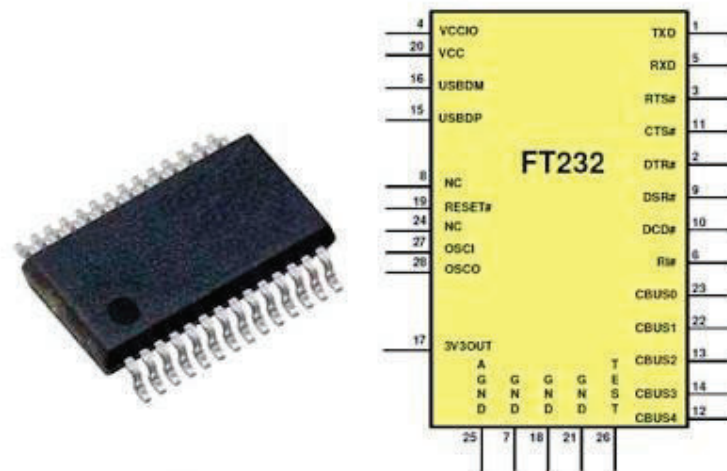


Figura 3.6 Diagrama esquemático del integrado FT232

Fuente: (Castro C. , 2012)

Se utilizó el FT232 para ayudar a la comunicación entre el computador y el PLC, debido a que en un principio la comunicación USB en forma directa presentó algunas fallas al momento de la transmisión de datos.

3.4.6 FUENTE DE ALIMENTACIÓN

La función de la fuente de alimentación es transformar la tensión alterna en tensión continua lo más estable posible, para ello, es necesario utilizar los siguientes dispositivos:

- Transformador de Entrada.
- Rectificador de Diodos.
- Filtro
- Regulador Lineal

3.4.6.1 Transformador de entrada

Los aparatos electrónicos existentes en el mercado trabajan con un voltaje mucho menor que el presente en las tomas corrientes de casas, empresas, fábricas, etc. (generalmente 110 / 220 Voltios de AC).

El transformador de la fuente de alimentación reduce la tensión de la red a una más adecuada, es decir, convierte la tensión AC a un nivel de voltaje más apropiado para las necesidades del circuito.

3.4.6.2 Rectificador de diodos

La etapa de rectificación se lleva acabo con la ayuda de tres tipos de circuitos que son:

- El Rectificador de media onda
- El Rectificador de onda completa
 - Rectificador de onda completa con tap central
 - Rectificador de onda completa tipo puente

En este caso se ha escogido el Rectificador de Onda Completa con puente de diodos. Mediante el uso de 4 diodos se elimina la necesidad de la conexión intermedia en el secundario del transformador.

3.4.6.3 Filtro

Para completar el proceso de transformación de la corriente alterna a continua, se adiciona un circuito llamado filtro de rectificación. Este circuito es formado por elementos que almacenan la energía como condensadores y bobinas.

3.4.6.4 Regulador de voltaje

Para que los diferentes dispositivos como el microcontrolador, TFT, etc. trabajen correctamente, se necesita un voltaje de 5 Voltios de corriente continua, y para ello se utiliza el regulador de voltaje 7805.

Este regulador se encarga de que el voltaje de salida no varíe de su valor nominal, obteniendo 5V constantes en la salida de voltaje.

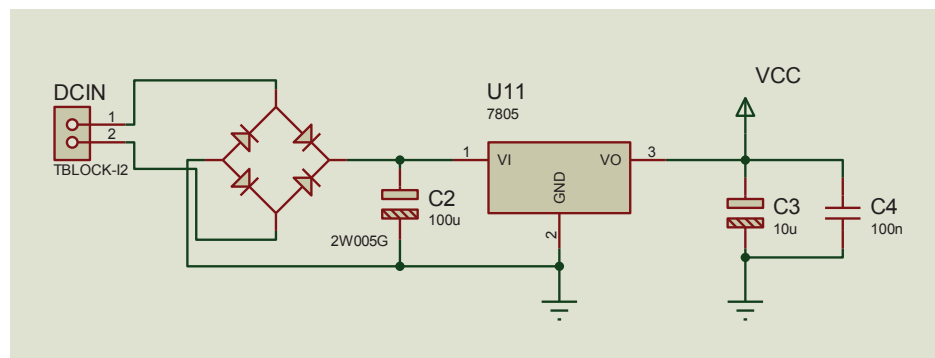


Figura 3.7 Diagrama fuente de alimentación

En la figura 3.7 se muestra el diagrama de conexión de la fuente de alimentación en donde:

- DCIN es la entrada de voltaje DC
- Puente de diodos: Conectado a DCIN ayuda a proteger a los capacitores y al regulador de voltaje en caso de que se conecte de manera incorrecta en DCIN
- Capacitor C2: Filtra el voltaje que pasa por el puente de diodos
- 7805: Es el regulador de voltaje el cual consta de 3 pines de los cuales el pin 1 recoge el voltaje que ha sido filtrado, el pin 2 se conecta a tierra para tener un

voltaje de referencia, el pin 3 proporciona los 5 voltios que se necesitan para alimentar a la placa del PLC y a la pantalla TFT.

- Capacitores C3 y C4: actúan como filtros eliminando cualquier residuo de ruido antes que el voltaje VCC alimente a los dispositivos

3.5 DISEÑO Y CONSTRUCCIÓN DEL PLC

3.5.1 DISEÑO DEL PLC

Para el diseño del PLC se tomó como base el siguiente diagrama de flujo.



Figura 3.8 Diagrama de flujo PLC

3.5.2 CONSTRUCCIÓN DEL PLC

El PLC está formado por los siguientes elementos:

1.- Ocho entradas digitales (IN1-IN8) aisladas con opto acopladores de corriente continua, como protección de las mismas, consta de un transistor y un diodo zener. Adicional a esto, se encuentra un led que indica la activación de cada una de las entradas.

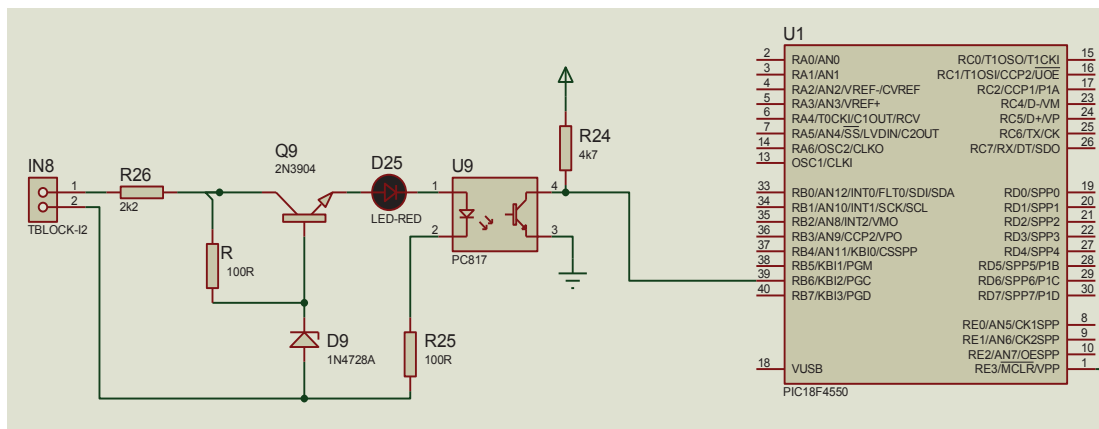


Figura 3.9 Diagrama entradas digitales

En la figura 3.9 se puede apreciar el diagrama de conexión para cada una de las entradas en donde:

- TBLOCK-12 (IN8). - Este es un borne de dos puntos donde se receptorá una señal en DC que pueda activar la entrada.
- Transistor 2N3904.- Actuando en conjunto con un diodo zener 1N4728A (D9), el cual se encarga de supervisar la tensión suministrada por la fuente, mientras el transistor se encuentra en modo de colector común; este conjunto nos permitirá tener un valor de voltaje constante, aun cuando se varíe el voltaje de la entrada, protegiendo el diodo led y el optoacoplador.
- Diodo LED (D25). - Actúa como indicador de que la corriente está circulando y que la entrada se encuentra activada.
- Optoacoplador PC817.- Aísla eléctricamente el circuito de entrada de la entrada física del PIC.

- R24, R25, R26.- Son resistencias que ayudan a que pase el voltaje y la corriente necesaria a cada uno de los elementos del circuito
- VCC: Es el voltaje de alimentación que pasara por el optoacoplador y activara la entrada del PIC.

2.- Cuatro Salidas Digitales (OUT1- OUT4) aisladas con opto acopladores de corriente continua. Las salidas son a relé de 5V, que pueden ser NA y NC, estas se activan mediante un transistor y al igual que las entradas digitales tienen un led que indica la activación de las salidas.

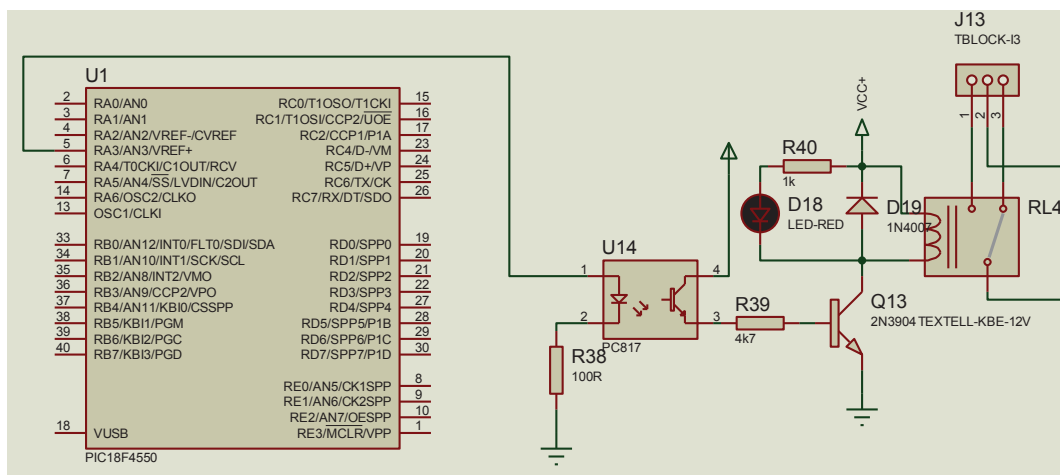


Figura 3.10 Diagrama salidas digitales

En la figura 3.10 se puede observar la conexión de las salidas del PLC con cada uno de sus componentes:

- Optoacoplador PC817.- Aísla eléctricamente el circuito de salida de la salida física del PIC, el optoacoplador se activa con la señal que envía la salida del PIC out4. La corriente máxima es de 50 mA.
- Transistor 2N3904.- Funciona en modo corte – saturación permitiendo que se active el relé cuando este se satura. La corriente en colector I_c es de 50 mA.
- Relé RL4.- Tienes 2 posiciones una normalmente abierta (NA) y otra normalmente cerrada (NC), las salidas del relé soportan entre 120 VAC y 240 VAC con un máximo de 10 A, y hasta un máximo de 25 VDC y 10 A, estas se encuentran unidas a un juego de bornes J13, donde se pueden

conectar el dispositivo que se quiera accionar. La corriente nominal que necesita la bobina es de 150 mA.

- Bornera J13.- Esta bornera tiene 3 terminales donde se conectarán las salidas del relé, la salida de la mitad será el punto común y las de los extremos serán la una NC y la otra NA.
- Diodo LED D18.- Es un led indicador que permite saber si la salida está funcionando correctamente.
- VCC. - Es la entrada de voltaje que proporciona la fuente de alimentación para accionar cada uno de los elementos del circuito.

Al igual que en las entradas el circuito consta de resistencias de diferente valor para que exista el paso de voltajes y corrientes adecuadas que permitirán el funcionamiento correcto del mismo.

3.- Temporizadores On Delay pueden trabajar desde 1000ms hasta 1 hora, en pasos de 1s según sea el caso. Este proceso se lo realiza a través del software.

4.- Comunicación USB permite cargar el programa desde el computador hacia la memoria EEPROM del PLC a través de un convertidor USB serial (RS232) el cual se conecta con la entrada serial del PIC que corresponde a los pines 25 y 26 (Tx, Rx). Se utiliza un USB tipo B de 4 pines, de los cuales 2 son para energizar el USB (pin 1 y 4) y los otros dos (pin 2 y 3) se utilizan para la transmisión de información. Los niveles de transmisión de la señal varían de 0 a 0,3 V para bajos (ceros) y de 2,8 a 3,6 V para altos (unos). Los pines 2 y 3 corresponden a D+ y D- respectivamente los cuales se conectan al RS232 a los pines 15 y 16 que corresponden a la entrada USB del mismo para realizar la conversión al protocolo serie.

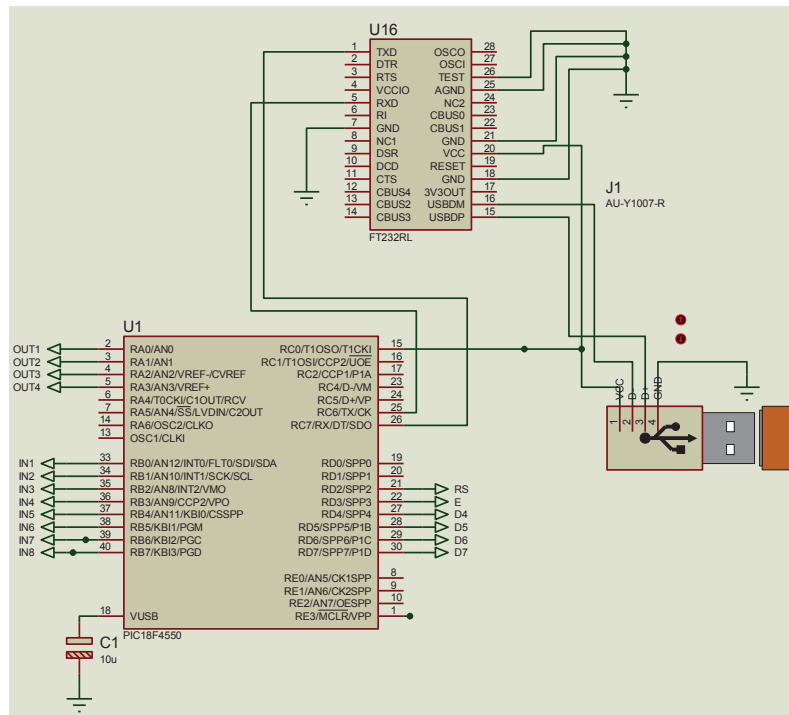


Figura 3.11 Diagrama comunicación USB

Todos los dispositivos integrados tanto en las entradas como en las salidas, al igual que el PIC y la pantalla TFT necesitan un voltaje de alimentación de 5V y una corriente muy baja para trabajar en el orden de los miliamperios (mA).

El controlador lógico programable se lo desarrolló en una sola placa. La misma que se encarga de manejar la programación y el control de las entradas/salidas, como también se encarga de la comunicación con la pantalla TFT.

En la figura 3.12 se puede observar la placa del PLC con todos los dispositivos que lo conforman. El diagrama del circuito, las pistas de la placa se encuentran en el Anexo 5 y Anexo 6

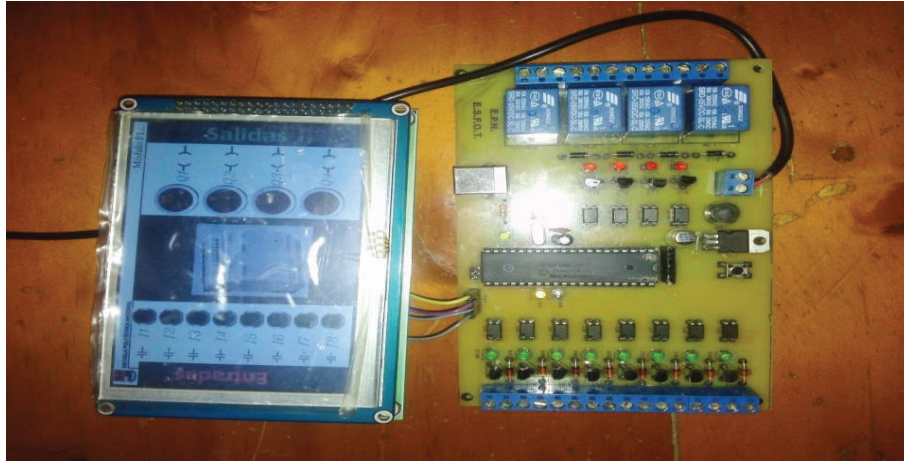


Figura 3.12 Placa PLC

El proyecto se basa en el PLC Logo 24RC de Siemens, por lo cual en la tabla se presenta las características eléctricas.

Tabla 3.2 Características eléctricas PLC

	PLC
Fuente de alimentación	
Tensión de entrada	5 VCC
Rango admisible	4,5...5,5 VCC
Consumo de corriente 5 V DC	50... 1800 mA
Potencia disipada en caso de 5 V	0,25 ...,9 W
Entradas digitales	
Cantidad	8
Separación galvánica	no
Tensión de entrada	
señal 0	<3 V c.c.
señal 1	>5 V c.c.
Salidas digitales	
Cantidad	4
Tipo de las salidas	Salidas a relé
Separación galvánica	sí
En grupos de	1
Activación de una entrada digital	sí
Corriente constante I_{th}	Máx. 10 A por relé
Carga de lámparas incandescentes	1.000 W

3.6 DISEÑO DEL SOFTWARE

3.6.1 DESCRIPCIÓN DEL SOFTWARE

El software del PLC se desarrolló mediante el programa Visual Studio 2012, a través de Windows Forms. Permite la interacción entre cada una de las entradas de dos en dos. Todas las entradas pueden interactuar con todas las salidas si fuese necesario.

Para asociar una serie de funciones entre varias entradas se necesita insertar funciones especiales. En la figura 3.13 se presenta un ejemplo en el cual se utiliza las entradas especiales.

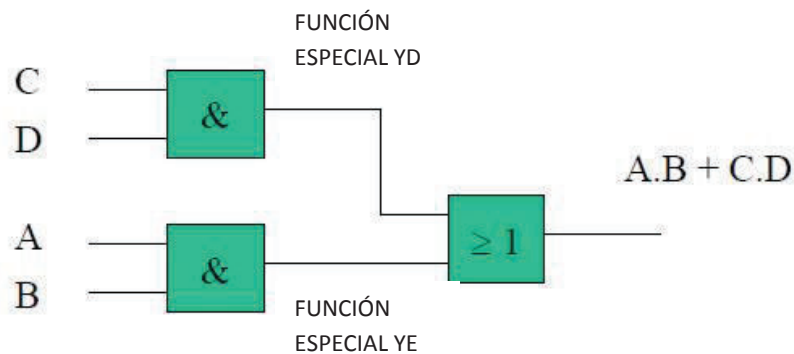


Figura 3.13 Funciones especiales

Como se puede observar se tiene las entradas A y B ingresando a la compuerta and, ese resultado se lo utiliza como una entrada para la función or. Para lograr programar en el interfaz de usuario la salida de la función and pasa a llamarse función especial YE. Lo mismo ocurre con la otra compuerta and cuyas entradas son C y D y su función especial es YD.

Se creó 252 funciones que contienen la relación entre las entradas, entradas y salidas, funciones especiales, funciones lógicas descritas anteriormente, y adicionalmente se tiene un timer que puede funcionar como on delay u off delay según se necesite.

Se estableció un recuadro de comunicación y verificación de los datos que se envían desde el software al PIC.

El software consta de dos partes:

3.6.1.1 Parte de control

En esta sección se encuentra el ingreso de las entradas y el operador o la función que se desee ejecutar, como se puede observar en la figura 3.14. Una vez agregada la operación, se selecciona la salida que se activará. Además, se puede adicionar un temporizador on delay.

Para verificar el ingreso de las funciones, se utiliza la opción visor de funciones, la cual, a través del número del contador, permite una vista previa de las funciones ingresadas.

Si se desea trabajar con un nuevo programa existe la opción de reinicio. Esta opción borra todas las funciones ingresadas anteriormente.

Finalmente, como método de seguridad para el PLC tenemos el ingreso de una clave. Únicamente con ella se podrá enviar datos al PLC para cargar un nuevo programa.

Figura 3.14 Cuadro de control

En la figura 3.14 consta de lo siguiente:

- Entrada 1 Entrada 2.- Se ingresan las entradas que van a ser utilizadas, pueden ser desde In1 hasta In8 y también pueden ingresarse YD y YE, se puede ingresar en mayúsculas o minúsculas.
- Operador. - Se ingresa la función lógica a utilizar.
- Insertar tiempo. - Se puede ingresar un valor para el timer en intervalos de 1 segundo.
- Salidas: Se selecciona la salida que será activada con cada función.
- YD y YE. - Se debe ingresar los números que están presentes en la columna Contador, según sea el caso. Estos representan a las funciones creadas previamente y que pueden interactuar entre sí. Para activar esta casilla de Insertar Especial, se debe ingresar primero en el recuadro de Entrada 1 y Entrada 2 las funciones YD y YE en las entradas, en el operador el tipo de función lógica y presionar el botón Insertar Especial.
- Recuadro Operación (texto). - Muestra las entradas utilizadas y la función lógica que realizan.
- Recuadro Operación (función). - Muestra cuál de las 252 funciones creadas se va a ejecutar y nos ayuda a comprobar que sea la correcta.
- Recuadro Operación (contador). - Lista de una en una todas las funciones ingresadas y ese número se utiliza para asignar la salida o en las funciones especiales.
- Botón reiniciar. - Borra todos los datos de los recuadros

3.6.1.2 Parte de comunicación

En la parte de comunicación se utiliza un convertidor USB - SERIAL RS232 el cual ha sido descrito anteriormente.

La función de la parte de comunicación es seleccionar el puerto al cual está conectado el PLC, como también realiza la selección de la velocidad de transmisión de los datos. Se recomienda una velocidad de 9600 bps.

El botón “Salir” permite la desconexión del computador con el PLC. Si no se selecciona el mismo, el programa puede fallar.

Comunicación

SELECCIONE ANTES DE CONECTAR

COM6

9600

RECEPCION DE DATOS

Cargando...

fin de la Carga

1 1 0 0
22 2 0 0
208 3 1 2
252 2 0 0

Figura 3.15 Cuadro de comunicación

INTERFAZ PLC

ESCUELA POLITÉCNICA NACIONAL

Control

Entrada 1 in1 Operador and Entrada 2 in2

YD 1

YE 2

Ingrese la posicion

Operacion (Texto)	Operacion (Funcion)	Operacion (Contador)
IN1 AND IN2	1	1
IN3 AND IN4	22	2
1 AND 2	208	3
Tiempo: 2	252	2

Visor de funciones

Ingrese la Operacion (Contador)

Funcion Obtenida IN3 AND IN4

Ingreso de Clave

Comunicación

SELECCIONE ANTES DE CONECTAR

COM6

9600

RECEPCION DE DATOS

Cargando...

fin de la Carga

1 1 0 0
22 2 0 0
208 3 1 2
252 2 0 0

Figura 3.16 Interfaz del PLC

3.6.2 DIAGRAMA DE FLUJO DEL SOFTWARE

En la figura 3.16 se presenta el diagrama de flujo del software del PLC

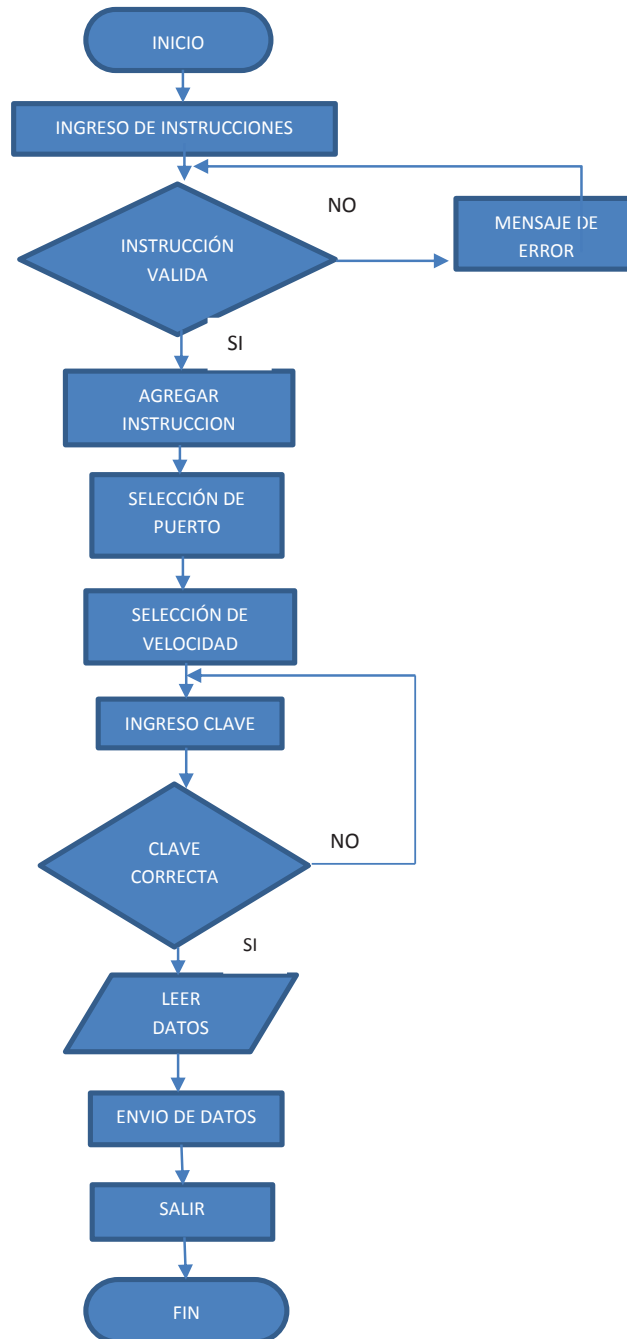


Figura 3.17 Diagrama de flujo del software

3.6.3 INSTRUCCIONES Y VARIABLES

A continuación, se detalla las instrucciones y variables que tiene el PLC.

3.6.3.1 Variables

- Entradas Digitales. - IN1 - IN8
- Salidas Digitales. - OUT1 - OUT4
- Timer. - On Delay
- Funciones Especiales. - YE – YD

3.6.3.2 Instrucciones

- AND. - Multiplicación Lógica
- OR. - Suma Lógica
- XOR. - Or exclusiva
- NOT. - Inversión

CAPÍTULO IV

PRUEBAS EXPERIMENTALES

4.1 DESCRIPCIÓN FÍSICA

El PLC consta de dos partes: La primera es la placa donde se encuentra el PIC, las entradas y salidas, mientras que la segunda parte es la pantalla TFT que permite visualizar el funcionamiento del PLC. Estas dos partes se encuentran interconectadas entre sí.

4.2 PRUEBAS EXPERIMENTALES

Para verificar el correcto funcionamiento del PLC, se llevó a cabo un conjunto de pruebas.

4.2.1 PRUEBAS DE LAS FUNCIONES LÓGICAS

Se ingresó dos variables al programa, ya que el programa está diseñado para ingresar de dos en dos los datos de entradas y se comprobó que estas cumplan con los resultados del álgebra booleana. Esto se puede verificar en las diferentes figuras de cada función que se presenta a continuación.

4.2.1.1 Función and

The screenshot shows the 'INTERFAZ PLC' software interface. At the top, it displays the logo of 'ESCUELA POLITÉCNICA NACIONAL'. The interface is divided into several sections:

- Control:** Contains input fields for 'Entrada 1', 'Entrada 2', and 'Operador'. There are buttons for 'INSERTAR ESPECIAL' and 'INSERTAR TIEMPO'. Below this, there are buttons for 'Ingreso la posicion' and 'Salida 1' through 'Salida 4'.
- Operacion (Funcion):** A table with three columns: 'Operacion (Texto)', 'Operacion (Funcion)', and 'Operacion (Contador)'. The first row shows 'IN1 AND IN2' with '1' in the function and '1' in the counter. The second row shows 'Salida 1' with '248' in the function and '1' in the counter.
- AGREGAR and REINICIAR:** Buttons located below the operation table.
- Visor de funciones:** A section with a 'Ver' button and a field for 'Ingreso la Operacion (Contador)'. Below it is a field for 'Funcion Obtenida'.
- Ingreso de Clave:** A section with a text input field and an 'ACEPTAR' button.
- Comunicacion:** A section titled 'SELECCIONE ANTES DE CONECTAR' with dropdown menus for 'COM6' and '9600', and buttons for 'CONECTAR', 'Refrescar Conexion', and 'REINICIAR tabla'. Below this is a 'RECEPCION DE DATOS' section with a 'Cargando...' status and a table of data:

fn de la Carga
1 1 0 0
248 1 0 0
208 3 1 2
252 2 0 0

 There is also an 'ENVIAR DATO' button and a 'SALIR' button at the bottom right.

Figura 4.1 Programación función and

De acuerdo a la tabla 4.1, la función and únicamente activa la salida cuando las dos entradas tienen el estado 1, es decir, cuando las entradas se encuentran activas.

Tabla 4.1 Tabla lógica función and

Entrada 1	Entrada 2	Salida 1
0	0	0
0	1	0
1	0	0
1	1	1

En la figura 4.2 se comprueba el resultado de la función and en el PLC.

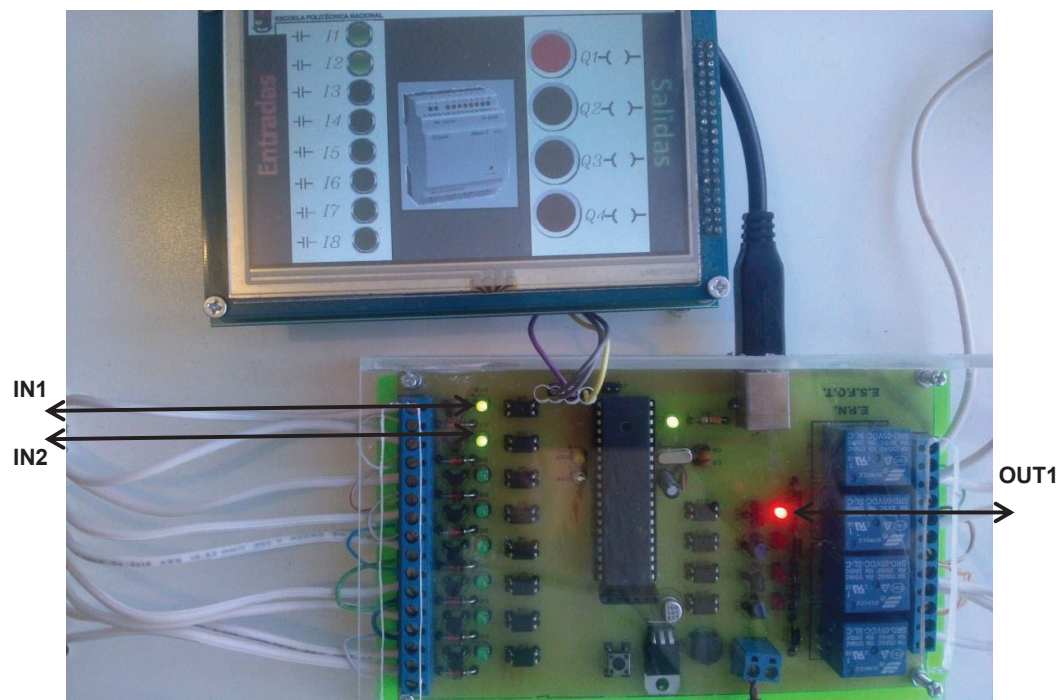


Figura 4.2 Resultado de la función and

4.2.1.2 Función or

En la figura 4.3 se muestra el ingreso de los datos para la prueba con la función OR.

The screenshot shows the 'INTERFAZ PLC' software interface for 'ESCUELA POLITÉCNICA NACIONAL'. The interface is divided into two main sections: 'Control' and 'Comunicacion'.

Control Section:

- Inputs: 'Entrada 1', 'Entrada 2', 'YD', 'YE'.
- Buttons: 'INSERTAR ESPECIAL', 'INSERTAR TIEMPO'.
- Table: 'Operacion' with columns 'Operacion (Texto)', 'Operacion (Funcion)', and 'Operacion (Contador)'. The table contains the following data:

Operacion (Texto)	Operacion (Funcion)	Operacion (Contador)
IN3 OR IN4	87	1
Salida1	248	1
- Buttons: 'AGREGAR', 'REINICIAR'.

Comunicacion Section:

- Buttons: 'CONECTAR', 'Actualizar Conexion'.
- Section: 'RECEPCION DE DATOS' with 'Cargando...' status and 'fin de la Carga' message.
- Table:

87 1 0 0
248 1 0 0
- Buttons: 'ENVIAR DATO', 'Reiniciar tabla', 'SALIR'.

Other Sections:

- 'Visor de funciones' with 'Ver' button and 'Ingreso de Operacion (Contador)' field.
- 'Funcion Obtenida' field.
- 'Ingreso de Clave' with text input and 'ACEPTAR' button.

Figura 4.3 Programación función or

Con la función OR de acuerdo a la tabla 4.2, se activa la salida cuando una o las dos entradas se encuentran activas. Caso contrario, si las dos entradas se encuentran desactivadas, la salida no se activará.

Tabla 4.2 Tabla lógica función or

Entrada 3	Entrada 4	Salida 1
0	0	0
0	1	1
1	0	1
1	1	1

En la figura 4.4 se comprueba el resultado de la función or en el PLC.

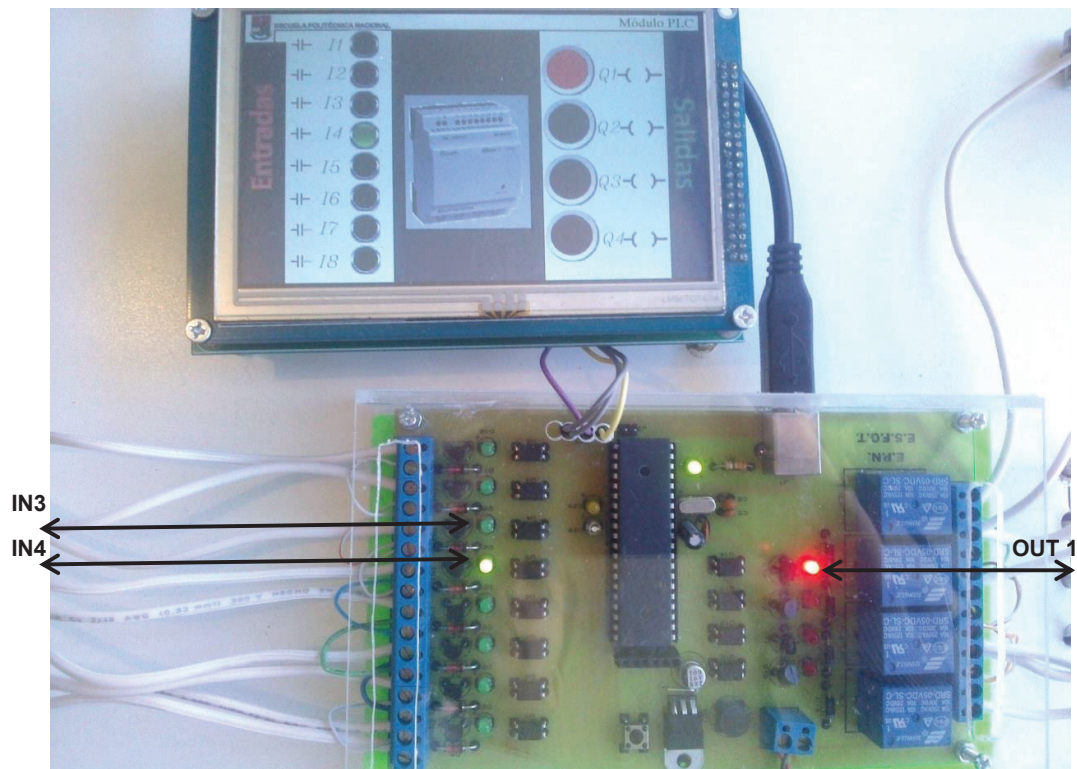


Figura 4.4 Resultado función or

4.2.1.3 Función not

En la figura 4.5 se muestra el ingreso de los datos para la prueba con la función not.

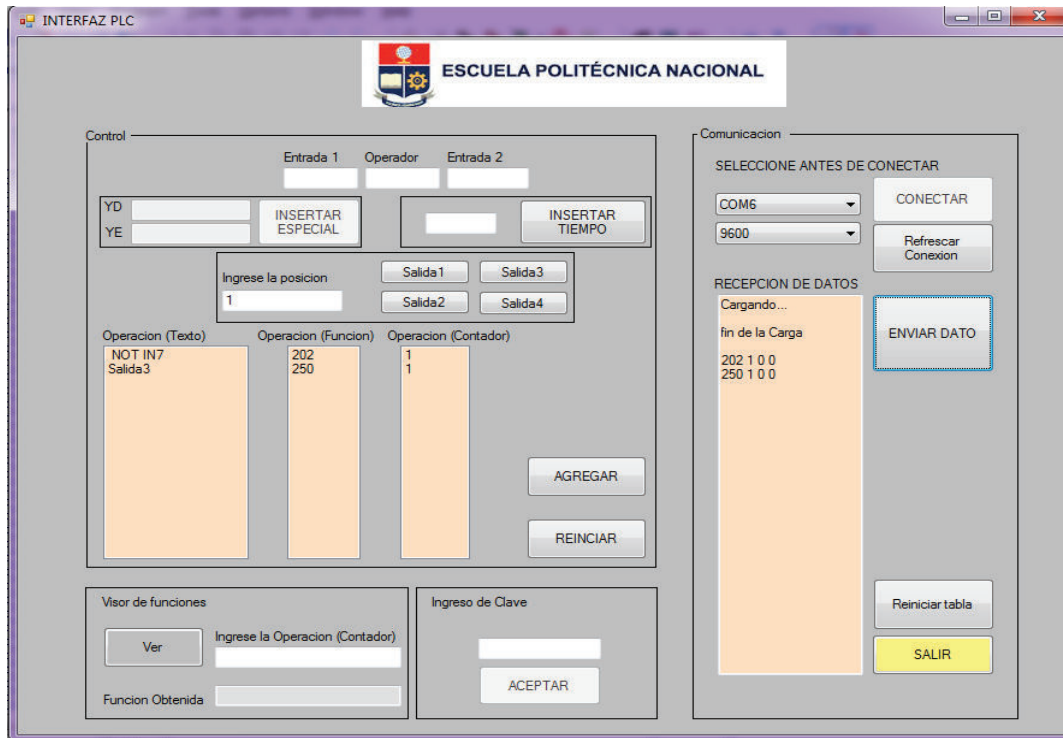


Figura 4.5 Programación función not

La función NOT como se puede observar en la tabla 4.3, dependiendo del estado en que se encuentre la entrada, se invierte su valor a la salida, activándola o desactivándola.

Tabla 4.3 Tabla lógica función not

Entrada 7	Salida 3
0	1
1	0

En la figura 4.6 se comprueba el resultado de la función not en el PLC.

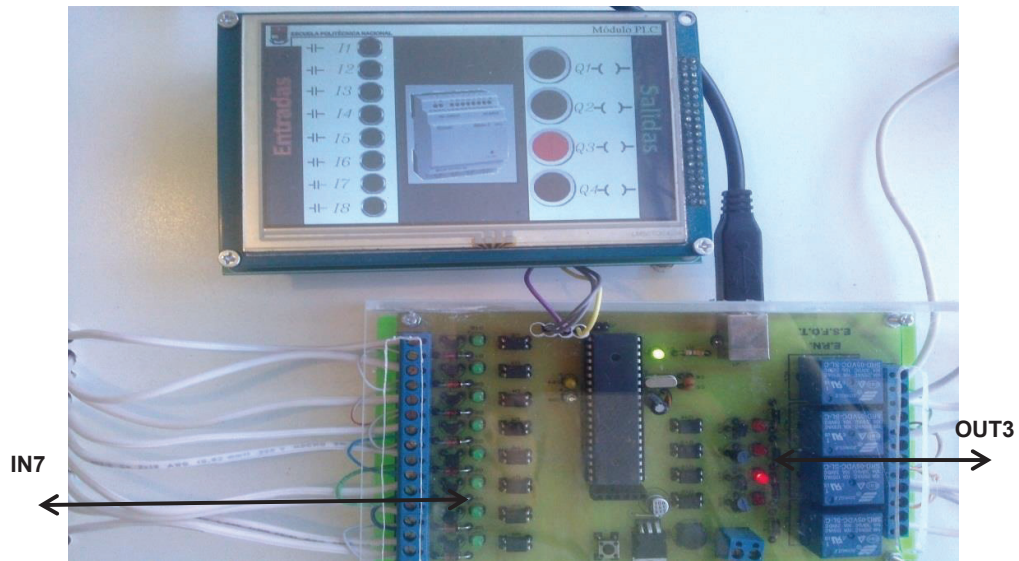


Figura 4.6 Resultado de la función not

4.2.1.3 Función xor

En la figura 4.7 se muestra el ingreso de los datos para la prueba con la función XOR.

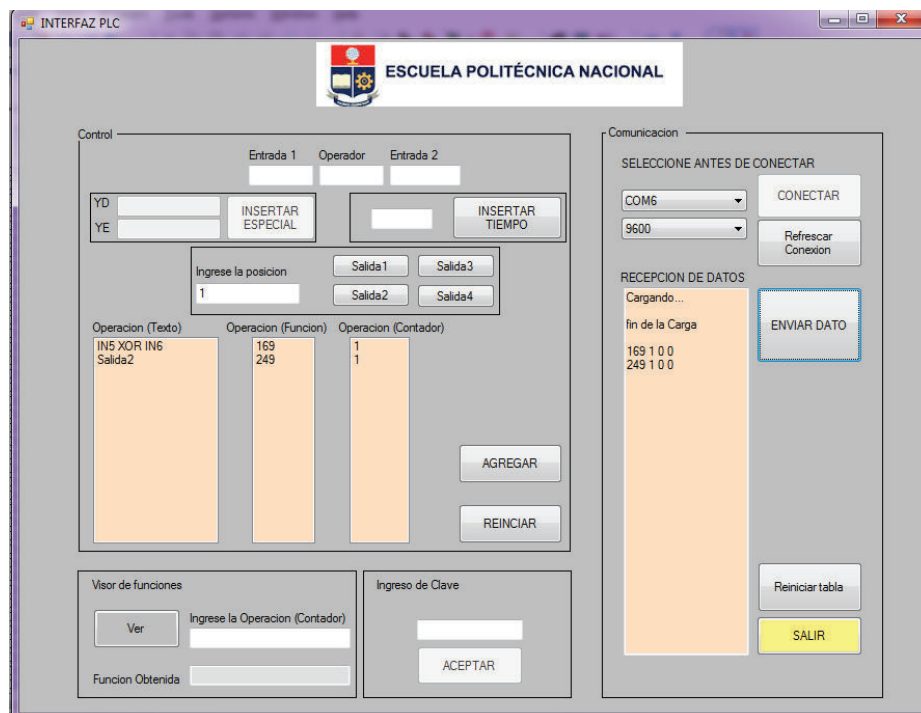


Figura 4.7 Programación función xor

En esta función según la tabla 4.4, cuando las entradas tienen diferentes estados, la entrada se activa.

Tabla 4.4 Tabla lógica función xor

Entrada 5	Entrada 6	Salida 2
0	0	0
0	1	1
1	0	1
1	1	0

En la figura 4.8 se comprueba el resultado de la Función XOR en el PLC.

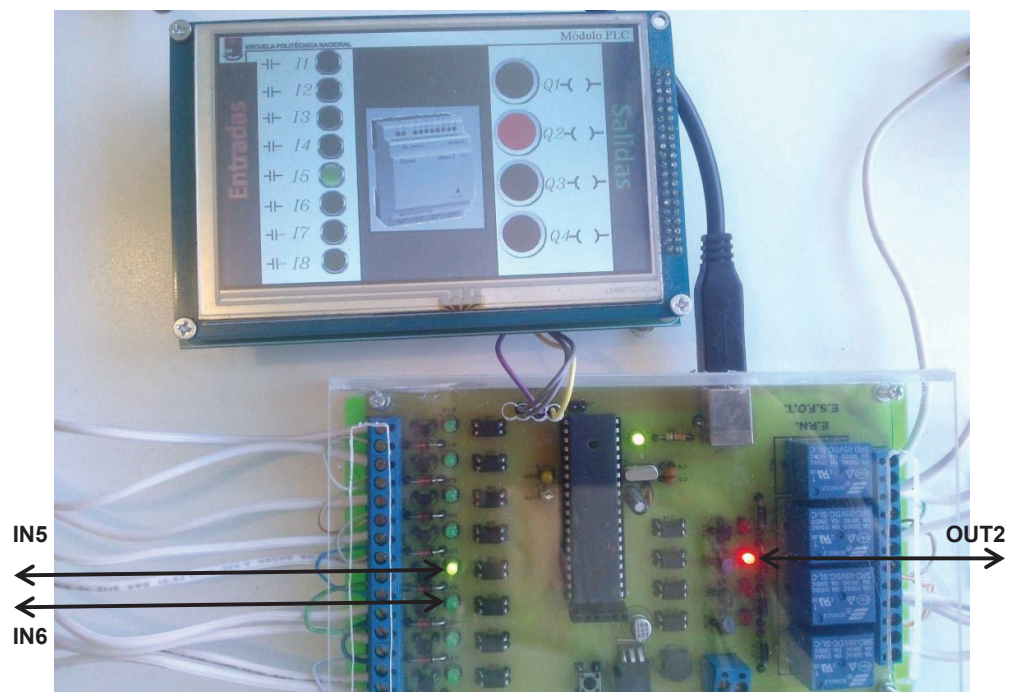


Figura 4.8 Resultado función xor

4.2.2 PRUEBA GENERAL DEL SISTEMA

En esta prueba se comprobará el funcionamiento de todas las entradas y salidas del PLC, el cual estará encargado de controlar un sistema de semáforos para una intersección de 2 vías, para lo cual se utilizará los siguientes elementos:

- PLC implementado.
- Switches para controlar las entradas.
- Fuente de alimentación de 12VDC para la señal de las entradas.
- 6 luces LED de 110 VAC de diferentes colores que simularán la luz de los semáforos.
- Braker de protección de 10A para el circuito de fuerza.
- Enchufe y cable N° 12AWG para el circuito de fuerza.
- Tabla MDF para montar todos los elementos.
- Cortapicos.
- Computador con el programa del PLC instalado.

4.2.2.1 Armado

La fuente de 12VDC alimentará a las ocho entradas digitales, las cuales se podrán activar con cada uno de los switches, además se utilizará las salidas normalmente cerradas de los relés para la conexión a 110V de los focos LED, el braker se pondrá en el cable que lleva la fase hacia el circuito de fuerza. El primer par de focos (verde y rojo) se conectarán a la salida 1, el segundo par de focos (verde y rojo) a la salida 3 del PLC mientras que unos de los focos amarillos se conectarán a la salida 2 y el otro a la salida 4.

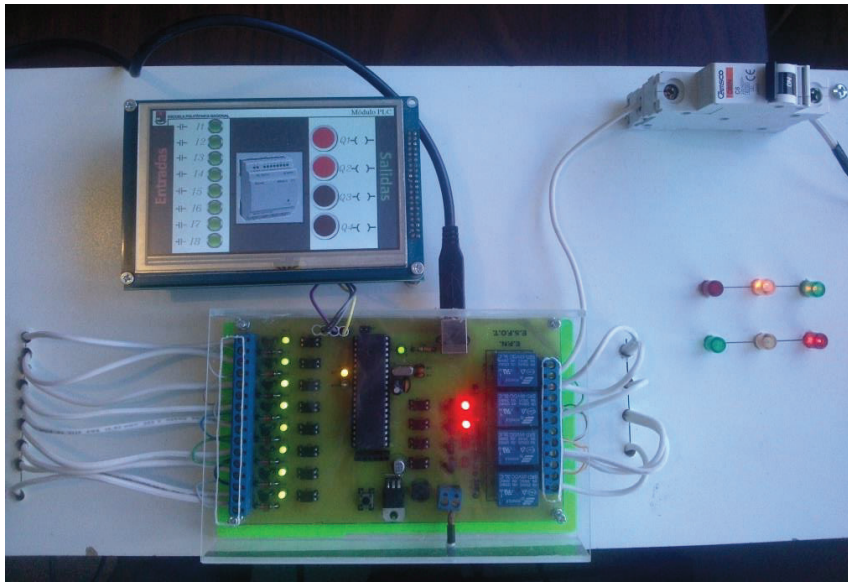


Figura 4.9 Circuito armado semáforo

4.2.2.2 Programación

Una vez armado el circuito se procede a ingresar las instrucciones en el software que controla al PLC como se muestra en la siguiente figura.

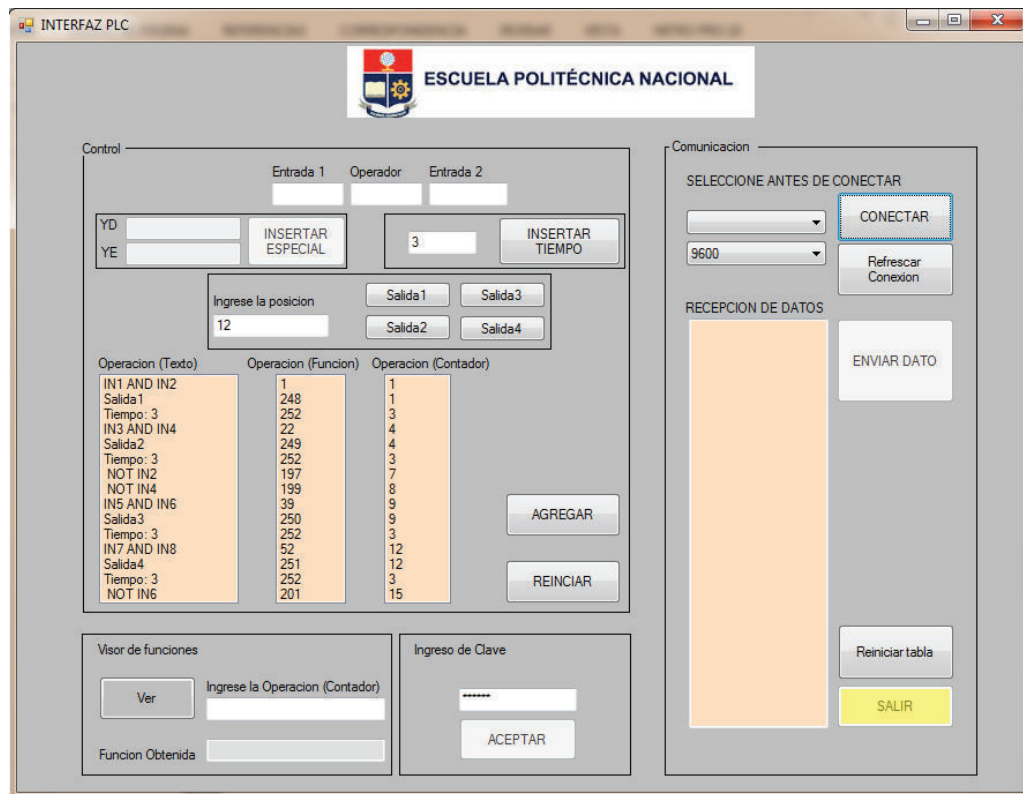


Figura 4.10 Programación

En la figura 4.10 podemos observar la programación para el control del semáforo mediante un listado de instrucciones que debe seguir el PLC de la siguiente manera:

- IN1 e IN2 deben estar activadas y estas activarán la salida 1 durante 3 segundos.
- IN3 e IN4 se activarán unos 3 segundos después de la salida 1 y estas activarán la salida 2 durante 3 segundos.
- Para desactivar las salidas 1 y 2 se niega las entradas IN2 e IN4.
- Una vez desactivadas las salidas 1 y 2 IN5 e IN6 activaran la salida 3 durante 3 segundos.
- Después de 3 segundos IN7 and IN8 activaran la salida 4 durante 3 segundos.
- Se desactivará las salidas 3 y 4 negando las entradas IN6 e IN8 y el ciclo se repetirá.

4.3 ANÁLISIS TÉCNICO Y ECONÓMICO

De las pruebas realizadas y mencionadas anteriormente se demuestra que el PLC tiene un correcto funcionamiento en la parte de software como de hardware. Esto permite que el PLC sea utilizado por personas que empiezan a adquirir conocimientos de automatización, como para personas experimentadas en el área.

Con lo referente a costos, los mismos se detallan en la tabla 4.5.

Tabla 4.5 Costos PLC

ITEM	CANT	DESCRIPCIÓN	V.UNITARIO	V.TOTAL
1	1	PIC 18f4550	7,00	7,00
2	12	Opto acopladores PC817	1,28	15,36
3	8	Diodo Zener	0,52	4,16
4	50	Resistencias	0,05	2,5
5	18	Borneras	0,37	6,66
6	12	Transistores 2N3904	0,8	9,6
7	15	Led's	0,05	0,75
8	5	Capacitores	0,13	0,65
9	1	TFT Arduino	42,00	42,00
10	1	Cristal 20 MHz	1,4	1,4
11	1	Conector Mini USB	1,2	1,2
12	4	Relés	3,24	12,92
13	4	Diodos 1N4007	0,13	0,52
14	1	Cable USB	2,00	2,00
TOTAL				106,72

Equipos disponibles en el mercado:

- ¡LOGO! 24C, módulo lógico, display AL/E/S: 24V/24V/24V transistor, 8ED (4EA) / 4SD, memoria 200 bloques, reloj tiempo real, ampliable modularmente. Precio: 123,28\$ más costos de envío, no incluye cable de comunicación ni software (Electrónicos).
- Logo 8 24vdc Siemens Ethernet. Precio: 145\$ no incluye cable de comunicación ni software. (ELECTRONICA_2012)

A través del análisis económico, en comparación con los PLC existentes en el mercado, la construcción del mismo es más económica que su compra.

4.4 ALCANCES Y LIMITACIONES

4.4.1 ALCANCES

- El PLC construido realiza operaciones lógicas como: and, or, not, xor y el temporizador on delay.
- Permite la comunicación y transferencia de datos computador/PLC mediante el interfaz USB.

- Se garantiza la seguridad de las entradas y salidas, al estar aisladas por medio de opto acopladores.
- Se observa el funcionamiento del PLC a través de la pantalla TFT. Para conocer si está ejecutando el programa de manera correcta.
- Para evitar la modificación del programa, se adicionó el ingreso de una clave, la cual habilita el envío de datos.

4.4.2 LIMITACIONES

- Las operaciones lógicas se realizan con un máximo de dos entradas.
- La interfaz USB únicamente permite el envío de datos para la programación del PLC, ya que no permite verificar la existencia de datos anteriormente programados.
- En comparación con otros lenguajes de programación como el ladder, el lenguaje booleano requiere mayor conocimiento técnico.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Se logró construir un Controlador Lógico Programable con interface USB a base de Micro Controladores de fácil programación y de bajo costo. Como se puede observar en el análisis económico.
- El PLC es capaz de almacenar una serie de funciones para llevar a cabo una operación, sin que estas se borren de la memoria interna al desconectarlo de la fuente de energía. Los datos de la programación del PLC se encuentran almacenados en la memoria EEPROM ya que, por ser una memoria no volátil, garantiza el almacenamiento de los datos en caso de pérdida de energía.
- Actualmente se tiene diferentes softwares de desarrollo para aplicaciones Windows, uno de estos es Visual Studio 2012 que a través de sus controladores y código de programación permitieron que el interfaz de usuario sea óptimo y fácil de utilizar.
- Para construir un Controlador Lógico Programable básico no se necesitan de grandes inversiones ni alta tecnología, pues con recursos limitados se ha logrado desarrollar el presente proyecto.

RECOMENDACIONES

- Para la elección de los diferentes elementos del PLC es necesario realizar un análisis previo de las características y limitaciones de cada uno de ellos.
- Estudiar los manuales y data sheets del PIC y de los dispositivos, para una correcta configuración de la memoria del PIC como para el diseño de la placa del PLC.
- Se debe utilizar un software que contenga las herramientas necesarias para la adecuada programación del PIC.
- Tener cuidado con los cables que unen la pantalla TFT a la placa del PLC, porque en el caso de desconectarse, la pantalla puede sufrir daños. Se debe reiniciar el equipo, desconectándolo de la fuente.

REFERENCIA BIBLIOGRÁFICAS

- adrformación. (s.f.). *Curso de Autómatas Programables*. Recuperado el 15 de Febrero de 2015, de <http://www.adrformacion.com/cursos/automatas1/leccion1/tutorial1.html>
- AliExpress. (s.f.). Recuperado el 16 de Febrero de 2015, de <http://es.aliexpress.com/item/Simatic-LOGO-12-24RC-6ED1052-1MD00-0BA6-PLC-Programmable-Logic-Module-Controller-100-New-In-Box/883125674.html?spm=2114.43010208.4.1.OVjfoU>
- Angelica. (20 de Agosto de 2010). *Descripción del PIC 16F877*. Recuperado el 10 de Marzo de 2015, de <http://phoenixdark87.blogspot.com/2010/08/descripcion-del-pic-16f877.html>
- Arduino, A. (s.f.). *Aprendiendo Arduino*. Recuperado el 10 de Abril de 2015, de <https://aprendiendoarduino.wordpress.com/2014/11/23/tema-1-plataforma-hardware-arduino/>
- Automatización, H. (s.f.). *Serie PLC DirectLOGIC 205*. Recuperado el 16 de Febrero de 2015, de <http://hi-techautomatizacion.com/serie-plc-directlogic-205/>
- Ayala, A. (22 de Noviembre de 2012). *Controladores Programables*. Recuperado el 16 de Febrero de 2015, de <http://1607085.blogspot.com/2012/11/26-controladores-programables.html>
- Camacho, R. (9 de Abril de 2012). *Computo Integrado*. Recuperado el 25 de Febrero de 2015, de <http://rcmcomputointegrado.blogspot.com/2012/04/arquitectura-von-neumann.html>
- Castro, C. (26 de Enero de 2012). *IncoElectrónica*. Recuperado el 20 de Abril de 2015, de <https://sites.google.com/site/incoelectronicasas/home/conversor-usb-a-rs232>
- Castro, J. M. (2007). *Dispositivo de estado sólido en el espectro infrarrojo*. 57. Veracruz, México.
- Castro, J. (s.f.). *Tecnología de hoy y el futuro*. Recuperado el 20 de Marzo de 2015, de <https://sites.google.com/site/profejulioc/tutorias/informatica/el-computador>
- CCM. (s.f.). *Conectores USB*. Recuperado el 1 de Abril de 2015, de <http://es.ccm.net/contents/189-conector-usb-usb-2-0>

- COMPARE, P. (s.f.). *PLC COMPARE*. Recuperado el 16 de Febrero de 2015, de <http://plccompare.com/triangle-research-international-nano-10/>
- Costales, A. (2012). *Apuntes Control con Microcontroladores*. Quito.
- Culturación. (s.f.). *Cuál es la diferencia entre un micro USB y un mini USB*. Recuperado el 1 de Abril de 2015, de <http://culturacion.com/cual-es-la-diferencia-entre-un-micro-usb-y-un-mini-usb/>
- ELECTRONICA_2012. (s.f.). *Mercado Libre*. Recuperado el 27 de Febrero de 2016, de http://articulo.mercadolibre.com.ec/MEC-408466675-rele-inteligente-logo-8-24vdc-siemens-ethernet-_JM
- Electrónicos, A. E. (s.f.). *Aparatos Eléctricos y Electrónicos*. Recuperado el 27 de Febrero de 2016, de http://www.solucionesyservicios.biz/epages/64466233.sf/es_ES/?ObjectPath=/Shops/64466233/Products/6ED1052-1CC01-0BA6
- electronics, G. (s.f.). *Songle Relay. 1-2*. Recuperado el 12 de Abril de 2015
- Fonquernie, A. (17 de Octubre de 2012). *Características del U.S.B*. Recuperado el 20 de Marzo de 2015, de <https://fonquernie.wordpress.com/2012/10/17/caracteristicas-del-u-s-b/>
- Frino, L. (s.f.). *Luis Frino Electrónica*. Recuperado el 5 de Marzo de 2015, de <http://www.frino.com.ar/micros.htm>
- Inc, M. T. (2009). *DataSheet PIC 18F4550*.
- Maser, G. (s.f.). *El PLC*. Recuperado el 21 de Febrero de 2015, de http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/PLC/plc.htm
- MIKROELEKTRONIKA. (s.f.). *CARACTERÍSTICAS BÁSICAS DEL PIC16F887*. Recuperado el 5 de Marzo de 2016, de <http://learn.mikroe.com/ebooks/microcontroladorespic/chapter/caracteristicas-basicas-del-pic16f887/>
- Molina, J. (20 de Diciembre de 2005). *Lenguaje de Programación de PLC'S*. Recuperado el 20 de Febrero de 2015, de bibdigital.epn.edu.ec/.../LENGUAJES%20DE%20PROGRAMACION%20...
- Molinari, I. N. (2012). *Controladores lógicos Programables*. Recuperado el 5 de Marzo de 2015, de http://www.edudevices.com.ar/download/articulos/PLC/CURSO_PLC_02.pdf

- Moreno, R. P. (2004). *Ingeniería de la automatización industrial*. Ra-Ma.
- Pérez, E. M. (2007). *Microcontroladores PIC 3era Edición*. Marcombo.
- Querevalu, O. (17 de Septiembre de 2013). *SIEMENS - SENATI*. Recuperado el 15 de Febrero de 2015, de <http://logosiemensdzpt.blogspot.com/2013/09/siemens-senati.html>
- Quiroz, J. H. (Marzo de 2004). *ElectroIndustria*. Recuperado el 20 de Febrero de 2015, de <http://www.emb.cl/electroindustria/articulo.mvc?xid=81>
- S, P. A. (2004). INTRODUCCIÓN AL MICROCONTROLADOR.
- Siemens. (s.f.). *Manual LOGO*. Recuperado el 20 de Febrero de 2015, de https://cache.industry.siemens.com/dl/files/761/19625761/att_75984/v1/logo_s_09_99.pdf
- SuFiiAn. (26 de Noviembre de 2010). *SuFiiAn TaNjAWii*. Recuperado el 1 de Abril de 2015, de <http://sufiantanjawiinfokursaal.blogspot.com/2010/11/conexiones-y-puertos.html>
- Unicrom, E. (s.f.). *Historia del PLC, Modicon, Modbus*. Recuperado el 9 de Febrero de 2015, de <http://unicrom.com/historia-del-plc-modicon-modbus/>
- WordPress.com. (3 de Octubre de 2011). *Introducción a los Microprocesadores*. Recuperado el 25 de Febrero de 2015, de <https://introduccionalosmicroprocesadores.wordpress.com/2011/10/03/diferencia-entre-microprocesador-y-microcontrolador/>

ANEXOS

ANEXO 1:
SOFTWARE DE PIC 18F4550

```

// Software para PIC 18F4550 desarrollado en PicBasicPro//
//Lenguaje de programación Basic//

$regfile = "M328PDEF.DAT"

$crystal = 8000000

$hwstack = 40

$swstack = 16

$framesize = 32

$baud = 9600

'Variable dimensioning

Dim Output_ As Bit , Input_1 As Bit , Input_2 As Bit , Temporary As Bit , Jump As Bit

Dim Working As Bit , Ctrlbit As Bit , Edge As Bit , Tmpmem As Byte , Interval As Byte

Dim K As Byte , Pc_input As Byte , Countr As Integer , Ir As Bit

Dim M0 As Bit : Dim M1 As Bit : Dim M2 As Bit : Dim M3 As Bit : Dim M4 As Bit

Dim M5 As Bit : Dim M6 As Bit : Dim M7 As Bit : Dim M8 As Bit : Dim M9 As Bit

O1 Alias Porta.0 : O2 Alias Porta.1 : O3 Alias Porta.2 : O4 Alias Porta.3

I1 Alias Pinb.0 : I2 Alias Pinb.1 : I3 Alias Pinb.2 : I4 Alias Pinb.3

I5 Alias Pinb.4 : I6 Alias Pinb.5 : I7 Alias Pinb.6 : I8 Alias Pinb.7

Ddrb.0 = 1 : Ddrb.1 = 1 : Ddrb.2 = 1 : Ddrb.3 = 1

Ddrb.4 = 0 : Ddrb.5 = 0 : Ddrb.6 = 0 : Ddrb.7 = 0

Portb.4 = 1 : Portb.5 = 1 : Portb.6 = 1 : Portb.7 = 1:

Dim Matriz(800) As Byte

'Initialization of uTimer

Config Timer0 = Timer , Prescale = 8

On Timer0 Timer0_interapt

Load Timer0 = 250

```



```

Enable Interrupts

Enable Timer0

Ddrb.4 = 1

Set Portb.4          'Turn led "RECEIVED DATA" OFF

Ddrb.5 = 1

Reset Portb.5       'Led "READY FOR RECEIVING" starts to blink

K = 2               'RAM Memory address from where starts writing PLC program

Do                 'Routine for loading RAM memory

    Pc_input = Waitkey()          'Wait for data

    Matriz(k) = Pc_input          'generating indirect address

    Incr K

Loop Until Pc_input = 99        'Code for "End Of Transfer"

Reset Portb.4          'Turn led "RECEIVED DATA" ON

Set Portb.5          'Led "READY FOR RECEIVING" stops to blink

Tmpmem = 0           '{

Countr = 0           '{

Jump = 0             '{Initialization of used variables

Edge = 0            '{

Ctrlbit = 0         '{

Prgstart:           'Inception

K = 1               'address-1 because of increment in next subroutine

Gosub Readram       'Go and read RAM memory

'Program

Nastavi:

'Detection of logic function and timer

Select Case Tmpmem

```

```
Case 10 : Gosub Inputs          'Take logic conditions of input1 and input2
    Output_ = Input_1 And Input_2    'AND
    Gosub Outp                    'Set output of current logic gate
    Goto Checkend

Case 11 : Gosub Inputs          'Take logic conditions of input1 and input2
    Output_ = Input_1 Or Input_2    'OR
    Gosub Outp                    'Set output of current logic gate
    Goto Checkend

Case 12 : Gosub Inputs          'Take logic conditions of input1 and input2
    Output_ = Input_1
    Output_ = Not Output_          'NOT
    Gosub Outp                    'Set output of current logic gate
    Goto Checkend

Case 13 : Gosub Inputs          'Take logic conditions of input1 and input2
    Output_ = Input_1 And Input_2
    Output_ = Not Output_          'NAND
    Gosub Outp                    'Set output of current logic gate
    Goto Checkend

Case 14 : Gosub Inputs          'Take logic conditions of input1 and input2
    Output_ = Input_1 Or Input_2
    Output_ = Not Output_          'NOR
    Gosub Outp                    'Set output of current logic gate
    Goto Checkend

Case 15 : Gosub Inputs          'Take logic conditions of input1 and input2
    Output_ = Input_1 Xor Input_2    'XOR
    Gosub Outp                    'Set output of current logic gate
    Goto Checkend
```

```

Case 24 : Gosub Tajmer          'TIMER
    Gosub Outp                  'Set output of Timer
    Goto Checkend

End Select

'Input detection

Inputs:

Gosub Readram                  '/
Gosub Inpt                     '| Find first input of current log.gate
Input_1 = Temporary           '\
Gosub Readram                  '/
Gosub Inpt                     '| Find second input of current log.gate
Input_2 = Temporary           '\

Return                          'Return from subroutine Inputs

'Finding if input in current log.gate is phisical input,phisical output or memory marker

Inpt:

Select Case Tmpmem

    Case 16 : Temporary = I1

    Case 17 : Temporary = I2

    Case 18 : Temporary = I3

    Case 19 : Temporary = I4

    Case 20 : Temporary = O1

        Temporary = Not Temporary

    Case 21 : Temporary = O2

        Temporary = Not Temporary

    Case 22 : Temporary = O3

        Temporary = Not Temporary

```

Case 23 : Temporary = O4

 Temporary = Not Temporary

Case 30 : Temporary = M0

Case 31 : Temporary = M1

Case 32 : Temporary = M2

Case 33 : Temporary = M3

Case 34 : Temporary = M4

Case 35 : Temporary = M5

Case 36 : Temporary = M6

Case 37 : Temporary = M7

Case 38 : Temporary = M8

Case 39 : Temporary = M9

End Select

Return

'Finding if output from current log.gate is physical output or memory marker

Outp:

Gosub Readram

Select Case Tmpmem

Case 20 : Output_ = Not Output_

 O1 = Output_

Case 21 : Output_ = Not Output_

 O2 = Output_

Case 22 : Output_ = Not Output_

 O3 = Output_

Case 23 : Output_ = Not Output_

 O4 = Output_

Case 30 : M0 = Output_

```

Case 31 : M1 = Output_
Case 32 : M2 = Output_
Case 33 : M3 = Output_
Case 34 : M4 = Output_
Case 35 : M5 = Output_
Case 36 : M6 = Output_
Case 37 : M7 = Output_
Case 38 : M8 = Output_
Case 39 : M9 = Output_

End Select

Return

'Timer routine

Tajmer:

Gosub Readram          '/'
Gosub Inpt             '| Find input of Timer
Input_1 = Temporary   '\
If Jump = 1 Then Goto L1      'Control bit for bypassing Timer input scan when
Timer Workings

Edge = Input_1

If Edge = 0 Then Set Jump    'If input Timer input is low SET control bit

L1:

If Input_1 = 1 Then          'These are 3 conditions which have to be accomplished
to start uTimer

If Edge = 0 Then

If Working = 0 Then

Start Timer0

Set Working

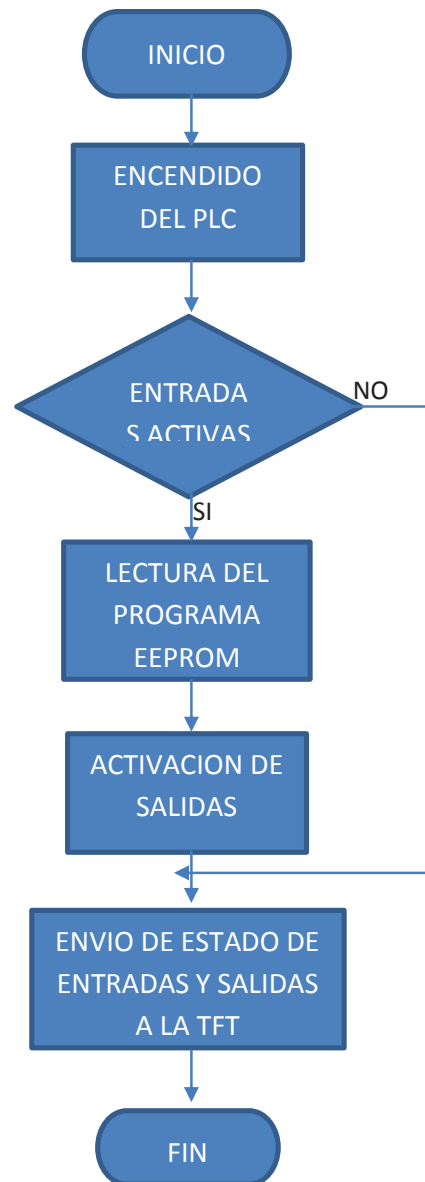
End If

```

```
Goto L2
End If
End If
Stop Timer0
Countr = 0           'Reset elapsed time counter
Working = 0         'Reset Timer condition bit
Output_ = 0         'Reset output of Timer
Ctrlbit = 0
L2:
If Ctrlbit = 1 Then
    K = K + 1
Else
    Gosub Readram
    Interval = Tmpmem
    Ctrlbit = 1
End If
If Interval = 0 Then
    Stop Timer0
    Countr = 0
    Reset Output_
    Jump = 0
    Reset Working
    Ctrlbit = 0
Else
    If Working = 1 Then
        Set Output_
    Else
```

```
    Reset Output_  
End If  
End If  
Return  
  
'Checking if it is end of PLC program and if it is, go again from start  
Checkend:  
Gosub Readram  
If Tmpmem = 99 Then  
    Goto Prgstart  
Else  
    Goto Nastavi  
End If  
  
'Interrupt routine for Timer interval  
Timer0_interapt:  
    Countr = Countr + 1           'Elapsed time counter  
    If Countr = 4000 Then        '4000x250=1s  
        Interval = Interval - 1 'Previously set interval by user  
        Countr = 0              'Reset elapsed time counter  
    End If  
Return  
  
'Routine for reading RAM memory in Execution mode  
Readram:  
    Incr K                       'Increment address value  
    Tmpmem = Matriz(k)  
Return  
End
```

ANEXO 2:
DIAGRAMA DE FLUJO PIC 18F4550



ANEXO 3:
SOFTWARE INTERFAZ USUARIO

```

//Software para Interfaz de usuario//
//Desarrollado en Visual Studio 2012//

Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel

Public Class InterfazPLC

    'VARIABLES GLOBALES
    Dim strE1 As String
    Dim strOperador As String
    Dim strE2 As String
    Dim strTODO As String

    Dim intE1 As Integer
    Dim intOp As Integer
    Dim intE2 As Integer

    Dim contador As Integer
    Dim numero(1000) As Integer
    Dim palabra(1000) As String
    Dim counter(1000) As Integer

    Dim StrBufferOut(1000) As String
    Dim StrBufferIn(1000) As String

    Dim entrada1(1000) As Integer
    Dim entrada2(1000) As Integer

    Dim strEnviar As String

    Dim myPort As Array

    Dim strClave As String

    Delegate Sub SetTextCallback(ByVal [text] As String)

    'PRECARGA INICIAL
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        strE1 = ""
        strOperador = ""
        strE2 = ""
        strTODO = ""
        contador = 1

        txtYD.Enabled = False
        txtYE.Enabled = False
        btnY.Enabled = False

        myPort = IO.Ports.SerialPort.GetPortNames()
        puertos.Items.AddRange(myPort)
        btnEnviarDato.Enabled = False
        btnSalir.Enabled = False
        btnAceptar.Enabled = False

        txtClave.PasswordChar = "*"
    End Sub

```

```

'BOTON AGREGAR
Private Sub btnAgregar_Click(sender As Object, e As EventArgs) Handles
btnAgregar.Click
    Dim cambio1 As Integer
    Dim cambio2 As Integer
    Dim strcambio1 As String
    Dim strcambio2 As String
    Dim aux As Integer
    aux = 0

    strE1 = txtE1.Text.ToUpper()
    strE2 = txtE2.Text.ToUpper()
    strOperador = txtOp.Text.ToUpper()

    If strE1 = "IN1" Then
        intE1 = 1
    ElseIf strE1 = "IN2" Then
        intE1 = 2
    ElseIf strE1 = "IN3" Then
        intE1 = 3
    ElseIf strE1 = "IN4" Then
        intE1 = 4
    ElseIf strE1 = "IN5" Then
        intE1 = 5
    ElseIf strE1 = "IN6" Then
        intE1 = 6
    ElseIf strE1 = "IN7" Then
        intE1 = 7
    ElseIf strE1 = "IN8" Then
        intE1 = 8
    ElseIf strE1 = "OUT1" Then
        intE1 = 9
    ElseIf strE1 = "OUT2" Then
        intE1 = 10
    ElseIf strE1 = "OUT3" Then
        intE1 = 11
    ElseIf strE1 = "OUT4" Then
        intE1 = 12
    ElseIf strE1 = "YD" Then
        intE1 = 13
    Else
        If strOperador = "NOT" Then

            Else
                MessageBox.Show("No existe dicha entrada/salida en la entrada 1")
                intE1 = 0
                intE2 = 0
                intOp = 0
                strE1 = ""
                strE2 = ""
                strOperador = ""
                txtE1.Text = ""
                txtE2.Text = ""
                txtOp.Text = ""
                strTODO = ""
                Exit Sub
            End If
        End If
    End If

```

```

If strE2 = "IN1" Then
    intE2 = 1
ElseIf strE2 = "IN2" Then
    intE2 = 2
ElseIf strE2 = "IN3" Then
    intE2 = 3
ElseIf strE2 = "IN4" Then
    intE2 = 4
ElseIf strE2 = "IN5" Then
    intE2 = 5
ElseIf strE2 = "IN6" Then
    intE2 = 6
ElseIf strE2 = "IN7" Then
    intE2 = 7
ElseIf strE2 = "IN8" Then
    intE2 = 8
ElseIf strE2 = "OUT1" Then
    intE2 = 9
ElseIf strE2 = "OUT2" Then
    intE2 = 10
ElseIf strE2 = "OUT3" Then
    intE2 = 11
ElseIf strE2 = "OUT4" Then
    intE2 = 12
ElseIf strE2 = "YE" Then
    intE2 = 14
Else
    MessageBox.Show("No existe dicha entrada/salida en la entrada 1")
    intE1 = 0
    intE2 = 0
    intOp = 0
    strE1 = ""
    strE2 = ""
    strOperador = ""
    txtE1.Text = ""
    txtE2.Text = ""
    txtOp.Text = ""
    strTODO = ""
    Exit Sub
End If

If strOperador = "AND" Then
    intOp = 1
ElseIf strOperador = "OR" Then
    intOp = 2
ElseIf strOperador = "XOR" Then
    intOp = 3
ElseIf strOperador = "NOT" Then
    intOp = 4
    txtE1.Text = ""
Else
    MessageBox.Show("No Existe dicho Operador")
    strE1 = ""
    strE2 = ""
    strOperador = ""
    txtE1.Text = ""
    txtE2.Text = ""
    txtOp.Text = ""
    strTODO = ""
    intOp = 0
    intE1 = 0

```

```

        intE2 = 0
        Exit Sub
    End If

    'error de ingresar la misma funcion
    If intE1 = intE2 Then
        strE1 = ""
        strE2 = ""
        strOperador = ""
        txtE1.Text = ""
        txtE2.Text = ""
        txtOp.Text = ""
        strTODO = ""
        MsgBox.Show("se ingresaron variables iguales no disponibles")
        intE1 = 0
        intE2 = 0
        intOp = 0
        Exit Sub

        MsgBox.Show("Numero de Operaciones límite, No se puede ingresar mas
funciones. Pulse reiniciar para volver a ingresar")
        btnAgregar.Enabled = False
        btnTimer.Enabled = False
        intE1 = 0
        intE2 = 0
        intOp = 0
        Exit Sub
    End If

    '*****
    ' *****funciones*****
    ' *****

    '***** NOT *****

    If intOp = 4 Then
        txtE1.Text = ""
        strE1 = ""
        If intE2 = 1 Then
            numero(contador) = 196
            GoTo Operaciones1

        ElseIf intE2 = 2 Then
            numero(contador) = 197
            GoTo Operaciones1

        ElseIf intE2 = 3 Then
            numero(contador) = 198
            GoTo Operaciones1

        ElseIf intE2 = 4 Then
            numero(contador) = 199
            GoTo Operaciones1

        ElseIf intE2 = 5 Then
            numero(contador) = 200
            GoTo Operaciones1

        ElseIf intE2 = 6 Then
            numero(contador) = 201

```

```

        GoTo Operaciones1

    ElseIf intE2 = 7 Then
        numero(contador) = 202
        GoTo Operaciones1

    ElseIf intE2 = 8 Then
        numero(contador) = 203
        GoTo Operaciones1

    ElseIf intE2 = 9 Then
        numero(contador) = 204
        GoTo Operaciones1

    ElseIf intE2 = 10 Then
        numero(contador) = 205
        GoTo Operaciones1

    ElseIf intE2 = 11 Then
        numero(contador) = 206
        GoTo Operaciones1

    ElseIf intE2 = 12 Then
        numero(contador) = 207
        GoTo Operaciones1
    ElseIf intE2 = 14 Then
        btnAgregar.Enabled = False
        btnY.Enabled = True
        txtYD.Enabled = False
        txtYE.Enabled = True
        Exit Sub

    Else
        intE1 = 0
        intE2 = 0
        intOp = 0
        strE1 = ""
        strE2 = ""
        strOperador = ""
        txtE1.Text = ""
        txtE2.Text = ""
        txtOp.Text = ""
        strTODO = ""
        MessageBox.Show("orden de variables incorrectas o no existentes")
        Exit Sub
    End If
End If

If intOp = 1 Then

    If intE2 < intE1 Then
        If intE1 = 13 Then
            Else
                cambio1 = intE1
                cambio2 = intE2
                intE1 = cambio2
                intE2 = cambio1
                cambio1 = 0
                cambio2 = 0
                strcambio1 = strE1
            End If
        End If
    End If

```

```
        strcambio2 = strE2
        strE1 = strcambio2
        strE2 = strcambio1
        strcambio1 = ""
        strcambio2 = ""
    End If

End If
If intE1 = 1 And intE2 = 2 And intOp = 1 Then
    numero(contador) = 1
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 3 And intOp = 1 Then
    numero(contador) = 2
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 4 And intOp = 1 Then
    numero(contador) = 3
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 5 And intOp = 1 Then
    numero(contador) = 4
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 6 And intOp = 1 Then
    numero(contador) = 5
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 7 And intOp = 1 Then
    numero(contador) = 6
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 8 And intOp = 1 Then
    numero(contador) = 7
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 9 And intOp = 1 Then
    numero(contador) = 8
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 10 And intOp = 1 Then
    numero(contador) = 9
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 11 And intOp = 1 Then
    numero(contador) = 10
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 12 And intOp = 1 Then
    numero(contador) = 11
    GoTo Operaciones1

    outs
ElseIf intE1 = 9 And intE2 = 10 And intOp = 1 Then
    numero(contador) = 61
    GoTo Operaciones1

ElseIf intE1 = 9 And intE2 = 11 And intOp = 1 Then
    numero(contador) = 62
    GoTo Operaciones1
```



```

ElseIf intE1 = 9 And intE2 = 12 And intOp = 1 Then
    numero(contador) = 63
    GoTo Operaciones1

ElseIf intE1 = 10 And intE2 = 11 And intOp = 1 Then
    numero(contador) = 64
    GoTo Operaciones1

ElseIf intE1 = 11 And intE2 = 12 And intOp = 1 Then
    numero(contador) = 65
    GoTo Operaciones1

ElseIf intE1 = 13 And intE2 <> 14 Then

    btnAgregar.Enabled = False

    btnY.Enabled = True
    txtYD.Enabled = True
    txtYE.Enabled = False
    Exit Sub
    'condicion de 2 especiales
ElseIf intE1 = 13 And intE2 = 14 Then

    btnAgregar.Enabled = False

    btnY.Enabled = True
    txtYD.Enabled = True
    txtYE.Enabled = True
    Exit Sub
Else
    intE1 = 0
    intE2 = 0
    intOp = 0
    strE1 = ""
    strE2 = ""
    strOperador = ""
    txtE1.Text = ""
    txtE2.Text = ""
    txtOp.Text = ""
    strTODO = ""
    MessageBox.Show("orden de variables incorrectas o no existentes")
    Exit Sub
End If
End If

'*****OR*****

If intOp = 2 Then
    If intE2 < intE1 Then
        If intE1 = 13 Then
            Else
                cambio1 = intE1
                cambio2 = intE2
                intE1 = cambio2
                intE2 = cambio1
                cambio1 = 0
                cambio2 = 0
                strcambio1 = strE1
                strcambio2 = strE2

```

```
        strE1 = strcambio2
        strE2 = strcambio1
        strcambio1 = ""
        strcambio2 = ""
    End If
End If
If intE1 = 1 And intE2 = 2 Then
    numero(contador) = 66
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 3 Then
    numero(contador) = 67
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 4 Then
    numero(contador) = 68
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 5 Then
    numero(contador) = 69
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 6 Then
    numero(contador) = 70
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 7 Then
    numero(contador) = 71
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 8 Then
    numero(contador) = 72
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 9 Then
    numero(contador) = 73
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 10 Then
    numero(contador) = 74
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 11 Then
    numero(contador) = 75
    GoTo Operaciones1

ElseIf intE1 = 1 And intE2 = 12 Then
    numero(contador) = 76
    GoTo Operaciones1

'condicion de 2 especiales
ElseIf intE1 = 13 And intE2 = 14 Then

    btnAgregar.Enabled = False

    btnY.Enabled = True
    txtYD.Enabled = True
    txtYE.Enabled = True
    Exit Sub
Else
```

```

intE1 = 0
intE2 = 0
intOp = 0
strE1 = ""
strE2 = ""
strOperador = ""
txtE1.Text = ""
txtE2.Text = ""
txtOp.Text = ""
strTODO = ""
MessageBox.Show("orden de variables incorrectas o no existentes")
Exit Sub
End If

```

```
End If
```

```
'*****XOR*****
```

```
If intOp = 3 Then
```

```

If intE2 < intE1 Then
  If intE1 = 13 Then
  Else
    cambio1 = intE1
    cambio2 = intE2
    intE1 = cambio2
    intE2 = cambio1
    cambio1 = 0
    cambio2 = 0
    strcambio1 = strE1
    strcambio2 = strE2
    strE1 = strcambio2
    strE2 = strcambio1
    strcambio1 = ""
    strcambio2 = ""
  End If

```

```
End If
```

```

If intE1 = 1 And intE2 = 2 And intOp = 3 Then
  numero(contador) = 131
  GoTo Operaciones1

```

```

ElseIf intE1 = 1 And intE2 = 3 And intOp = 3 Then
  numero(contador) = 132
  GoTo Operaciones1

```

```

ElseIf intE1 = 1 And intE2 = 4 And intOp = 3 Then
  numero(contador) = 133
  GoTo Operaciones1

```

```

ElseIf intE1 = 1 And intE2 = 5 And intOp = 3 Then
  numero(contador) = 134
  GoTo Operaciones1

```

```

ElseIf intE1 = 1 And intE2 = 6 And intOp = 3 Then
  numero(contador) = 135
  GoTo Operaciones1

```

```

ElseIf intE1 = 1 And intE2 = 7 And intOp = 3 Then
  numero(contador) = 136

```

```

        GoTo Operaciones1

    ElseIf intE1 = 1 And intE2 = 8 And intOp = 3 Then
        numero(contador) = 137
        GoTo Operaciones1

    ElseIf intE1 = 1 And intE2 = 9 And intOp = 3 Then
        numero(contador) = 138
        GoTo Operaciones1

    ElseIf intE1 = 1 And intE2 = 10 And intOp = 3 Then
        numero(contador) = 139
        GoTo Operaciones1

    ElseIf intE1 = 1 And intE2 = 11 And intOp = 3 Then
        numero(contador) = 140
        GoTo Operaciones1

    ElseIf intE1 = 1 And intE2 = 12 And intOp = 3 Then
        numero(contador) = 141
        GoTo Operaciones1

    ' fin operaciones 1

ElseIf intE1 = 13 And intE2 <> 14 Then

    btnAgregar.Enabled = False

    btnY.Enabled = True
    txtYD.Enabled = True
    txtYE.Enabled = False
    Exit Sub
    'condicion de 2 especiales
ElseIf intE1 = 13 And intE2 = 14 Then

    btnAgregar.Enabled = False

    btnY.Enabled = True
    txtYD.Enabled = True
    txtYE.Enabled = True
    Exit Sub
Else
    intE1 = 0
    intE2 = 0
    intOp = 0
    strE1 = ""
    strE2 = ""
    strOperador = ""
    txtE1.Text = ""
    txtE2.Text = ""
    txtOp.Text = ""
    strTODO = ""
    MessageBox.Show("orden de variables incorrectas o no existentes")
    Exit Sub
End If
End If

Exit Sub

```

```

Operaciones1:
    strTODO = strE1 & " " & strOperador & " " & strE2
    palabra(contador) = strTODO
    lb1.Items.Add(strTODO)
    lb2.Items.Add(numero(contador))
    counter(contador) = contador
    lb3.Items.Add(counter(contador))
    entrada1(contador) = 0
    entrada2(contador) = 0
    contador = contador + 1
    intE1 = 0
    intE2 = 0
    intOp = 0
    strE1 = ""
    strE2 = ""
    strOperador = ""
    txtE1.Text = ""
    txtE2.Text = ""
    txtOp.Text = ""
    strTODO = ""
Exit Sub
End Sub

'BOTON REINICIAR *****

Private Sub btnReiniciar_Click(sender As Object, e As EventArgs) Handles
btnReiniciar.Click
    Dim H As Integer = 0

    strE1 = ""
    strE2 = ""
    strOperador = ""
    strTODO = ""
    txtVisor.Text = ""
    txtIngresoContador.Text = ""

    lb1.Items.Clear()
    lb2.Items.Clear()
    lb3.Items.Clear()

    txtE1.Text = ""
    txtE2.Text = ""
    txtOp.Text = ""

    For H = 0 To contador
        numero(H) = 0
        counter(H) = 0
        palabra(H) = ""
    Next

    btnAgregar.Enabled = True
    btnTimer.Enabled = True
    btnY.Enabled = False
    txtYD.Enabled = False
    txtYE.Enabled = False

    contador = 1

End Sub

```

```

Private Sub btnVer_Click(sender As Object, e As EventArgs) Handles btnVer.Click
    Dim I As Integer = counter(Convert.ToInt32(txtIngresoContador.Text))

    If counter(I) = 0 Then
        MessageBox.Show("Funcion aun no creada")
        txtVisor.Text = ""
        txtIngresoContador.Text = ""
    Else
        txtVisor.Text = palabra(I)
    End If
End Sub

'*****
'*****Yd y Ye*****
'*****

'condicion de solo 1 especial

Private Sub btnY_Click(sender As Object, e As EventArgs) Handles btnY.Click
    Dim numaux As Integer
    Dim numaux1 As Integer
    Dim N As Integer
    N = 0
    numaux = 0
    numaux1 = 0

    If intOp = 4 Then
        GoTo salto55
    End If

    Try
        strE1 = txtYD.Text.ToUpper()
        numaux = CInt(strE1)
    Catch ex As Exception
        MessageBox.Show(ex.Message)
        txtYD.Text = ""
        Exit Sub
    End Try

salto55:

    If intE2 = 14 Then
        Try
            strE2 = txtYE.Text.ToUpper()
            numaux1 = CInt(strE2)
        Catch ex As Exception
            MessageBox.Show(ex.Message)
            txtYE.Text = ""
            txtYD.Text = ""
            Exit Sub
        End Try

        'para la NOT y YE
        If intOp = 4 Then
            GoTo salto11
        End If

        'para saber si existe yd

```

```

        For N = 0 To contador
            If numaux = counter(N) Then
                GoTo salto11
            End If
        Next

        MessageBox.Show("No se crea todavia la funcion YD")
        Exit Sub
    End If
    Private Sub btnConectar_Click(sender As Object, e As EventArgs) Handles
btnConectar.Click
        Try
            SerialPort1.PortName = puertos.Text
            SerialPort1.BaudRate = cboBaudios.Text
            SerialPort1.Open()

            btnConectar.Enabled = False
            btnAceptar.Enabled = True
            btnSalir.Enabled = True
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try

    End Sub

    Private Sub btnEnviarDato_Click(sender As Object, e As EventArgs) Handles
btnEnviarDato.Click
        bufferIn.Clear()
        SerialPort1.Write("<") 'quitar el & vbcr para el programa de bascom
        btnAceptar.Enabled = False
        Dim O As Integer

        For O = 1 To contador - 1
            SerialPort1.Write(CStr(numero(O)) & vbCr) ' VALOR DE LA FUNCION
            SerialPort1.Write(CStr(counter(O)) & vbCr) ' valor del contador
            SerialPort1.Write(CStr(entrada1(O)) & vbCr) ' valor de B(x) o yd
            SerialPort1.Write(CStr(entrada2(O)) & vbCr) ' valor de C(x)
        Next
        'saltarEscritura:

        SerialPort1.Write(">" & vbCr & vbLf)
    End Sub

    Private Sub btnSalir_Click(sender As Object, e As EventArgs) Handles
btnSalir.Click
        SerialPort1.Close()
        btnConectar.Enabled = True
        btnEnviarDato.Enabled = False
        btnSalir.Enabled = False

    End Sub

    Private Sub SerialPort1_DataReceived(sender As Object, e As
SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
        RecievedText(SerialPort1.ReadExisting())
    End Sub

    Private Sub RecievedText(ByVal [text] As String)
        If Me.bufferIn.InvokeRequired Then
            Dim x As New SetTextCallback(AddressOf RecievedText)
            Me.Invoke(x, New Object() {(text)})
        End If
    End Sub

```

```
Else
    Me.bufferIn.Text &= [text]
End If
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    puertos.Items.Clear()
    SerialPort1.Close()
    myPort = IO.Ports.SerialPort.GetPortNames()
    puertos.Items.AddRange(myPort)
    btnConectar.Enabled = True
    btnEnviarDato.Enabled = False
    btnSalir.Enabled = False
End Sub

End Class
```


ANEXO 4:
SOFTWARE TFT

```

$regfile = "xm192d3def.dat"

$hwstack = 100

$swstack = 100

$framesize = 100

$crystal = 32000000

'$baud = 9600

Config Osc = Disabled , 32mhzosc = Enabled

Do

Loop Until Osc_status.1 = 1           'Check if RC2MRDY is ready

Set Osc_ctrl.4           'PLL enable

"configure the systemclock

Config Sysclock = 32mhz , Prescalea = 1 , Prescalebc = 1_1 'use PLL

Stop Watchdog

'Touch Interrupt

Const Use_touch = 1           'use Touch = 1

#if Use_touch = 1

    Config Pinf.0 = Input           'Set PINF.4 as Input

    'Because of the activated Pullup the PIN Level is high and there is a falling edge when switch is
    pressed

    Config Xpin = Portf.0 , Outpull = Pullup , Sense = Falling   'enable Pull up and reaction on falling
    edge

    Portf_int0mask = &B0000_0001           'include PIN0 in INTO Mask

    Portf_intctrl = &B0000_00_01

    'Low Level INTO Interrupt

    Dim Touch_flag As Bit

    Touch_flag = 0

    On Portf_int0 Touch_int

    Enable Portf_int0 , Lo

```

```
#endif

Config Priority = Static , Vector = Application , Lo = Enabled , Med = Enabled

Config Com5 = 9600 , Mode = Asynchronous , Parity = None , Stopbits = 1 , Databits = 8

Config Serialin = Buffered , Size = 200

Open "COM5:" For Binary As #5

Config Porte.7 = Output

Set Porte.7

$include "SSD1963_declarations.inc"

$include "SSD1963__sd-card_declarations.inc"

Waitms 100

Call Sdcard_initialize()

Gosub Sonido

Dim Touchxs As String * 10

Dim X As Word

Dim Y As Word

Dim Tecla As Byte

Dim Numero As Word

Dim N As Byte

Dim Startstop As Byte

Dim S As Byte

dim envio(40) as byte

dim envio1 as string*12

dim envio2 as byte

dim e1 as string*1

dim e2 as string*1

dim e3 as string*1
```

```
dim contador as integer
```

```
*****
```

```
'Start SSD1963
```

```
*****
```

```
Call Ssd1963_init()
```

```
*****
```

```
Enable Interrupts
```

```
*** MAIN *****
```

```
Waitms 100
```

```
Touch_flag = 0
```

```
Call Lcd_clear(black)
```

```
Call Lcd_draw_bmp( "logo.bmp" , 0 , 0)
```

```
Call Lcd_draw_bmp( "ok2.bmp" , 300 , 0)
```

```
Do
```

```
  Call Lcd_read_touch()
```

```
  If Touchx > 299 And Touchx < 401 Then
```

```
    If Touchy > 0 And Touchy < 51 Then
```

```
      Call Lcd_draw_bmp( "ok1.bmp" , 300 , 0)
```

```
      Gosub Sonido
```

```
      exit do
```

```
      'Tecla = 1
```

```
      'Touchxs = Str(touchx)
```

```
      'Touchxs = " 1 "
```

```
      'Call Lcd_text( " 1 " , 0 , 400 , 2 , White , Black)
```

```
    End If

End If

waitms 50

loop

Call Lcd_clear(black)

Call Lcd_draw_bmp( "inout.bmp" , 0 , 0)

Call Lcd_draw_bmp( "eoff.bmp" , 180 , 13)

Call Lcd_draw_bmp( "eoff.bmp" , 180 , 68)

Call Lcd_draw_bmp( "eoff.bmp" , 180 , 123)

Call Lcd_draw_bmp( "eoff.bmp" , 180 , 178)

Call Lcd_draw_bmp( "eoff.bmp" , 180 , 233)

Call Lcd_draw_bmp( "eoff.bmp" , 180 , 288)

Call Lcd_draw_bmp( "eoff.bmp" , 180 , 343)

Call Lcd_draw_bmp( "eoff.bmp" , 180 , 398)

Call Lcd_draw_bmp( "soff.bmp" , 515 , 30)

Call Lcd_draw_bmp( "soff.bmp" , 515 , 130)

Call Lcd_draw_bmp( "soff.bmp" , 515 , 230)

Call Lcd_draw_bmp( "soff.bmp" , 515 , 330)

waitms 50

Do

    contador = 1

    envio(contador) = waitkey(#5)

    envio(contador + 1) = waitkey(#5)

    envio(contador + 2) = waitkey(#5)

    envio(contador + 3) = waitkey(#5)

    envio(contador + 4) = waitkey(#5)
```

```

envio(contador + 5) = waitkey(#5)

envio(contador + 6) = waitkey(#5)

envio(contador + 7) = waitkey(#5)

envio(contador + 8) = waitkey(#5)

envio(contador + 9) = waitkey(#5)

envio(contador + 10) = waitkey(#5)

envio(contador + 11) = waitkey(#5)

print #5, chr(envio(contador));" ";chr(envio(contador + 1));" ";chr(envio(contador + 2)); " ";
chr(envio(contador + 3));" ";chr(envio(contador + 4));" ";chr(envio(contador + 5));" ";
chr(envio(contador + 6));" ";chr(envio(contador + 7));" ";chr(envio(contador + 8)); " ";
chr(envio(contador + 9)); " "; chr(envio(contador + 10)); " ";chr(envio(contador + 11))

contador = 1

if chr(envio(contador + 2)) = "A" then

    Call Lcd_draw_bmp( "eon.bmp" , 180 , 13)

elseif chr(envio(contador + 2)) = "a" then

    Call Lcd_draw_bmp( "eoff.bmp" , 180 , 13)

end if

'print #5, chr(envio1);" "

if chr(envio(contador + 1)) = "B" then

    Call Lcd_draw_bmp( "eon.bmp" , 180 , 68)

elseif chr(envio(contador + 1)) = "b" then

    Call Lcd_draw_bmp( "eoff.bmp" , 180 , 68)

end if

'print #5, chr(envio1); " "

if chr(envio(contador)) = "C" then

    Call Lcd_draw_bmp( "eon.bmp" , 180 , 123)

elseif chr(envio(contador)) = "c" then

    Call Lcd_draw_bmp( "eoff.bmp" , 180 , 123)

```

```
end if

'print #5, chr(envio1);" "

if chr(envio(contador + 11)) = "D" then

    Call Lcd_draw_bmp( "eon.bmp" , 180 , 178)

elseif chr(envio(contador + 11)) = "d" then

    Call Lcd_draw_bmp( "eoff.bmp" , 180 , 178)

end if

if chr(envio(contador + 10)) = "E" then

    Call Lcd_draw_bmp( "eon.bmp" , 180 , 233)

elseif chr(envio(contador + 10)) = "e" then

    Call Lcd_draw_bmp( "eoff.bmp" , 180 , 233)

end if

if chr(envio(contador + 9)) = "F" then

    Call Lcd_draw_bmp( "eon.bmp" , 180 , 288)

elseif chr(envio(contador + 9)) = "f" then

    Call Lcd_draw_bmp( "eoff.bmp" , 180 , 288)

end if

if chr(envio(contador + 8)) = "G" then

    Call Lcd_draw_bmp( "eon.bmp" , 180 , 343)

elseif chr(envio(contador + 8)) = "g" then

    Call Lcd_draw_bmp( "eoff.bmp" , 180 , 343)

end if

if chr(envio(contador + 7)) = "H" then

    Call Lcd_draw_bmp( "eon.bmp" , 180 , 398)

elseif chr(envio(contador + 7)) = "h" then

    Call Lcd_draw_bmp( "eoff.bmp" , 180 , 398)

end if
```

```
if chr(envio(contador + 6)) = "W" then
    Call Lcd_draw_bmp( "son.bmp" , 515 , 30)
elseif chr(envio(contador + 6)) = "w" then
    Call Lcd_draw_bmp( "soff.bmp" , 515 , 30)
end if

if chr(envio(contador + 5)) = "X" then
    Call Lcd_draw_bmp( "son.bmp" , 515 , 130)
elseif chr(envio(contador + 5)) = "x" then
    Call Lcd_draw_bmp( "soff.bmp" , 515 , 130)
end if

if chr(envio(contador + 4)) = "Y" then
    Call Lcd_draw_bmp( "son.bmp" , 515 , 230)
elseif chr(envio(contador + 4)) = "y" then
    Call Lcd_draw_bmp( "soff.bmp" , 515 , 230)
end if

if chr(envio(contador + 3)) = "Z" then
    Call Lcd_draw_bmp( "son.bmp" , 515 , 330)
elseif chr(envio(contador + 3)) = "z" then
    Call Lcd_draw_bmp( "soff.bmp" , 515 , 330)
end if

loop
End

Sonido:

Reset Porte.7

Waitms 100

Set Porte.7

waitms 50
```


Reset Porte.7

Waitms 100

Set Porte.7

Return

Touch_int:

Touch_flag = 1

Return

\$include "SSD1963__sd-card_routines.inc"

\$include "SSD1963_functions5.inc"

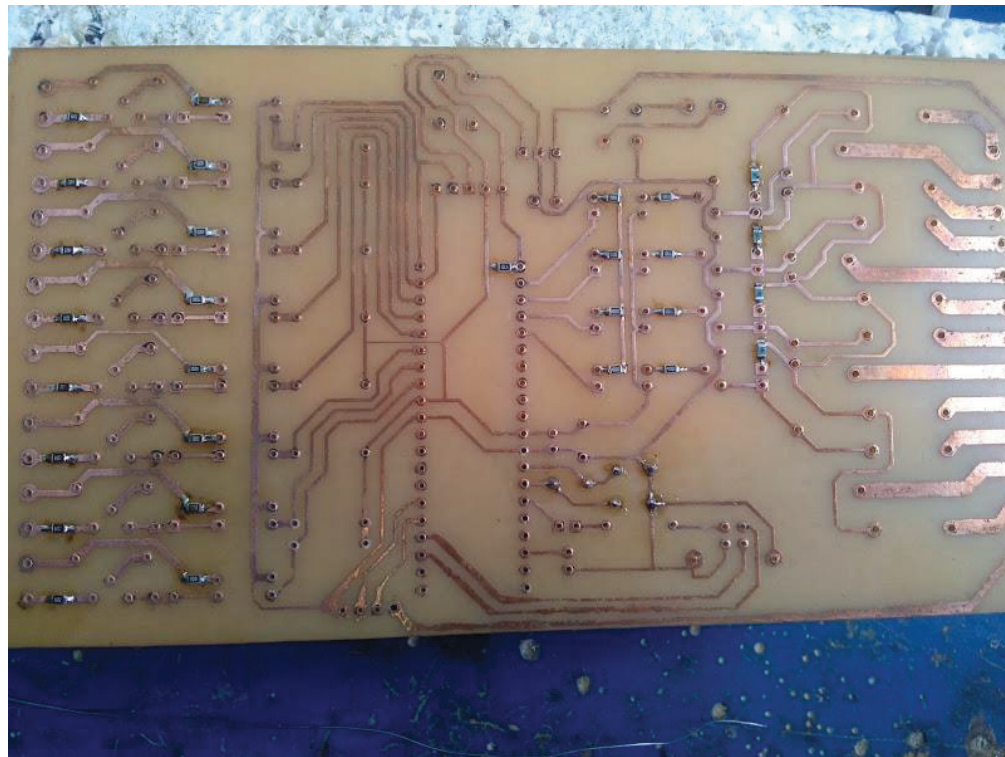
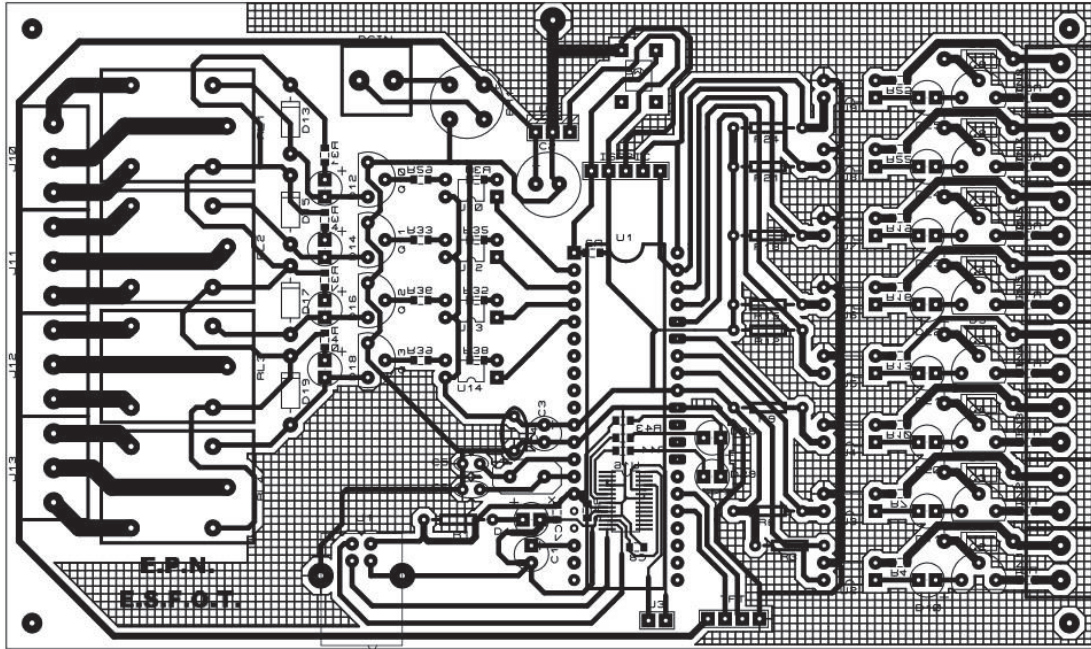
\$include "Font12x16.font"

\$include "font25x32.font"

\$include "font32x32.font"


ANEXO 5:
DISEÑO DEL CIRCUITO

ANEXO 6:
PLACA PLC



ANEXO 7:
DATA SHEETS

SONGLE RELAY

	RELAY ISO9002	SRD
---	---------------	------------



1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for highdensity P.C. board mounting technique.
- UL, CUL, TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.
(Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

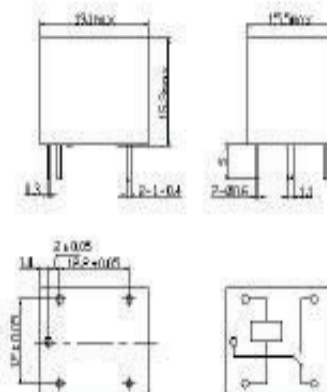
3. ORDERING INFORMATION

SRD	XX VDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil sensitivity	Contact form
SRD	03, 05, 06, 09, 12, 24, 48VDC	S: Sealed type	L: 0.36W	A: 1 form A
		F: Flux free type	D: 0.45W	B: 1 form B
				C: 1 form C

4. RATING

CCC	FILE NUMBER: CH0052885-2000	7A/240VDC
CCC	FILE NUMBER: CH0036746-99	10A/250VDC
UL/CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R9933789	10A/240VAC 28VDC

5. DIMENSION_(unit:mm) DRILLING_(unit:mm) WIRING DIAGRAM



6. COIL DATA CHART (AT20°C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) $\pm 10\%$	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75% Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500	abt. 0.51W			

7. CONTACT RATING

Item	Type	SRD	
		FORM C	FORM A
Contact Capacity		7A 28VDC	10A 28VDC
Resistive Load ($\cos\phi=1$)		10A 125VAC	10A 240VAC
		7A 240VAC	
Inductive Load ($\cos\phi=0.4$ L/R=7msec)		3A 120VAC	5A 120VAC
		3A 28VDC	5A 28VDC
Max. Allowable Voltage		250VAC/110VDC	250VAC/110VDC
Max. Allowable Power Force		800VAC/240W	1200VA/300W
Contact Material		AgCdO	AgCdO

8. PERFORMANCE (at initial value)

Item	Type	SRD
Contact Resistance		100m Ω Max.
Operation Time		10msec Max.
Release Time		5msec Max.
Dielectric Strength		
Between coil & contact		1500VAC 50/60HZ (1 minute)
Between contacts		1000VAC 50/60HZ (1 minute)
Insulation Resistance		100 M Ω Min. (500VDC)
Max. ON/OFF Switching		
Mechanically		300 operation/min
Electrically		30 operation/min
Ambient Temperature		-25°C to +70°C
Operating Humidity		45 to 85% RH
Vibration		
Endurance		10 to 55Hz Double Amplitude 1.5mm
Error Operation		10 to 55Hz Double Amplitude 1.5mm
Shock		
Endurance		100G Min.
Error Operation		10G Min.
Life Expectancy		
Mechanically		10 ⁷ operations. Min. (no load)
Electrically		10 ⁶ operations. Min. (at rated coil voltage)
Weight		abt. 10grs.

9. REFERENCE DATA





PIC18F2455/2550/4455/4550

Data Sheet

28/40/44-Pin, High-Performance,
Enhanced Flash, USB Microcontrollers
with nanoWatt Technology



MICROCHIP PIC18F2455/2550/4455/4550

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mbps) and Full Speed (12 Mbps)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1 Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, Peripherals on
- Idle: CPU off, Peripherals on
- Sleep: CPU off, Peripherals off
- Idle mode Currents Down to 5.8 μ A Typical
- Sleep mode Currents Down to 0.1 μ A Typical
- Timer1 Oscillator: 1.1 μ A Typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A Typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High-Precision PLL for USB
- Two External Clock modes, Up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator Options allow Microcontroller and USB module to Run at Different Clock Speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns (Tcy/16)
 - Compare is 16-bit, max. resolution 83.3 ns (Tcy)
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I²C™ Master and Slave modes
- 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

Special Microcontroller Features:

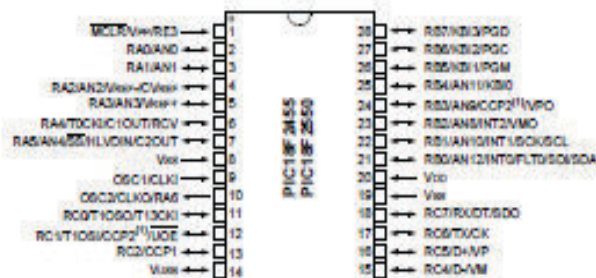
- C Compiler Optimized Architecture with Optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory Typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory Typical
- Flash/Data EEPROM Retention: > 40 Years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) via Two Pins
- Optional Dedicated ICD/ICSP Port (44-pin, TQFP package only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP			Timers 8/16-Bit	
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I ² C™	10-Bit I ² C™		Compare/Match
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

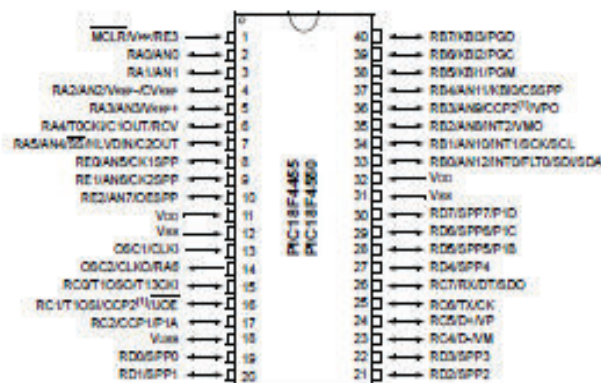
PIC18F2455/2550/4455/4550

Pin Diagrams

28-Pin PDIP, SOIC



40-Pin PDIP



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

PIC18F2455/2550/4455/4550

TABLE 1-1: DEVICE FEATURES

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, <small>RESET</small> Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, <small>RESET</small> Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, <small>RESET</small> Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, <small>RESET</small> Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-Pin PDIP 28-Pin SOIC	28-Pin PDIP 28-Pin SOIC	40-Pin PDIP 44-Pin QFN 44-Pin TQFP	40-Pin PDIP 44-Pin QFN 44-Pin TQFP

PIC18F2455/2550/4455/4550

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	PDIP, SOIC			
RB0/AN12/INT0/FLT0/SDI/SDA	21			PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0		I/O	TTL	Digital I/O.
AN12		I	Analog	Analog input 12.
INT0		I	ST	External interrupt 0.
FLT0		I	ST	PWM Fault Input (CCP1 module).
SDI		I	ST	SPI data in.
SDA		I/O	ST	I ² C™ data I/O.
RB1/AN10/INT1/SCK/SCL	22			
RB1		I/O	TTL	Digital I/O.
AN10		I	Analog	Analog input 10.
INT1		I	ST	External interrupt 1.
SCK		I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL		I/O	ST	Synchronous serial clock input/output for I ² C mode.
RB2/AN8/INT2/VMO	23			
RB2		I/O	TTL	Digital I/O.
AN8		I	Analog	Analog input 8.
INT2		I	ST	External interrupt 2.
VMO		O	—	External USB transceiver VMO output.
RB3/AN9/CCP2/VPO	24			
RB3		I/O	TTL	Digital I/O.
AN9		I	Analog	Analog input 9.
CCP2 ⁽¹⁾		I/O	ST	Capture 2 input/Compare 2 output/PWM2 output.
VPO	O	—	External USB transceiver VPO output.	
RB4/AN11/KBI0	25			
RB4		I/O	TTL	Digital I/O.
AN11		I	Analog	Analog input 11.
KBI0		I	TTL	Interrupt-on-change pin.
RB5/KBI1/PGM	26			
RB5		I/O	TTL	Digital I/O.
KBI1		I	TTL	Interrupt-on-change pin.
PGM	I/O	ST	Low-Voltage ICSP™ Programming enable pin.	
RB6/KBI2/PGC	27			
RB6		I/O	TTL	Digital I/O.
KBI2		I	TTL	Interrupt-on-change pin.
PGC	I/O	ST	In-Circuit Debugger and ICSP programming clock pin.	
RB7/KBI3/PGD	28			
RB7		I/O	TTL	Digital I/O.
KBI3		I	TTL	Interrupt-on-change pin.
PGD	I/O	ST	In-Circuit Debugger and ICSP programming data pin.	

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels I = Input
 O = Output P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.



NHD-7.0-800480WF-CTXI#-T

TFT (Thin-Film-Transistor) Color Liquid Crystal Display Module

NHD-	Newhaven Display
7.0-	7.0" Diagonal
800480-	800xRGBx480 pixels
WF-	Model
C-	Built-in Controller
T-	White LED backlight
X-	TFT
I-	6:00 view, Wide Temp
#-	RoHS Compliant
T-	Touch Panel

Newhaven Display International, Inc.

2511 Technology Drive, Suite 101
Elgin IL, 60124

Ph: 847-844-8795 Fax: 847-844-8796

www.newhavendisplay.com

nhtech@newhavendisplay.com

nhsales@newhavendisplay.com

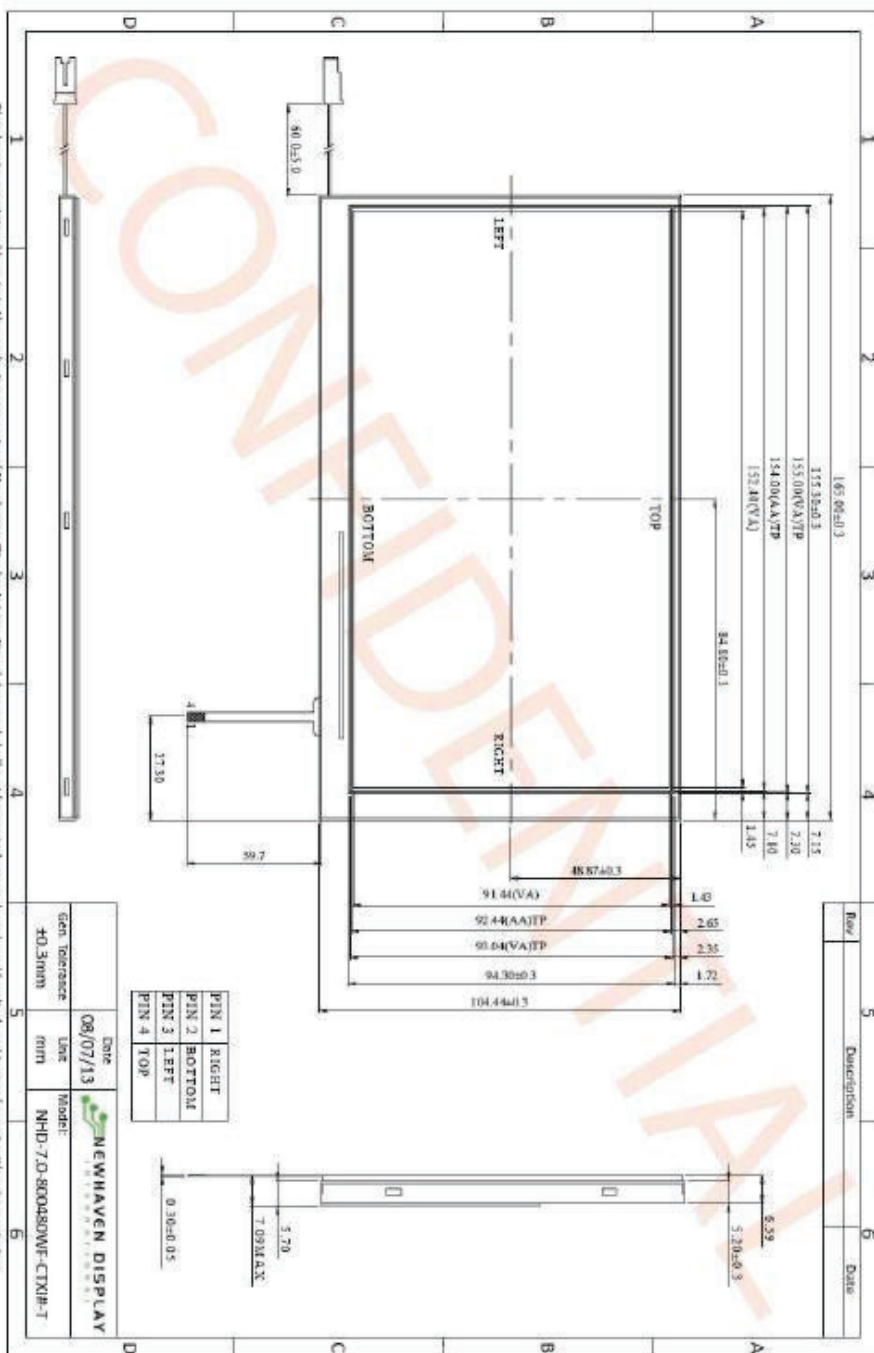
Document Revision History

Revision	Date	Description	Changed by
0	8/31/2010	Initial Release	MC
1	10/5/2010	Mechanical Drawing updated	BE
2	2/8/2011	Initialization code added	AK
3	2/23/2011	Grammar/symbols updated	BE
4	10/7/2011	Backlight and Touch Panel pin descriptions added	AK
5	1/6/2012	Pixel data format updated	AK
6	8/7/2013	Updated Functions and Features, Mechanical Drawing, Electrical and Optical Characteristics.	JN

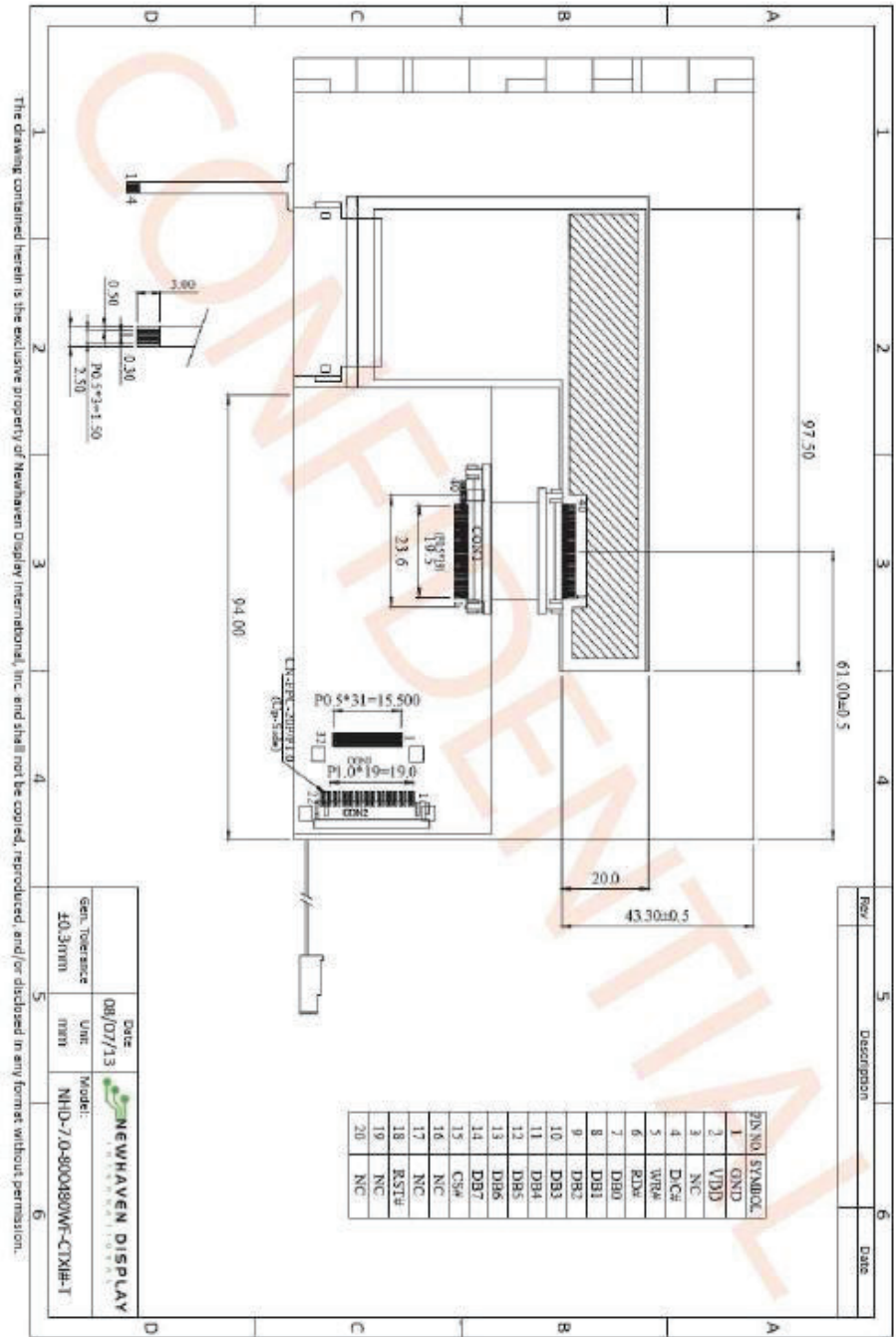
Functions and Features

- 800xRGBx480 resolution
- LED backlight
- 8-bit MCU interface
- 262,144 colors
- SSD1963 8-bit controller

Mechanical Drawing



The drawing contained herein is the exclusive property of Newhaven Display International, Inc. and shall not be copied, reproduced, and/or disclosed in any format without permission.



This drawing contained herein is the exclusive property of Newhaven Display International, Inc. and shall not be copied, reproduced, and/or disclosed in any format without permission.

PIN NO.	SYMBOL
1	GNND
2	VIDD
3	NC
4	D/Cd
5	WRd
6	RDX
7	DB0
8	DB1
9	DB2
10	DB3
11	DB4
12	DB5
13	DB6
14	DB7
15	CS#
16	NC
17	NC
18	RST#
19	NC
20	NC

Gen. Tolerance	±0.3mm
Date	08/07/13
Unit	mm
Model	NHD-7.0-900890WF-CTX#-1
NEHAVEN DISPLAY	

Rev	Description	Date
5		

CON2 LCD Pin Description

Pin No.	Symbol	Connection	Function Description
1	GND	Power Supply	Ground
2	VDD	Power Supply	Power supply for logic (+3.3V)
3	NC	-	No Connect
4	D/C#	MPU	Register Select signal: 1=Data, 0=Command
5	WR#	MPU	Active LOW Write signal (8080 mode) Read/Write signal (6800 mode)
6	RD#	MPU	Active LOW Read signal (8080 mode) Edge trigger Enable signal (6800 mode)
7-14	DB0-DB7	MPU	8-bit bi-directional data bus
15	CS#	MPU	Active LOW Chip Select signal
16	NC	-	No Connect
17	NC	-	No Connect
18	RST#	MPU	Active LOW Reset signal
19	NC	-	No Connect
20	NC	-	No Connect

LCD connector: 1.0mm pitch,20-Conductor FFC, top contact

Recommended connection: 1.0mm pitch, 20-conductor FFC cable

CON1 TFT Pin-Out (before controller board):

Pin No.	Symbol	Connection	Function Description
1	GND	Power Supply	Power Ground
2	GND	Power Supply	Power Ground
3	NC	-	No connect
4-7	VDD	Power Supply	Power Supply (+3.3V)
8	NC	-	No connect
9	DE	MPU	Data Enable
10-12	GND	Power Supply	Power Ground
13-15	B5-B3	MPU	Blue B5 (MSB) to B3
16	GND	Power Supply	Power Ground
17-19	B2-B0	MPU	Blue B2 – B0(LSB)
20	GND	Power Supply	Power Ground
21-23	G5-G3	MPU	Green B5 (MSB) to B3
24	GND	Power Supply	Power Ground
25-27	G2-G0	MPU	Green B2 – B0(LSB)
28	GND	Power Supply	Power Ground
29-31	R5-R3	MPU	Red B5 (MSB) to B3
32	GND	Power Supply	Power Ground
33-35	R2-R0	MPU	Red B2 – B0(LSB)
36-37	GND	Power Supply	Power Ground
38	DCLK	MPU	Clock (Falling edge triggered)
39-40	GND	Power Supply	Power Ground

LCD connector: 0.5mm pitch,40-Conductor FFC, Bottom contact

Recommended connection: 0.5mm pitch, 40-conductor FFC cable

Backlight Pin-Out:

Pin No.	Symbol	Connection	Function Description
1	VDD	Power Supply	Power Supply (+9.9V)
2	GND	Power Supply	Power Ground

Backlight connector: JST p/n: BHSR-02V5-1

Recommended mating connector: JST p/n: SM 02B-BHSS-1

Touch Panel Pin-Out:

Pin No.	Symbol	Connection	Function Description
1	X-	Touch Panel	Touch Panel – RIGHT
2	Y-	Touch Panel	Touch Panel – BOTTOM
3	X+	Touch Panel	Touch Panel – LEFT
4	Y+	Touch Panel	Touch Panel – TOP

Touch Panel Characteristics

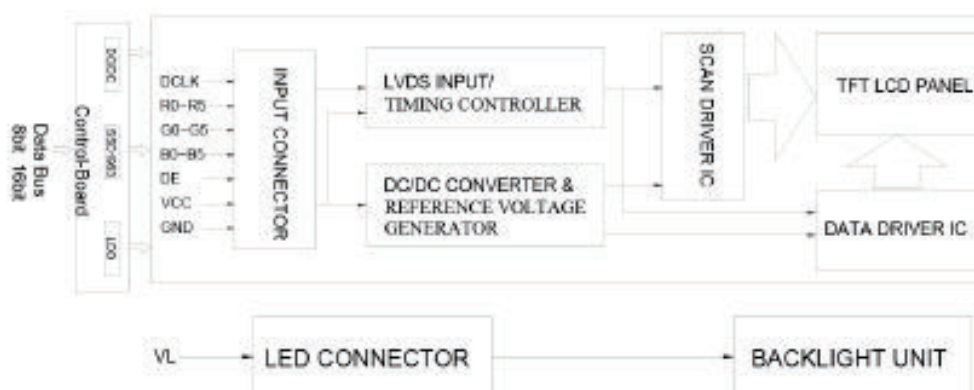
Item	Min.	Typ.	Max.	Unit
Linearity	-	-	1.5	%
Circuit Resistance – X-Axis	450	800	1300	Ω
Circuit Resistance – Y-Axis	100	350	800	Ω
Insulation Resistance	10	-	-	MΩ
Operating Voltage	-	-	5	V
Chattering	-	-	10	ms
Transmittance	82	-	-	%
Activation Force	50	-	200	g
Pen Writing Durability	100,000	-	-	Characters
Pitting Durability	1,000,000	-	-	Touches
Surface Hardness	3	-	-	H
Haze	-	7	-	%

Electrical Characteristics

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating Temperature Range	Top	Absolute Max	-20	-	+70	°C
Storage Temperature Range	Tst	Absolute Max	-30	-	+80	°C
Supply Voltage	VDD		3.0	3.3	3.6	V
Supply Current	IDD	VDD=3.3V 25°C	-	200	260	mA
"H" Level Input	Vih		0.8V*DD	-	VDD+ 0.5	V
"L" Level Input	Vil		-	-	0.2*VDD	V
Backlight Supply Voltage	VLED		-	9.9	-	V
Backlight Supply Current	ILED	VLED=9.9V	140	160	180	mA
Backlight Lifetime	-	Until half-brightness	10,000	20,000	-	Hrs.

Optical Characteristics

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Viewing Angle – Top		Cr ≥10	-	50	-	°
Viewing Angle – Bottom			-	70	-	°
Viewing Angle – Left			-	70	-	°
Viewing Angle – Right			-	70	-	°
Contrast Ratio	Cr	-	250	400	-	-
Luminance	YL	-	300	350	-	cd/m ²
Response Time (rise)	Tr	-	-	5	10	ms
Response Time (fall)	Tf	-	-	11	16	ms



Controller Information

Built-in SSD1963 controller.

Please download specification at http://www.newhavendisplay.com/app_notes/SSD1963.pdf

PC817 Series

High Density Mounting Type Photocoupler

- Lead forming type (I type) and taping reel type (P type) are also available. (PC817V/PC817P)
- TUV (VDE8884) approved type is also available as an option.

■ Features

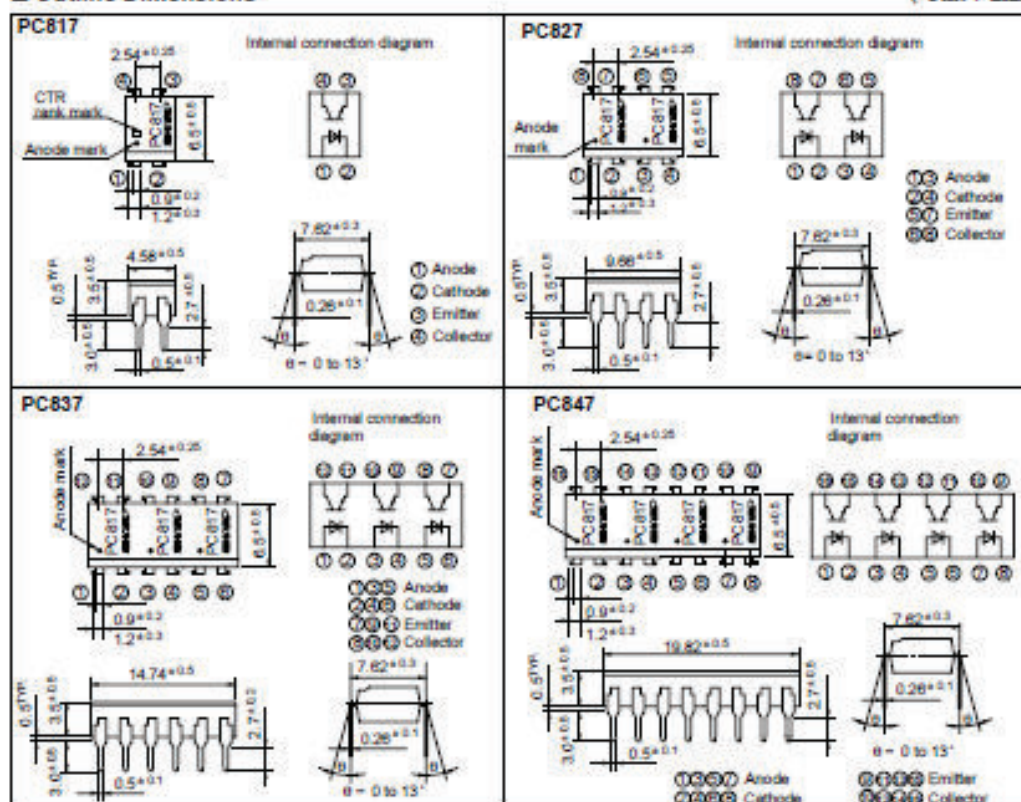
1. Current transfer ratio
(CTR: MIN. 50% at $I_f = 5\text{mA}$, $V_{CE} = 5\text{V}$)
2. High isolation voltage between input and output (V_{iso} : 5 000V_{max})
3. Compact dual-in-line package
PC817 : 1-channel type
PC827 : 2-channel type
PC837 : 3-channel type
PC847 : 4-channel type
4. Recognized by UL, file No. E64380

■ Applications

1. Computer terminals
2. System appliances, measuring instruments
3. Registers, copiers, automatic vending machines
4. Electric home appliances, such as fan heaters, etc.
5. Signal transmission between circuits of different potentials and impedances

■ Outline Dimensions

(Unit : mm)



* In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that occur in equipment using any of SHARP's devices, shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest version of the device specification sheets before using any SHARP's device.*

Absolute Maximum Ratings

(Ta = 25°C)

Parameter		Symbol	Rating	Unit
Input	Forward current	I_F	50	mA
	¹⁾ Peak forward current	I_{FM}	1	A
	Reverse voltage	V_R	6	V
	Power dissipation	P	70	mW
Output	Collector-emitter voltage	V_{CE0}	35	V
	Emitter-collector voltage	V_{EC0}	6	V
	Collector current	I_C	50	mA
	Collector power dissipation	P_C	150	mW
	Total power dissipation	P_{tot}	200	mW
²⁾ Isolation voltage		V_{iso}	5 000	V_{max}
Operating temperature		T_{op}	-30 to +100	°C
Storage temperature		T_{stg}	-55 to +125	°C
³⁾ Soldering temperature		T_{sol}	260	°C

*1 Pulse width \rightarrow 100 μ s, Duty ratio : 0.001

*2 40 to 60%RH, AC for 1 minute

*3 For 10 seconds

Electro-optical Characteristics

(Ta = 25°C)

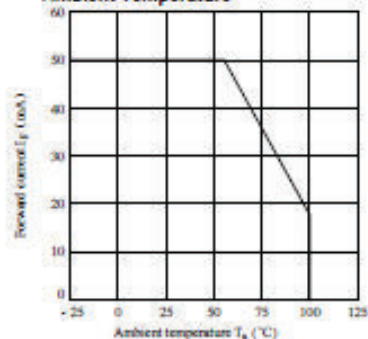
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input	Forward voltage	V_F	$I_F = 20\text{mA}$	-	1.2	1.4	V	
	Peak forward voltage	V_{FM}	$I_{FM} = 0.5\text{A}$	-	-	3.0	V	
	Reverse current	I_R	$V_R = 4\text{V}$	-	-	10	μA	
	Terminal capacitance	C_i	$V = 0, f = 1\text{kHz}$	-	30	250	pF	
Output	Collector dark current	I_{C0}	$V_{CE} = 20\text{V}$	-	-	10^{-7}	A	
Transfer characteristics	⁴⁾ Current transfer ratio	CTR	$I_F = 5\text{mA}, V_{CE} = 5\text{V}$	50	-	600	%	
	Collector-emitter saturation voltage	$V_{CE(sat)}$	$I_F = 20\text{mA}, I_C = 1\text{mA}$	-	0.1	0.2	V	
	Isolation resistance	R_{iso}	DC500V, 40 to 60%RH	5×10^{10}	10^{11}	-	Ω	
	Floating capacitance	C_f	$V = 0, f = 1\text{MHz}$	-	0.6	1.0	pF	
	Cut-off frequency	Response time	Rise time	t_r	-	4	18	μs
			$V_{CE} = 2\text{V}, I_C = 2\text{mA}, R_L = 100\Omega$	-	-	-	-	

*4 Classification table of current transfer ratio is shown below.

Model No.	Rank mark	CTR (%)
PCS17A	A	80 to 160
PCS17B	B	130 to 260
PCS17C	C	200 to 400
PCS17D	D	300 to 600
PCS●7AB	A or B	80 to 260
PCS●7BC	B or C	130 to 400
PCS●7CD	C or D	200 to 600
PCS●7AC	A, B or C	80 to 400
PCS●7BD	B, C or D	130 to 600
PCS●7AD	A, B, C or D	80 to 600
PCS●7	A, B, C, D or No mark	50 to 600

● : 1 or 2 or 3 or 4

Fig. 1 Forward Current vs. Ambient Temperature



Future Technology Devices International Ltd.

FT232R USB UART IC

The FT232R is a USB to serial UART interface with the following advanced features:

- Single chip USB to asynchronous serial data transfer interface.
- Entire USB protocol handled on the chip. No USB specific firmware programming required.
- Fully integrated 1024 bit EEPROM storing device descriptors and CBUS I/O configuration. □ Fully integrated USB termination resistors.
- Fully integrated clock generation with no external crystal required plus optional clock output selection enabling a glue-less interface to external MCU or FPGA.
- Data transfer rates from 300 baud to 3 Mbaud (RS422, RS485, RS232) at TTL levels.
- 128 byte receive buffer and 256 byte transmit buffer utilising buffer smoothing technology to allow for high data throughput.
- FTDI's royalty-free Virtual Com Port (VCP) and Direct (D2XX) drivers eliminate the requirement for USB driver development in most cases.
- Unique USB FTDIChip-ID™ feature.
- Configurable CBUS I/O pins.
- Transmit and receive LED drive signals.
- UART interface support for 7 or 8 data bits, 1 or 2 stop bits and odd / even / mark / space / no parity
- FIFO receives and transmits buffers for high data throughput.
- Synchronous and asynchronous bit bang interface options with RD# and WR# strobes.
- Device supplied pre-programmed with unique USB serial number.
- Supports bus powered, self-powered and highpower bus powered USB configurations.
- Integrated +3.3V level converter for USB I/O.
- Integrated level converter on UART and CBUS for interfacing to between +1.8V and +5V logic.
- True 5V/3.3V/2.8V/1.8V CMOS drive output and TTL input.
- Configurable I/O pin output drive strength.
- Integrated power-on-reset circuit.
- Fully integrated AVCC supply filtering - no external filtering required.
- UART signal inversion option.
- +3.3V (using external oscillator) to +5.25V (internal oscillator) Single Supply Operation.
- Low operating and USB suspend current.
- Low USB bandwidth consumption.
- UHCI/OHCI/EHCI host controller compatible.
- USB 2.0 Full Speed compatible.
- -40°C to 85°C extended operating temperature range.
- Available in compact Pb-free 28 Pin SSOP and QFN-32 packages (both RoHS compliant).

Typical Applications

- USB to RS232/RS422/RS485 Converters
- Upgrading Legacy Peripherals to USB
- Cellular and Cordless Phone USB data transfer cables and interfaces
- Interfacing MCU/PLD/FPGA based designs to USB

- USB Audio and Low Bandwidth Video data transfer
- PDA to USB data transfer
- USB Smart Card Readers
- USB Instrumentation
- USB Industrial Control
- USB MP3 Player Interface
- USB FLASH Card Reader and Writers
- Set Top Box PC - USB interface
- USB Digital Camera Interface
- USB Hardware Modems
- USB Wireless Modems
- USB Bar Code Readers
- USB Software and Hardware Encryption Dongles

Driver Support

Royalty free VIRTUAL COM PORT (VCP)

DRIVERS for...

- Windows 10 32,64-bit
- Windows 8/8.1 32,64-bit
- Windows 7 32,64-bit
- Windows Vista and Vista 64-bit
- Windows XP and XP 64-bit
- Windows 98, 98SE, ME, 2000, Server 2003, XP, Server 2008 and server 2012 R2
- Windows XP Embedded
- Windows CE 4.2, 5.0 and 6.0
- Mac OS 8/9, OS-X
- Linux 2.4 and greater

Royalty free D2XX Direct Drivers

(USB Drivers + DLL S/W Interface)

- Windows 10 32,64-bit
- Windows 8/8.1 32,64-bit
- Windows 7 32,64-bit
- Windows Vista and Vista 64-bit
- Windows XP and XP 64-bit
- Windows 98, 98SE, ME, 2000, Server 2003, XP, Server 2008 and server 2012 R2
- Windows XP Embedded
- Windows CE 4.2, 5.0 and 6.0
- Linux 2.4 and greater
- Android(J2xx)

The drivers listed above are all available to download for free from FTDI website (www.ftdichip.com). Various 3rd party drivers are also available for other operating systems - see FTDI website (www.ftdichip.com) for details.

For driver installation, please refer to <http://www.ftdichip.com/Documents/InstallGuides.htm>

Part Numbers

Part Number	Package
FT232RQ-xxxx	32 Pin QFN
FT232RL-xxxx	28 Pin SSOP

Note: Packing codes for xxxx is:

- Reel: Taped and Reel, (SSOP is 2,000pcs per reel, QFN is 6,000pcs per reel).
- Tube: Tube packing, 47pcs per tube (SSOP only)
- Tray: Tray packing, 490pcs per tray (QFN only)

For example: FT232RQ-Reel is 6,000pcs taped and reel packing

USB Compliant

The FT232R is fully compliant with the USB 2.0 specification and has been given the USB-IF Test-ID (TID) 40680004 (Rev B) and 40770018 (Rev C).



FT232R Block Diagram

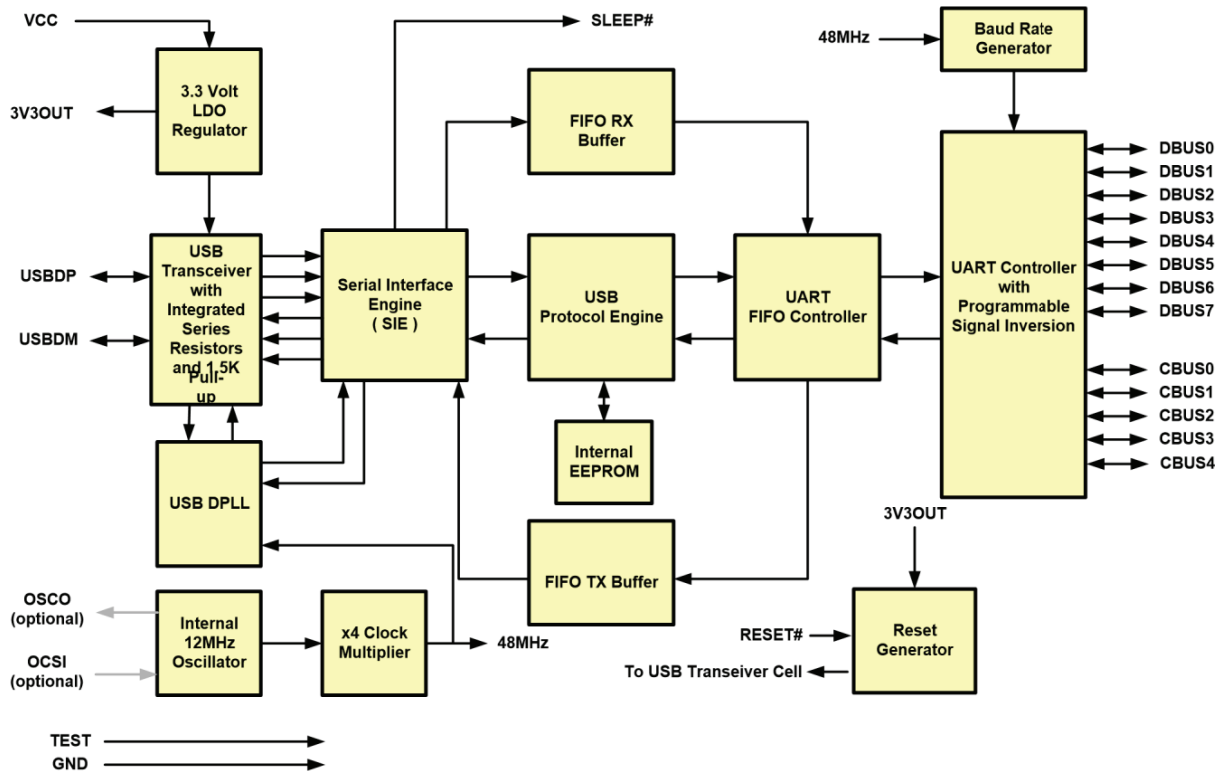


Figure 2.1 FT232R Block Diagram

For a description of each function please refer to Section 4.

Device Pin Out and Signal Description

28-LD SSOP Package

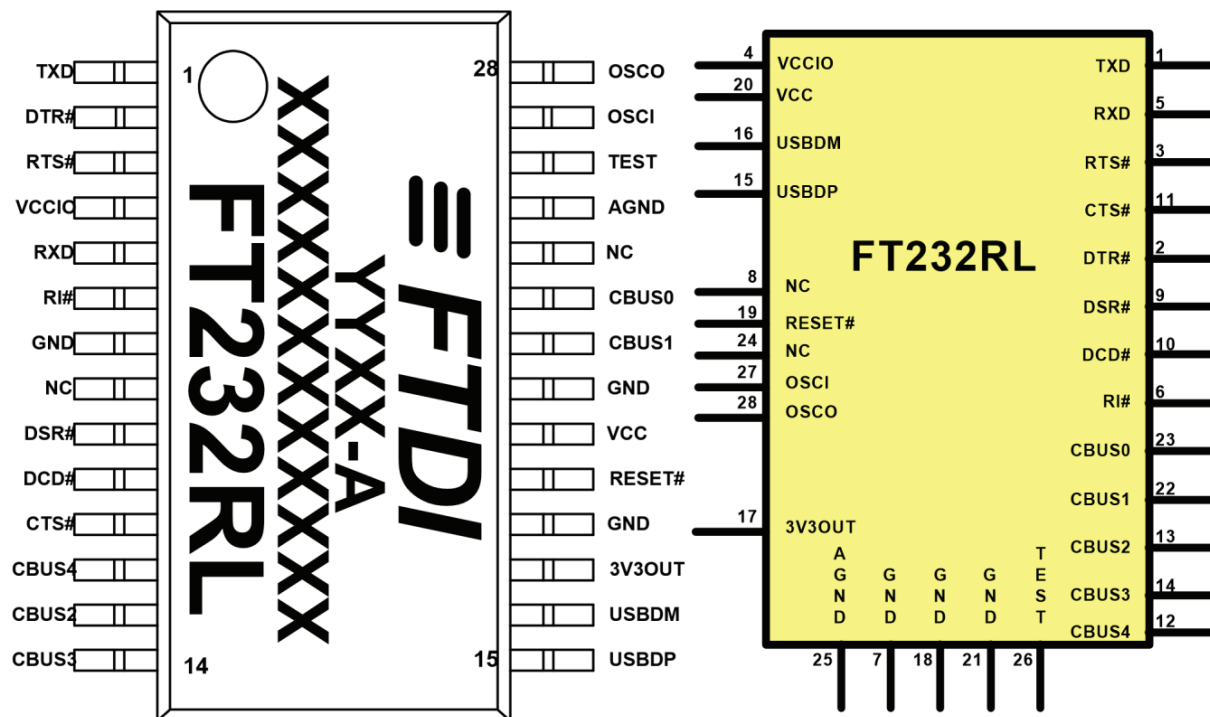


Figure 3.1 SSOP Package Pin Out and Schematic Symbol

SSOP Package Pin Out Description

Note: The convention used throughout this document for active low signals is the signal name followed by #

Pin No.	Name	Type	Description
15	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5kΩ pull up resistor to 3.3V.
16	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor.

Table 3.1 USB Interface Group

Pin No.	Name	Type	Description
---------	------	------	-------------

4	VCCIO	PWR	+1.8V to +5.25V supply to the UART Interface and CBUS group pins (1...3, 5, 6, 9...14, 22, 23). In USB bus powered designs connect this pin to 3V3OUT pin to drive out at +3.3V levels, or connect to VCC to drive out at 5V CMOS level. This pin can also be supplied with an external +1.8V to +2.8V supply in order to drive outputs at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to VCC. This means that in bus powered designs a regulator which is supplied by the +5V on the USB bus should
Pin No.	Name	Type	Description
			be used.
7, 18, 21	GND	PWR	Device ground supply pins
17	3V3OUT	Output	+3.3V output from integrated LDO regulator. This pin should be decoupled to ground using a 100nF capacitor. The main use of this pin is to provide the internal +3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the VCCIO pin.
20	VCC	PWR	+3.3V to +5.25V supply to the device core. (see Note 1)
25	AGND	PWR	Device analogue ground supply for internal clock multiplier

Table 3.2 Power and Ground Group

Pin No.	Name	Type	Description
8, 24	NC	NC	No internal connection
19	RESET#	Input	Active low reset pin. This can be used by an external device to reset the FT232R. If not required can be left unconnected, or pulled up to VCC.
26	TEST	Input	Puts the device into IC test mode. Must be tied to GND for normal operation, otherwise the device will appear to fail.
27	OSCI	Input	Input 12MHz Oscillator Cell. Optional – Can be left unconnected for normal operation. (see Note 2)
28	OSCO	Output	Output from 12MHZ Oscillator Cell. Optional – Can be left unconnected for normal operation if internal Oscillator is used. (see Note 2)

Table 3.3 Miscellaneous Signal Group

Pin No.	Name	Type	Description
1	TXD	Output	Transmit Asynchronous Data Output.
2	DTR#	Output	Data Terminal Ready Control Output / Handshake Signal.
3	RTS#	Output	Request to Send Control Output / Handshake Signal.
5	RXD	Input	Receiving Asynchronous Data Input.

6	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low (20ms active low pulse) can be used to resume the PC USB host controller from suspend.
9	DSR#	Input	Data Set Ready Control Input / Handshake Signal.
10	DCD#	Input	Data Carrier Detect Control Input.
Pin No.	Name	Type	Description
11	CTS#	Input	Clear To Send Control Input / Handshake Signal.
12	CBUS4	I/O	Configurable CBUS output only Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is SLEEP#. See CBUS Signal Options, Table 3.99.
13	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is TXDEN. See CBUS Signal Options, Table 3.99.
14	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is PWREN#. See CBUS Signal Options, Table 3.99. PWREN# should be used with a 10kΩ resistor pull up.
22	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is RXLED#. See CBUS Signal Options, Table 3.99.
23	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is TXLED#. See CBUS Signal Options, Table 3.99.

Table 3.4 UART Interface and CUSB Group (see note 3)

Notes:

1. The minimum operating voltage VCC must be +4.0V (could use VBUS=+5V) when using the internal clock generator. Operation at +3.3V is possible using an external crystal oscillator.
2. For details on how to use an external crystal, ceramic resonator, or oscillator with the FT232R, please refer Section 7.6
3. When used in Input Mode, the input pins are pulled to VCCIO via internal 200kΩ resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

QFN-32 Package

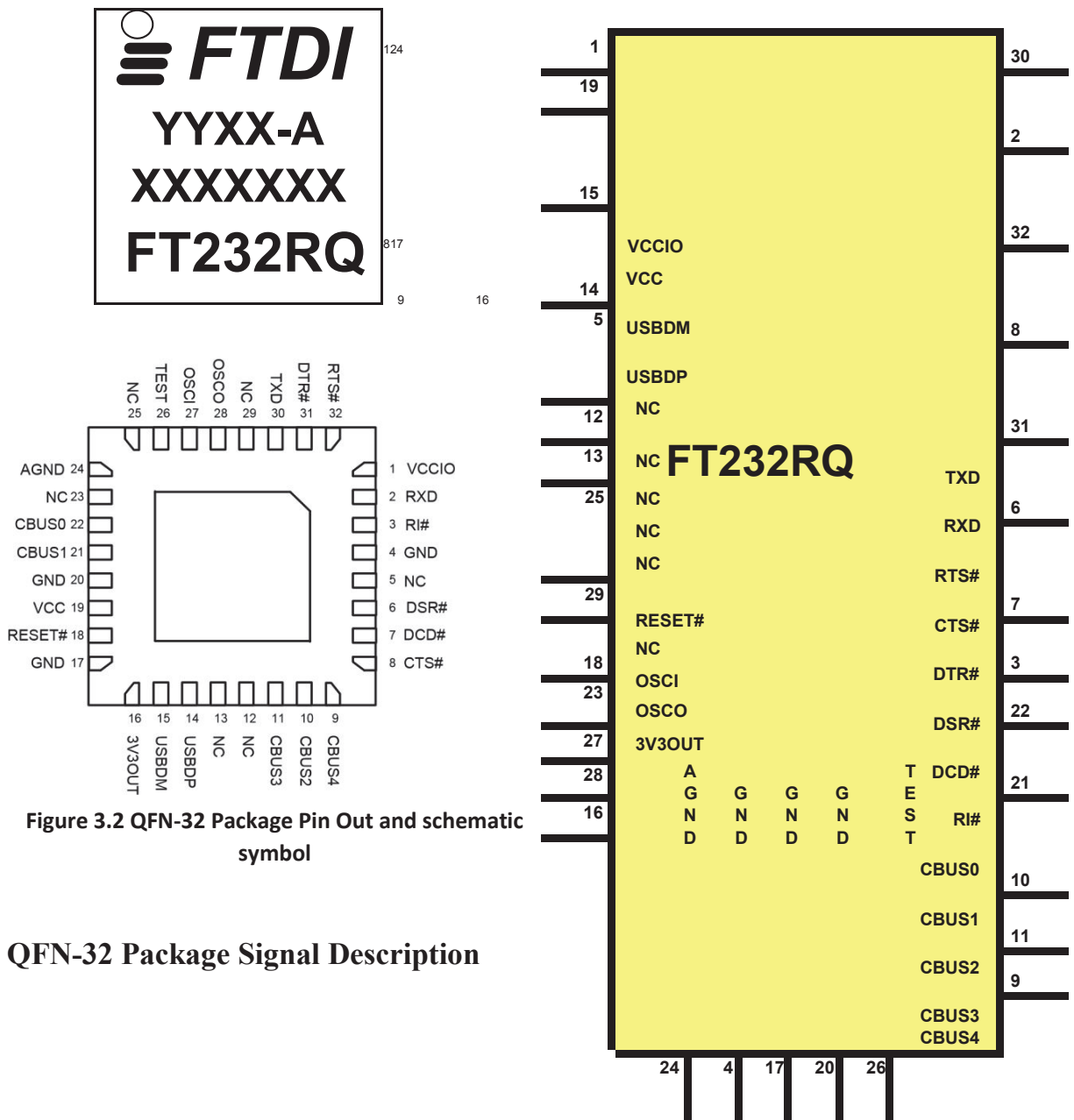


Figure 3.2 QFN-32 Package Pin Out and schematic symbol

QFN-32 Package Signal Description

Pin No.	Name	Type	Description
14	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5kΩ pull up resistor to +3.3V.
15	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor.

Table 3.5 USB Interface Group

Pin No.	Name	Type	Description
---------	------	------	-------------

1	VCCIO	PWR	+1.8V to +5.25V supply for the UART Interface and CBUS group pins (2,3,6,7,8,9,10,11,21,22,30,31,32). In USB bus powered designs connect this pin to 3V3OUT to drive out at +3.3V levels, or connect to VCC to drive out at +5V CMOS level. This pin can also be supplied with an external +1.8V to +2.8V supply in order to drive out at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to VCC. This means that in bus
Pin No.	Name	Type	Description
			powered designs a regulator which is supplied by the +5V on the USB bus should be used.
4, 17, 20	GND	PWR	Device ground supply pins.
16	3V3OUT	Output	+3.3V output from integrated LDO regulator. This pin should be decoupled to ground using a 100nF capacitor. The purpose of this output is to provide the internal +3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the VCCIO pin.
19	VCC	PWR	+3.3V to +5.25V supply to the device core. (See Note 1).
24	AGND	PWR	Device analogue ground supply for internal clock multiplier.

Table 3.6 Power and Ground Group

Pin No.	Name	Type	Description
5, 12, 13, 23, 25, 29	NC	NC	No internal connection. Do not connect.
18	RESET#	Input	Active low reset. Can be used by an external device to reset the FT232R. If not required can be left unconnected, or pulled up to VCC.
26	TEST	Input	Puts the device into IC test mode. Must be tied to GND for normal operation, otherwise the device will appear to fail.
27	OSCI	Input	Input 12MHz Oscillator Cell. Optional – Can be left unconnected for normal operation. (See Note 2).
28	OSCO	Output	Output from 12MHz Oscillator Cell. Optional – Can be left unconnected for normal operation if internal Oscillator is used. (See Note 2).

Table 3.7 Miscellaneous Signal Group

Pin No.	Name	Type	Description
30	TXD	Output	Transmit Asynchronous Data Output.
31	DTR#	Output	Data Terminal Ready Control Output / Handshake Signal.
32	RTS#	Output	Request to Send Control Output / Handshake Signal.
2	RXD	Input	Receiving Asynchronous Data Input.

3	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low (20ms active low pulse) can be used to resume the PC USB host controller from suspend.
6	DSR#	Input	Data Set Ready Control Input / Handshake Signal.
7	DCD#	Input	Data Carrier Detect Control Input.
Pin No.	Name	Type	Description
8	CTS#	Input	Clear To Send Control Input / Handshake Signal.
9	CBUS4	I/O	Configurable CBUS output only Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is SLEEP#. See CBUS Signal Options, Table 3.99.
10	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is TXDEN. See CBUS Signal Options, Table 3.99.
11	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is PWREN#. See CBUS Signal Options, Table 3.99. PWREN# should be used with a 10kΩ resistor pull up.
21	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is RXLED#. See CBUS Signal Options, Table 3.99.
22	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory default configuration is TXLED#. See CBUS Signal Options, Table 3.99.

Table 3.8 UART Interface and CBUS Group (see note 3)

Notes:

1. The minimum operating voltage VCC must be +4.0V (could use VBUS=+5V) when using the internal clock generator. Operation at +3.3V is possible using an external crystal oscillator.
2. For details on how to use an external crystal, ceramic resonator, or oscillator with the FT232R, please refer to Section 7.6.
3. When used in Input Mode, the input pins are pulled to VCCIO via internal 200kΩ resistors. These pins can be programmed to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

CBUS Signal Options

The following options can be configured on the CBUS I/O pins. CBUS signal options are common to both package versions of the FT232R. These options can be configured in the internal EEPROM using the software utility FT_PPLOG or MPROG, which can be downloaded from the FTDI Utilities (www.ftdichip.com). The default configuration is described in Section 8.

CBUS Signal Option	Available On CBUS Pin	Description
--------------------	-----------------------	-------------

TXDEN	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Enable transmit data for RS485
PWREN#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Output is low after the device has been configured by USB, then high during USB suspending mode. This output can be used to control power to external logic P-Channel logic level MOSFET switch. Enable the interface pull-down option when using the PWREN# in this way.*
TXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Transmit data LED drive: Data from USB Host to FT232R. Pulses low when transmitting data via USB. See Section 7.5 for more details.
RXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Receive data LED drive: Data from FT232R to USB Host. Pulses low when receiving data via USB. See Section 7.5 for more details.
TX&RXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	LED drive – pulses low when transmitting or receiving data via USB. See Section 7.5 for more details.
SLEEP#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Goes low during USB suspend mode. Typically used to power down an external TTL to RS232 level converter IC in USB to RS232 converter designs.
CLK48	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	48MHz $\pm 0.7\%$ Clock output. **
CLK24	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	24 MHz Clock output.**
CLK12	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	12 MHz Clock output.**
CLK6	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	6 MHz $\pm 0.7\%$ Clock output. **
CBitBangI/O	CBUS0, CBUS1, CBUS2, CBUS3	CBUS bit bang mode option. Allows up to 4 of the CBUS pins to be used as general purpose I/O. Configured individually for CBUS0, CBUS1, CBUS2 and CBUS3 in the internal EEPROM. A separate application note, AN232R-01, available from FTDI website (www.ftdichip.com) describes in more detail how to use CBUS bit bang mode.
BitBangWRn	CBUS0, CBUS1, CBUS2, CBUS3	Synchronous and asynchronous bit bang mode WR# strobe output.
BitBangRDn	CBUS0, CBUS1, CBUS2, CBUS3	Synchronous and asynchronous bit bang mode RD# strobe output.

Table 3.9 CBUS Configuration Control

* PWREN# must be used with a 10k Ω resistor pull up.

**When in USB suspend mode the outputs clocks are also suspended.

Devices Characteristics and Ratings

Absolute Maximum Ratings

The absolute maximum ratings for the FT232R devices are as follows. These are in accordance with the Absolute Maximum Rating System (IEC 60134). Exceeding these may cause permanent damage to the device.

Parameter	Value	Units
Storage Temperature	-65 to 150	°C
Floor Life (Out of Bag) At Factory Ambient (30°C / 60% Relative Humidity)	168 (IPC/JEDEC J-STD-033A MSL Level 3 Compliant)*	Hours
Ambient Temperature (Power Applied)	-40 to 85	°C
MTTF FT232RL	11162037	hours
MTTF FT232RQ	4464815	hours
VCC Supply Voltage	-0.5 to +6.00	V
DC Input Voltage – USBDP and USBDM	-0.5 to +3.8	V
DC Input Voltage – High Impedance Bidirectional	-0.5 to + (VCC +0.5)	V
DC Input Voltage – All Other Inputs	-0.5 to + (VCC +0.5)	V
DC Output Current – Outputs	24	mA
DC Output Current – Low Impedance Bidirectional	24	mA
Power Dissipation (VCC = 5.25V)	500	mW

Table 5.1 Absolute Maximum Ratings

* If devices are stored out of the packaging beyond this time limit the devices should be baked before use. The devices should be ramped up to a temperature of +125°C and baked for up to 17 hours.

DC Characteristics

DC Characteristics (Ambient Temperature = -40°C to +85°C)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
VCC1	VCC Operating Supply Voltage	4.0	---	5.25	V	Using Internal Oscillator
VCC1	VCC Operating Supply Voltage	3.3	---	5.25	V	Using External Crystal
VCC2	VCCIO Operating Supply Voltage	1.8	---	5.25	V	
Icc1	Operating Supply Current	---	15	---	mA	Normal Operation
Icc2	Operating Supply Current	50	70	100	μA	USB Suspend
3V3	3.3v regulator output	3.0	3.3	3.6	V	

Table 5.2 Operating Voltage and Current

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.3 UART and CBUS I/O Pin Characteristics (VCCIO = +5.0V, Standard Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.2	2.7	3.2	V	I source = 1mA
Vol	Output Voltage Low	0.3	0.4	0.5	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.4 UART and CBUS I/O Pin Characteristics (VCCIO = +3.3V, Standard Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.1	2.6	2.8	V	I source = 1mA
Vol	Output Voltage Low	0.3	0.4	0.5	V	I sink = 2mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.5 UART and CBUS I/O Pin Characteristics (VCCIO = +2.8V, Standard Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	1.32	1.62	1.8	V	I source = 0.2mA
Vol	Output Voltage Low	0.06	0.1	0.18	V	I sink = 0.5mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.6 UART and CBUS I/O Pin Characteristics (VCCIO = +1.8V, Standard Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	3.2	4.1	4.9	V	I source = 6mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 6mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.7 UART and CBUS I/O Pin Characteristics (VCCIO = +5.0V, High Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.2	2.8	3.2	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**

VHys	Input Switching Hysteresis	20	25	30	mV	**
------	----------------------------	----	----	----	----	----

Table 5.8 UART and CBUS I/O Pin Characteristics (VCCIO = +3.3V, High Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.1	2.6	2.8	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.9 UART and CBUS I/O Pin Characteristics (VCCIO = +2.8V, High Drive Level)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	1.35	1.67	1.8	V	I source = 0.4mA
Vol	Output Voltage Low	0.12	0.18	0.35	V	I sink = 3mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	**
VHys	Input Switching Hysteresis	20	25	30	mV	**

Table 5.10 UART and CBUS I/O Pin Characteristics (VCCIO = +1.8V, High Drive Level) **

Only input pins have an internal 200KΩ pull-up resistor to VCCIO

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Vin	Input Switching Threshold	1.3	1.6	1.9	V	
VHys	Input Switching Hysteresis	50	55	60	mV	

Table 5.11 RESET# and TEST Pin Characteristics

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
UVoh	I/O Pins Static Output (High)	2.8		3.6	V	RI = 1.5kΩ to 3V3OUT (D+) RI = 15KΩ to GND (D-)

UVol	I/O Pins Static Output (Low)	0		0.3	V	RI = 1.5k Ω to 3V3OUT (D+) RI = 15k Ω to GND (D-)
UVse	Single Ended Rx Threshold	0.8		2.0	V	
UCom	Differential Common Mode	0.8		2.5	V	
UVDif	Differential Input Sensitivity	0.2			V	
UDrvZ	Driver Output Impedance	26	29	44	Ohms	See Note 1

Table 5.12 USB I/O Pin (USBDP, USBDM) Characteristics

EEPROM Reliability Characteristics

The internal 1024 Bit EEPROM has the following reliability characteristics:

Parameter	Value	Units
Data Retention	10	Years
Write	10,000	Cycles
Read	Unlimited	Cycles

Table 5.13 EEPROM Characteristics

Internal Clock Characteristics

The internal Clock Oscillator has the following characteristics:

Parameter	Value			Unit
	Minimum	Typical	Maximum	
Frequency of Operation (see Note 1)	11.98	12.00	12.02	MHz
Clock Period	83.19	83.33	83.47	ns
Duty Cycle	45	50	55	%

Table 5.14 Internal Clock Characteristics

Note 1: Equivalent to +/-1667ppm

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
Voh	Output Voltage High	2.1	2.8	3.2	V	I source = 3mA
Vol	Output Voltage Low	0.3	0.4	0.6	V	I sink = 8mA
Vin	Input Switching Threshold	1.0	1.2	1.5	V	

Table 5.15 OSCI, OSCO Pin Characteristics – see Note 1

Note1: When supplied, the FT232R is configured to use its internal clock oscillator. These characteristics only apply when an external oscillator or crystal is used.

Thermal Characteristics

The FT232RL package has the following thermal characteristics:

Parameter	Value	Units	Conditions
Theta JA (Θ_{JA})	55.82	°C/W	Still air
Theta JC (Θ_{JC})	24.04	°C/W	

Table 5.16 FT232RL Thermal Characteristics

The FT232RQ package has the following thermal characteristics:

Parameter	Value	Units	Conditions
Theta JA (Θ_{JA})	31.49	°C/W	Still air, center pad soldered to PCB, 9 vias to another plane
Theta JA (Θ_{JA})	62.31	°C/W	Still air, center pad unsoldered
Theta JC (Θ_{JC})		°C/W	

Table 5.17 FT232RQ Thermal Characteristics

USB Power Configurations

The following sections illustrate possible USB power configurations for the FT232R. The illustrations have omitted pin numbers for ease of understanding since the pins differ between the FT232RL and FT232RQ package options.

All USB power configurations illustrated apply to both package options for the FT232R device. Please refer to Section 3 for the package option pin-out and signal descriptions.

USB Bus Powered Configuration

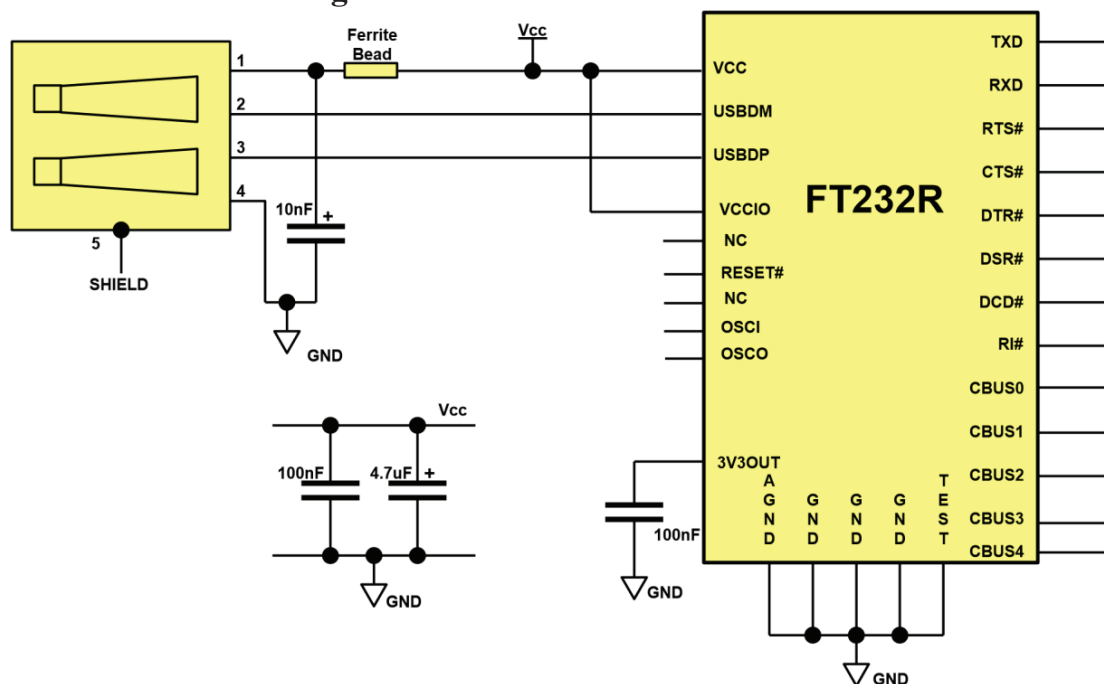


Figure 6.1 Bus Powered Configuration

Figure 6.11 illustrates the FT232R in a typical USB bus powered design configuration. A USB bus powered device gets its power from the USB bus. Basic rules for USB bus power devices are as follows –

- i) On plug-in to USB, the device should draw no more current than 100mA.
- ii) In USB Suspend mode the device should draw no more than 2.5mA.
- iii) A bus powered high power USB device (one that draws more than 100mA) should use one of the CBUS pins configured as PWREN# and use it to keep the current below 100mA on plug-in and 2.5mA on USB suspend.
- iv) A device that consumes more than 100mA cannot be plugged into a USB bus powered hub.
- v) No device can draw more than 500mA from the USB bus.

The power descriptors in the internal EEPROM of the FT232R should be programmed to match the current drawn by the device.

A ferrite bead is connected in series with the USB power supply to reduce EMI noise from the FT232R and associated circuitry being radiated down the USB cable to the USB host. The value of the Ferrite Bead depends on the total current drawn by the application. A suitable range of Ferrite Beads is available from Steward (www.steward.com), for example Steward Part # MI0805K400R-10.

Note: If using PWREN# (available using the CBUS) the pin should be pulled to VCCIO using a 10kΩ resistor.

1.5k Ω resistor is not pulled up to any power supply (hub or host is powered down), so no current flows down USBDP via the 1.5k Ω pull-up resistor. Failure to do this may cause some USB host or hub controllers to power up erratically.

Figure 6.22 illustrates a self-powered design which has a +4V to +5.25V supply.

Note:

1. When the FT232R is in reset, the UART interface I/O pins are tri-stated. Input pins have internal 200k Ω pull-up resistors to VCCIO, so they will gently pull high unless driven by some external logic.
2. When using internal FT232R oscillator the VCC supply voltage range must be +4.0V to 5.25V.
3. When using external oscillator the VCC supply voltage range must be +3.3V to 5.25V. Any design which interfaces to +3.3V or +1.8V would be having a +3.3V or +1.8V supply to VCCIO.

USB Bus Powered with Power Switching Configuration

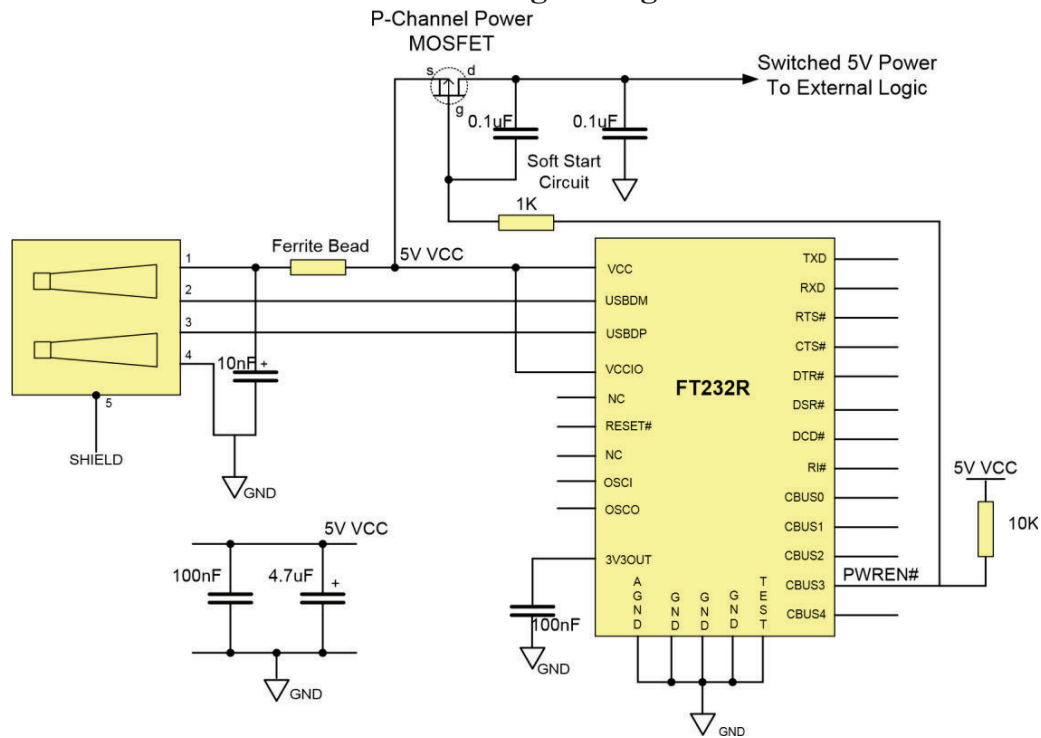


Figure 6.3 Bus Powered with Power Switching Configuration

A requirement of USB bus powered applications, is when in USB suspend mode, the application draws a total current of less than 2.5mA. This requirement includes external logic. Some external logic has the ability to power itself down into a low current state by monitoring the PWREN#

signal. For external logic that cannot power itself down in this way, the FT232R provides a simple but effective method of turning off power during the USB suspend mode.

Figure 6.33 shows an example of using a discrete P-Channel MOSFET to control the power to external logic. A suitable device to do this is an International Rectifier (www.irf.com) IRLML6402, or equivalent. It is recommended that a “soft start” circuit consisting of a 1k Ω series resistor and a 0.1 μ F capacitor is used to limit the current surge when the MOSFET turns on. Without the soft start circuit it is possible that the transient power surge, caused when the MOSFET switches on, will reset the FT232R or the USB host/hub controller. The soft start circuit example shown in Figure 6.33 powers up with a slew rate of approximately 12.5V/ms. Thus supply voltage to external logic transitions from GND to +5V in approximately 400 microseconds.

As an alternative to the MOSFET, a dedicated power switch IC with inbuilt “soft-start” can be used. A suitable power switch IC for such an application is the Micrel (www.micrel.com) MIC2025-2BM or equivalent.

With power switching controlled designs the following should be noted:

- i) The external logic to which the power is being switched should have its own reset circuitry to automatically reset the logic when power is re-applied when moving out of suspend mode.
- ii) Set the Pull-down on Suspend option in the internal FT232R EEPROM.
- iii) One of the CBUS Pins should be configured as PWREN# in the internal FT232R EEPROM, and used to switch the power supply to the external circuitry. This should be pulled high through a 10 k Ω resistor.
- iv) For USB high-power bus powered applications (one that consumes greater than 100mA, and up to 500mA of current from the USB bus), the power consumption of the application must be set in the Max Power field in the internal FT232R EEPROM. A high-power bus powered application uses the descriptor in the internal FT232R EEPROM to inform the system of its power requirements.
- v) PWREN# gets its VCC from VCCIO. For designs using 3V3 logic, ensure VCCIO is not powered down using the external logic. In this case use the +3V3OUT.

USB Bus Powered with Selectable External Logic Supply

3.3V or 5V

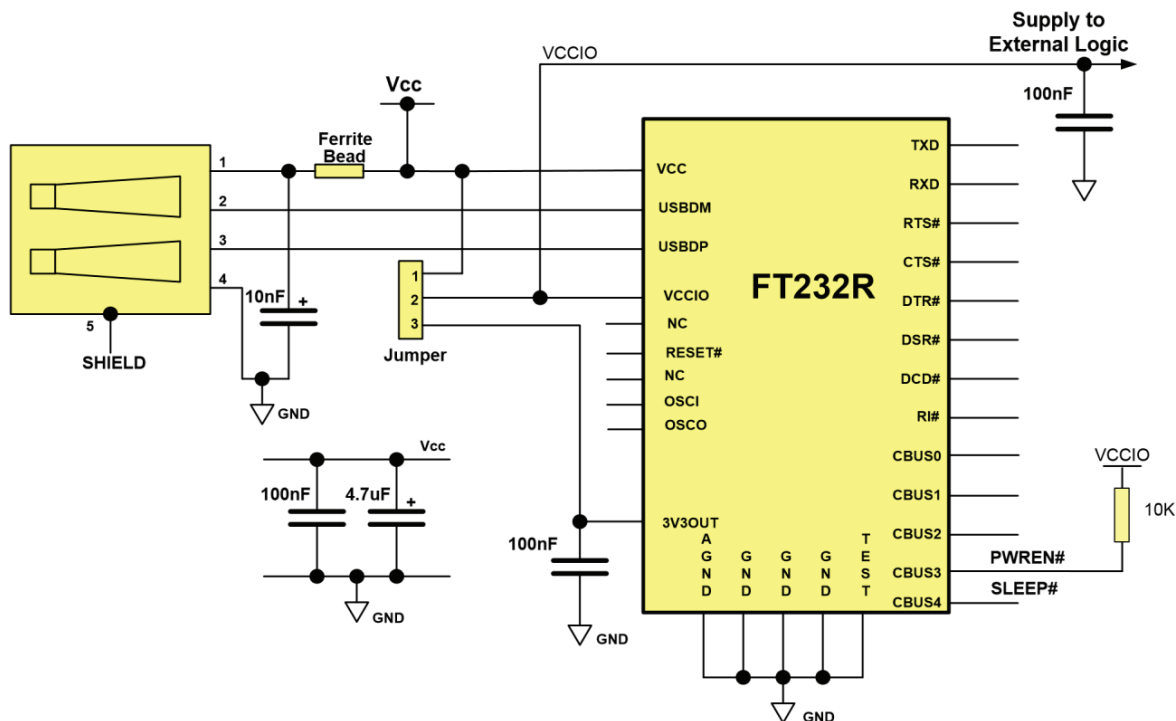


Figure 6.4 USB Bus Powered with +3.3V or +5V External Logic Power Supply

Figure 6.44 illustrates a USB bus power application with selectable external logic supply. The external logic can be selected between +3.3V and +5V using the jumper switch. This jumper is used to allow the FT232R to be interfaced with a +3.3V or +5V logic devices. The VCCIO pin is either supplied with +5V from the USB bus (jumper pins 1 and 2 connected), or from the +3.3V output from the FT232R 3V3OUT pin (jumper pins 2 and 3 connected). The supply to VCCIO is also used to supply external logic.

With bus powered applications, the following should be noted:

- i) To comply with the 2.5mA current supply limit during USB suspend mode, PWREN# or SLEEP# signals should be used to power down external logic in this mode. If this is not possible, use the configuration shown in Section 6.3.
- ii) The maximum current sourced from the USB bus during normal operation should not exceed 100mA, otherwise a bus powered design with power switching (Section 6.3) should be used.

Another possible configuration could use a discrete low dropout (LDO) regulator which is supplied by the 5V on the USB bus to supply between +1.8V and +2.8V to the VCCIO pin and to the external logic. In this case VCC would be supplied with the +5V from the USB bus and the VCCIO would be supplied from the output of the LDO regulator. This results in the FT232R I/O pins driving out at between +1.8V and +2.8V logic levels.

For a USB bus powered application, it is important to consider the following when selecting the regulator:

- i) The regulator must be capable of sustaining its output voltage with an input voltage of +4.35V. A Low Drop Out (LDO) regulator should be selected.
- ii) The quiescent current of the regulator must be low enough to meet the total current requirement of $\leq 2.5\text{mA}$ during USB suspend mode.

A suitable series of LDO regulators that meets these requirements is the MicroChip/Telecom (www.microchip.com) TC55 series of devices. These devices can supply up to 250mA current and have a quiescent current of under $1\mu\text{A}$.