



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

DISEÑO DE UN SISTEMA ELECTRÓNICO DE ENVÍO Y
RECEPCIÓN DE DATOS BASADO EN UN PROTOCOLO
PROPIO

T E S I S

Que para obtener el Título de:

INGENIERO MECÁNICO ELECTRICISTA

Presenta

ALEXIS SAMPEDRO PINTO

Asesor: M. en I. Fernando Macedo Chagolla



México 2010



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

Agradezco a mis pilares familiares: Manuel Sampedro Juárez y Haydee Luvia Pinto Araujo, que sin el apoyo incondicional de padres, este trabajo no estaría en las manos de usted apreciable lector.

Agradezco a la UNAM: PREPA 2 y FES ARAGÓN, que por medio de sus instituciones me formaron para ser un hombre con principios y valores, además de cultivar en mí el conocimiento y la experiencia para este hermoso camino llamado vida.

Agradezco al M. en I. Fernando Macedo Chagolla, que como siempre lo he dicho y lo diré “mas que un profesor y guía, es un gran amigo”.

Proyecto de Investigación para el desarrollo de la tesis.

Objetivos de la investigación

- Plantear un proyecto electrónico (una comunicación serie en alterna para medianas distancias) usando bases electrónicas y de control aprendidas durante la carrera Ingeniero Mecánico Electricista, Eléctrico Electrónico en la especialidad de Sistemas Digitales.
- Analizar el planteamiento del proyecto anterior para realizar una propuesta de desarrollo.
- Desarrollar la fundamentación teórica del proyecto anterior con las características del análisis obtenido para dicho proyecto.
- Desarrollar el prototipo, de la propuesta teórica, del proyecto planteado.

Preguntas de investigación

- ¿Qué tipo de dispositivos se pueden usar para la adaptación de una comunicación de mediana distancia?
- ¿Por qué se plantea el proyecto en corriente alterna?
- ¿Qué voltaje se utilizaría para la transmisión de la comunicación que se plantea?
- ¿Existe algún dispositivo electrónico como adaptador de una comunicación serie de mediana distancia?
- ¿Cuál sería la instalación del dispositivo?
- ¿Para que tipo de sistemas se está diseñado el proyecto?
- ¿Qué tipo de ventajas tiene el proyecto para el sistema donde se utilizará?
- ¿Qué tipo de inconvenientes existen en el proyecto para el sistema donde se utilizará?
- ¿Cuál es la viabilidad del proyecto?

Justificación de la investigación

- ✓ En cualquier diseño electrónico siempre se deben tener en cuenta diversos factores, uno de ellos y de gran importancia es la distancia en la cual se va a comunicar el dispositivo con otro (si se habla en la época actual, la transmisión de la información ha llegado a un nivel inimaginable, se puede transmitir datos desde satélites en otros planetas dentro de nuestro sistema solar y hacerla llegar hasta satélites que viajan en nuestra órbita, el proyecto que se plantea es desarrollar un sistema electrónico para establecer una comunicación bidireccional en corriente alterna para circuitos electrónicos de corriente directa, con la finalidad de aplicar los conocimientos de

electrónica aprendidos en carrera Ingeniero Mecánico Electricista (IME) desarrollando un proyecto electrónico completo (propuesta, análisis, desarrollo y prototipo).

Criterios para evaluar la importancia potencial de una investigación

- ✓ **Conveniencia.** El proyecto a desarrollar tiene como finalidad acoplar un sistema electrónico de CD a un sistema de comunicación en corriente alterna para poder enviar la información a una distancia mediana, es decir, sirve como un sistema acoplador electrónico para la transmisión de datos.
- ✓ **Relevancia social.** El proyecto puede emplearse en cualquier diseño electrónico con características similares que necesite enviar información a dispositivos periféricos que se encuentren a mediana distancia, por ello su desarrollo es en forma modular y de tal manera modificar algunas etapas para su adaptación, en un caso particular para la aplicación son las casas inteligentes (DOMÓTICA).
- ✓ **Implicaciones prácticas.** Los sistemas que transmiten información son muy diversos, pero casi todos son dispositivos dependientes de un sistema PC, el proyecto planteado realiza una comunicación independiente entre dispositivos, es decir, es un dispositivo electrónico periférico auxiliar para transmitir información siendo independiente a un sistema de control complejo.
- ✓ **Valor teórico.** El desarrollo de opciones para realizar una comunicación entre dispositivos siempre es aceptable en un mundo donde el desarrollo de sistemas de control es una necesidad de la vida cotidiana.

El proyecto planteado tiene como objetivo establecer la comunicación a mediana distancia, dejando una gran cantidad de opciones para ampliar el alcance del mismo, algunas propuestas son:

- La velocidad de transferencia, la limitante de dispositivos de recepción son contemplados como una propuesta de desarrollo a futuro para el mejoramiento del sistema que se diseña en este proyecto.
- El proyecto solo realiza una comunicación unidireccional, se propone realizar un estudio más amplio para una comunicación bidireccional.
- La elaboración de diversos dispositivos electrónicos esclavos adaptados a este protocolo de comunicación.

Viabilidad de la investigación

- ✓ Factibilidad de la realización de un estudio en cuando a los recursos disponibles. El proyecto se puede realizar satisfactoriamente ya que se diseña con los dispositivos electrónicos que se encuentran en el mercado, para evitar el desarrollo de nuevas investigaciones y reducir costos.

Fuentes de información

- **Diseño Lógico., Antonio Lloris – Alberto Prieto** (En la referencia se encuentra todo con respecto a los sistemas digitales para la manipulación de señales).
- **Microcontrolador PIC16F84 Desarrollo de proyectos., Enrique Palacios Municio, Fernando Remiro Domínguez, Lucas J. López Pérez** (En la referencia bibliográfica se utilizan diseños de proyectos electrónicos manipulados con microcontroladores para la desarrollo del proyecto electrónico a elaborar).
- **Microcontroladores PIC, Diseño práctico de aplicaciones, José Ma. Angulo Usategui, Ignacio Angulo Martínez** (En la referencia citada se tiene como apoyo para el manejo y el control del Microcontrolador PIC 16f84).
- **Metodología de la investigación., Roberto Hernández Sampieri, Carlos Fernández Collado, Pilar Baptista Lucio** (La referencia anterior será indispensable para tener una metodología en el desarrollo del proyecto planteado).
- **Apuntes de Circuitos Digitales** (Los apuntes sirven como apoyo para el análisis electrónico de los sistemas digitales que se usarán en el desarrollo del proyecto).
- **Data Sheet de dispositivos a utilizar** (Las hojas de especificaciones son referencia importante para el control y análisis de los componentes que se utilizan en el desarrollo para el proyecto).
- **PROTEUS** (Software para el diseño y creación de sistemas de circuitos electrónicos).

TEMARIO

➤ CAPÍTULO. I.- <u>Especificaciones del sistema</u>	10
I.1.- Introducción	10
I.2.- Especificaciones del proyecto para la transmisión general de datos	10
I.3.- Elaboración de un protocolo propio para el sistema de comunicación planteado	16
I.4.- Elaboración de estrategias de desarrollo para el análisis electrónico y fabricación de prototipos	21
➤ CAPÍTULO. II.- <u>Diseño del dispositivo electrónico maestro</u>	25
II.1.- Introducción	25
II.2.- Adquisición de datos	25
II.3.- Sincronización	56
II.4.- Interfaz de envío	59
II.5.- Codificación de señal	64
➤ CAPÍTULO. III.- <u>Diseño del dispositivo electrónico esclavo</u>	70
III.1.- Introducción	70
III.2.- Interfaz de recepción	70
III.3.- Sincronización	73
III.4.- Direccionamiento	74
III.5.- Decodificación de señal	76

➤ <u>CAPÍTULO. IV.- Construcción y pruebas del sistema</u>	81
IV.1.- Introducción	81
IV.2.-Elaboración del prototipo maestro	81
IV.3.-Elaboración del prototipo esclavo	88
<u>Conclusiones</u>	91
<u>Anexo</u>	92

CAPÍTULO. I.- ESPECIFICACIONES DEL SISTEMA

I.1.- Introducción

El proyecto tiene la función principal de realizar el desarrollo completo de una propuesta de ingeniería eléctrica-electrónica y de tal manera romper con el paradigma existente de la unión entre teoría y práctica, es por ello que se aplica, para este proyecto, el conocimiento adquirido durante la carrera de IME y así aportar una visión de aplicación y llevarla hasta el desarrollo de un prototipo, es por ello que, para la elaboración de este proyecto se hace uso de la mayor variedad posible de dispositivos electrónicos para entender su funcionamiento y una aplicación concreta en la vida cotidiana.

Por lo anterior y de acuerdo a las preguntas realizadas en el *Proyecto de Investigación para el desarrollo de la tesis*, se comienza por delimitar las características del sistema que se plantea en ésta, para así dar forma y un seguimiento congruente en el desarrollo del proyecto. En este capítulo se manejan tres aspectos principales para el inicio del proyecto; primero, las *Especificaciones del proyecto para la transmisión general de datos*, en esta parte del capítulo se delimitan las características de comunicación que el sistema tendrá; segundo, la *Elaboración de un protocolo propio para el sistema de comunicación planteado*, es aquí donde se establecen las características de comunicación que el sistema tendrá mediante un protocolo propuesto para este fin; y por último, la *Elaboración de estrategias de desarrollo para el análisis electrónico y fabricación de prototipos*, en esta última parte del capítulo se realiza el seguimiento por el cual se desarrolla el proyecto de una forma sistémica y secuencial para el desarrollo electrónico y así tener, en capítulos posteriores, el desarrollo y la elaboración de prototipos de forma coherente y lógica.

I.2.- Especificaciones del proyecto para la transmisión general de datos

En este capítulo se describe el tipo de comunicación que el sistema realiza (comunicación unidireccional, envío y recepción de datos) y se enfoca, como ejemplo de aplicación, a la Domótica y tener así un claro ejemplo del alcance del proyecto, así mismo dentro del este capítulo y de acuerdo a los parámetros para dicho campo de prueba, se toman las características que limitan el sistema.

El sistema de comunicación que puede realizar el proyecto se describe a continuación:

La Comunicación Unidireccional. Es cuando el dispositivo Transmisor (Tx) tiene la capacidad de transmitir datos a uno o varios dispositivos Receptores (Rx), la forma de funcionamiento es la siguiente: El dispositivo Tx manda la o las direcciones a los dispositivos Rx que desea transmitir información, posteriormente transmite el dato, hay que aclarar que el dato a transmitir es el mismo para todas las direcciones seleccionadas anteriormente, ejemplificando en el campo de la Domótica, si un usuario desea apagar todo el alumbrado de una habitación, o tal vez, desea poner una intensidad luminosa fija a todos los dispositivos Rx que controlan el alumbrado de la habitación, se transmiten las direcciones de todos los dispositivos a controlar y posteriormente la actividad a realizar.

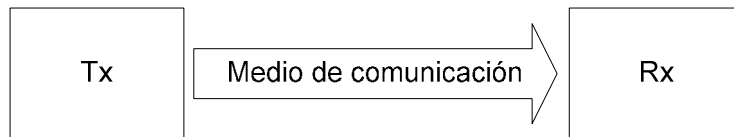


Fig. I.2.1 Diagrama a bloques de un sistema de comunicación unidireccional.

El proyecto tiene la finalidad, de diseñar dispositivos electrónicos básicos que cumplan con la función del tipo de comunicación planteado anteriormente, sin embargo en el Anexo se describen dos tipos de comunicación más que los dispositivos maestros y esclavos podrían realizar como propuesta de desarrollo posterior, para el mejoramiento de este proyecto de tesis (La Comunicación Bidireccional Tipo Básica y La Comunicación Bidireccional Tipo Compleja) Ver Anexo.

Los dispositivos electrónicos básicos que se desarrollan en este proyecto, se acoplaran a otros dispositivos electrónicos periféricos, los cuales tendrán una intercomunicación mediante el sistema de comunicación diseñado, para ello se comienza por determinar a los dispositivos periféricos que controlan y que son controlados en el sistema de comunicación, estos dispositivos se nombran de la siguiente manera:

Todos los dispositivos que controlan se denominan → Maestros

Todos los dispositivos que son controlados se denominan → Esclavos

Una vez identificados los dos dispositivos principales (Maestro y Esclavo), se determinan los tipos de comunicación que existen entre los ellos, que son:

Maestro → Esclavo Comunicación Unidireccional

Maestro ← → Esclavo BB Comunicación Bidireccional Tipo Básica.
(Comunicación propuesta como mejoramiento de este proyecto, ver Anexo)

Maestro ← → Maestro Comunicación Bidireccional Tipo Compleja.
(Comunicación propuesta como mejoramiento de este proyecto, ver Anexo)

El sistema de comunicación que se realiza en este proyecto, se diseña para poder transmitir a una distancia mínima entre dispositivos de 100 metros, para ello en la transmisión de datos se utiliza con una señal alterna.

En un sistema de comunicación, una de las características más importantes a considerar, es el número de hilos que se necesitan para realizar el intercambio de datos, además, el costo monetario también tiene un peso considerable para la viabilidad de un proyecto. Es por ello que al considerar esas dos premisas primordiales que son: número de hilos y costos, se hace esta comunicación de datos con una señal alterna de 110VAC, debido a que dicha señal alterna es la que se maneja en una casa habitación, es importante agregar que la caída de tensión en largas distancias con este tipo de señales es inapreciable, por su característica de señal senoidal, que tiene sólo una componente de frecuencia y que la señal se maneja a baja frecuencia (60 Hz), Ver Fig. 1.2.2.

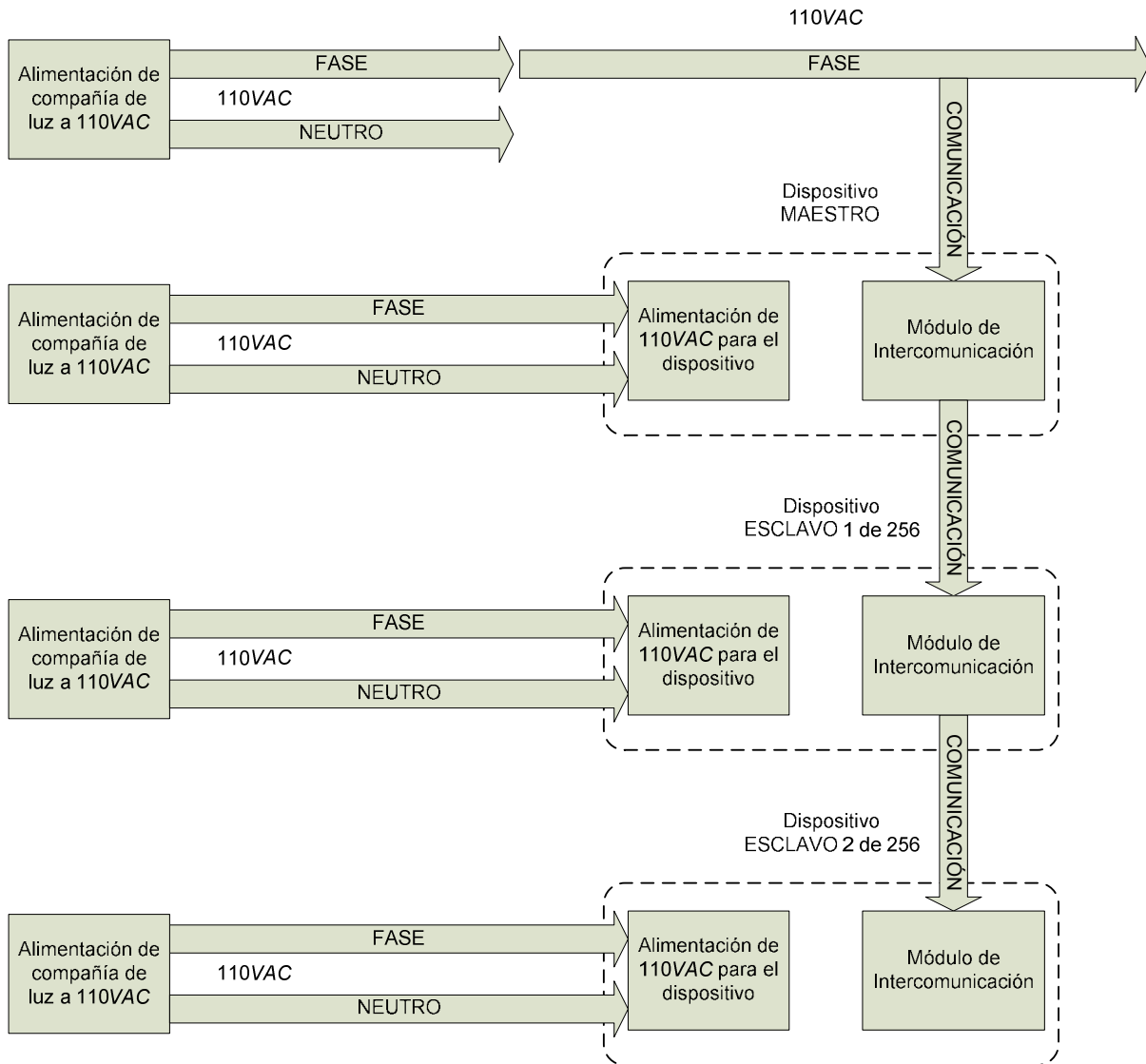


Fig. I.2.2 Es necesario un solo hilo para una comunicación unidireccional, la referencia (para cerrar el circuito) se toma del lugar donde se conecta la alimentación del dispositivo esclavo.

Un sistema de comunicación simple, es la que se realiza de un punto a otro sin posibilidad de respuesta, es decir, de una sola vía ó una comunicación unidireccional.

El sistema completo de comunicación que se desarrolla en el presente trabajo, consta de diversas etapas, cada una de ellas tiene un propósito específico, y están desglosadas detalladamente en los capítulos posteriores. Por el momento se da un panorama general del cómo se realiza la comunicación dividiendo en módulos dichas etapas.

El sistema consta de tres módulos fundamentales e independientes, el dispositivo MAESTRO, el BUS DE COMUNICACIÓN y el dispositivo ESCLAVO, el primero tiene la función de controlar todo o parte de un proceso, toma decisiones e interactúa entre el sistema electrónico y el usuario, por otro lado, el segundo establece una relación entre cualquier dispositivo periférico y por último, el tercero, que recibe la información ya procesada y lleva dicha información a uno o varios actuadores que realizan el proceso para un sistema de control.

La Fig. 1.2.3 Muestra la aplicación del sistema de comunicación planteado en el campo de la Domótica, donde un sistema de control principal puede distribuir la información a otros dispositivos Maestros ó Esclavos, hay que tener presente que todo el sistema de comunicación no excede un número total de 3 hilos para la elaboración de la red de comunicación y alimentación eléctrica (el número de dispositivos a conectar en un sistema depende de la norma especificada para el amperaje del cable usado en la alimentación)

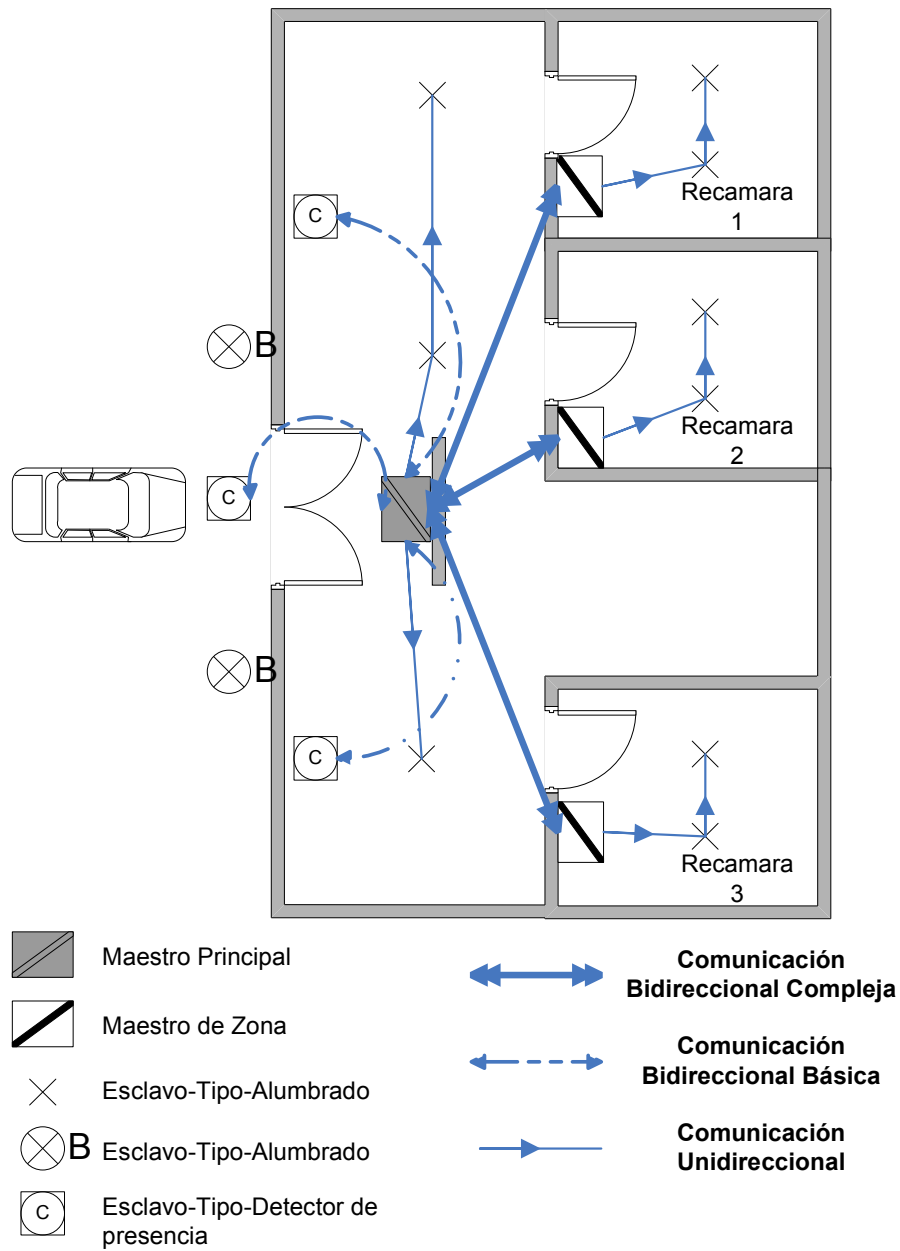


Fig. I.2.3 Diagrama gráfico que muestra la conexión de dispositivos y los tipos de comunicación que existe entre ellos.

I.3.- Elaboración de un protocolo propio para el sistema de comunicación planteado

Otro aspecto que define el diseño de los dispositivos que componen el sistema, es la manera de comunicarse entre ellos, para lo cual se desarrolla un protocolo de comunicación propio con el objetivo de hacer una propuesta de diseño y elaboración de nuevas formas de comunicación propuestas por el desarrollador de prototipos y así promover la creación de tecnologías alternativas.

Se elabora un protocolo propio con la finalidad de hacer el proceso completo de manejo de la información de las señales obtenidas, para ello en este tema se realiza el análisis de las señales obtenidas, se acopla a una señal que transporta la información y se elabora un protocolo propio para la codificación y decodificación de la misma.

Se comienza con el estudio de la señal que se adoptará como transmisora de datos, esta es una señal alterna de 110V (Ver Fig. I.3.1) y la forma en la cual se transmiten los datos son de la siguiente manera:

Primero se analiza la señal de 110VAC, para poder determinar las características con las que la comunicación se va a transmitir.

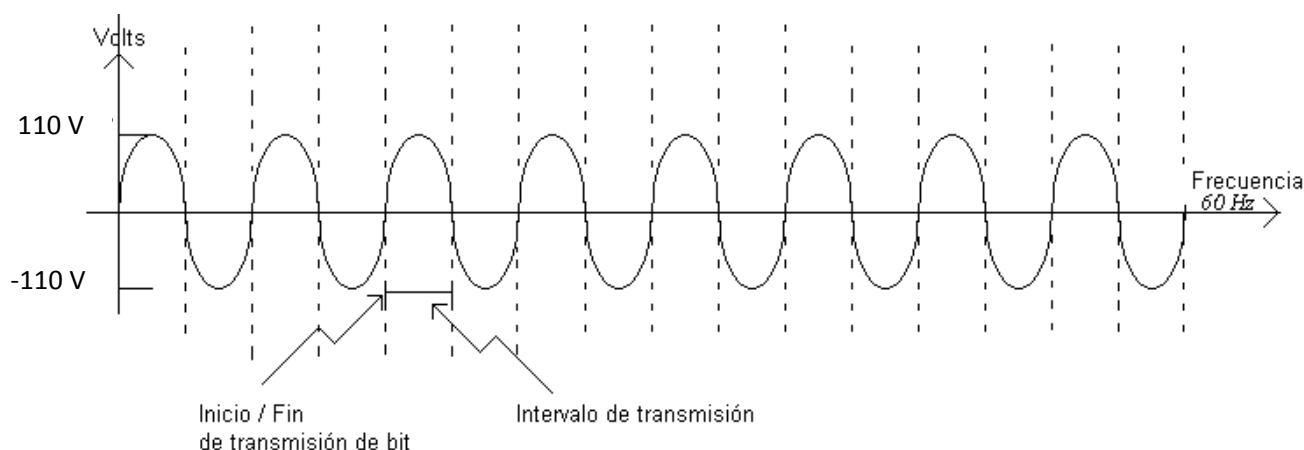


Fig. I.3.1 Muestra una señal alterna de 110V y 60Hz, acotada en sus semiciclos, para poder realizar el estudio de la transmisión de datos.

La señal alterna tiene una frecuencia de 60HZ, es decir, 60 ciclos/s, y éstos a su vez dan un total de 120 semiciclos/s, este semiciclo es el que se toma como referencia para poder transmitir uno de los bits que componen al dato y estos últimos (agrupados) componen la información a transmitir.

La Fig. I.3.2, es un ejemplo de como el sistema transmite una serie de 8 bits y el dato a transmitir es **b'10110001'**, cada semiciclo es un bit que se transmite y el cruce por cero de la señal indica el inicio y fin del mismo, es decir, el dato a transmitir utilizará 8 semiciclos de la señal alterna, cuando se requiera que la señal envíe un "1" lógico, se deja pasar el semiciclo en el que se encuentre la señal (puede ser +110V ó -110V) , de lo contrario si se desea mandar un "0" lógico no habrá semiciclo es decir, la señal será 0V, hay que tener presente y por eso es necesario recalcar, que en una serie de niveles lógicos en cero, el indicador de inicio y fin esta dado por el sensado externo de la señal en su cruce por cero.

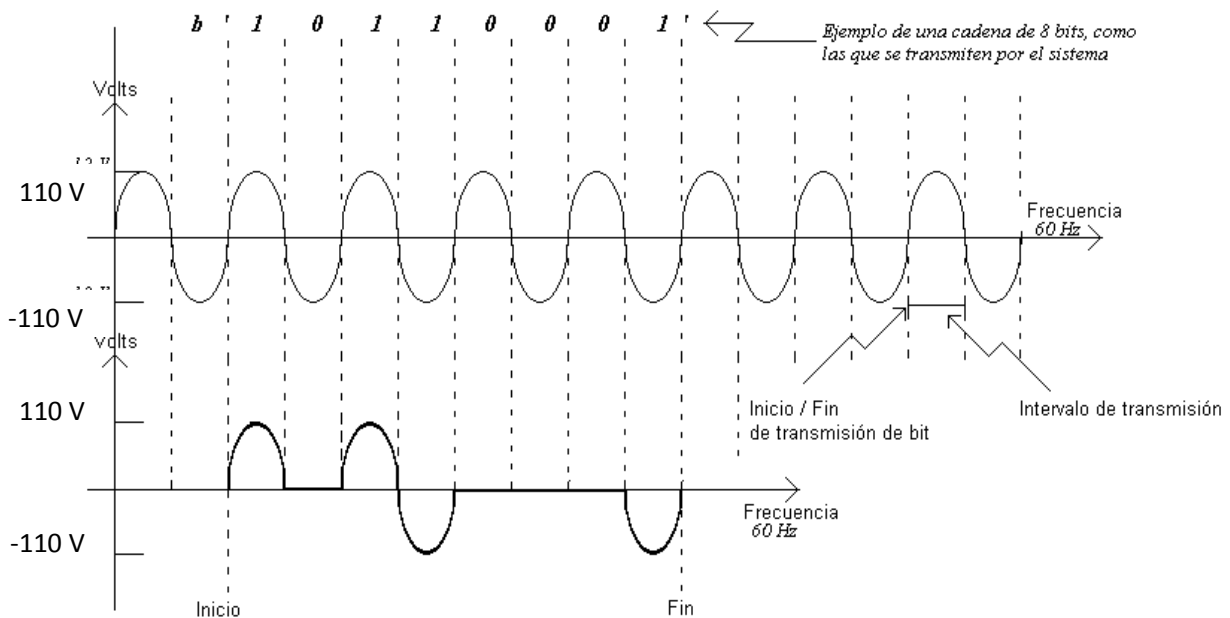


Fig. I.3.2 Muestra la forma de cómo se interpreta una serie de ocho bits en la señal alterna de 110V en que transmite.

Una vez establecida la forma de transmisión de la información, es necesario realizar un protocolo de comunicación, el cual sirve para poder controlar el tráfico de datos que habrá dentro de la comunicación, este protocolo engloba las características básicas que un sistema de comunicación emplea, como lo son: el bit de inicio/fin, de sincronización, el bit de selección de dirección/dato, etc., las cuales se describen a continuación.

Protocolo para una comunicación unidireccional

Uno de los objetivos que se plantea para en el desarrollo de este proyecto, es establecer una comunicación unidireccional en el sistema electrónico, es por ello que se describe el protocolo a seguir para dicha comunicación. También es importante aclarar que el protocolo a seguir es parte de uno de los objetivos de la propuesta para este proyecto, por lo anterior se establece como una parte importante del proyecto la elaboración de un protocolo propio, éste tiene la finalidad de desarrollar en su totalidad el diseño de los dispositivos electrónicos para así promover la creación de nuevas tecnologías, en este caso electrónicas y de diseño.

Primero se describe la simbología que se usa para el protocolo de comunicación propuesto.

Descripción de términos usados

TA → Transferencia Activa

TA	1	Inicia Transferencia
	0	Fin de Transferencia

CT → Código de Transferencia

CT	0	0	Código para dispositivos Esclavo (Unidireccional)
	0	1	Código para dispositivos Esclavo BB (Bidireccional)
	1	*	Código para dispositivos Maestro (Bidireccional)

DD → Dirección ó Dato

DD	1	La serie siguiente de 8 bits es una dirección
	0	La serie siguiente de 8 bits es un dato

Ahora el protocolo a seguir en una comunicación unidireccional es el siguiente, teniendo en cuenta que la señal de transmisión se encuentra en un estado de nivel bajo y ésta la controla el dispositivo que envía la transferencia de datos:

- Siempre se comienza con la activación de la señal de inicio **TA**, ésta se mantiene constante hasta que el dispositivo que inició la transmisión la desactiva poniéndola en un estado de voltaje bajo o "0" lógico que indicara el fin de la transmisión.
- Después de mandar la señal de inicio o poner en "1" lógico a **TA**, se envía el código de transferencia (**CT**), en este caso el **CT** para un dispositivo Esclavo es el "00".

Los dos puntos anteriores son la base para el inicio de la transmisión general de datos, ahora una vez identificando el tipo de dispositivo receptor al que se dirige la transmisión, se envía un protocolo específico para éste, en la comunicación unidireccional es el siguiente:

Tomando como ejemplo se envía a un dispositivo periférico con dirección b'00101000', el dato b'11101010', se debe tener en cuenta que siempre se envía primero la dirección o direcciones y después se envía el dato o datos.

TA	DD	D	I	R	.	8	bits
1	1	0	0	1	0	1	0 0 0

Una vez detectado el bit de sincronización S, se envía el dato

TA	DD	D	A	T	O	.	8	bits
1	0	1	1	1	0	1	0 1 0	

Vale la pena destacar que el bit **TA** siempre se mantiene encendido durante la transmisión, para indicar el fin de la transmisión solo se pone en un estado bajo, es decir **TA** → "0 Lógico".

Ahora se muestra el protocolo de transmisión completo para una comunicación unidireccional, tomando como ejemplo los datos anteriores.

Dispositivo Esclavo con dirección: **b'00101000'**

Dato a transmitir: **b'11101010'**

INICIO			TRANSMISIÓN DE DATOS														FIN						
TA	CT		TA	DD	D I R . B							TA	DD	D A T O B							TA		
					bits									bits									
1	0	0	1	1	0	0	1	0	1	0	0	0	1	0	1	1	1	0	1	0	1	0	0
			<i>Protocolo base para la transmisión a dispositivos esclavos</i>										<i>Protocolo base para la transmisión a dispositivos esclavos</i>										

I.4.- Elaboración de estrategias de desarrollo para el análisis electrónico y fabricación de prototipos

En la parte final de este capítulo, se elabora una estrategia general de funcionamiento para poder identificar el flujo que el sistema necesita para su desarrollo, es decir, con esta estrategia de funcionamiento se plantea identificar de una forma lógica y secuencial el desarrollo que el proyecto necesita para la elaboración de los objetivos y del prototipo, evitando así errores de planeación y de acoplamiento que cualquier proyecto en desarrollo tiene.

Como estrategia de desarrollo, primero se plantea el análisis y desarrollo del dispositivo que “Transmite” la información (Tx), seguido de la primer etapa se desarrolla el dispositivo que “Recibe” la información (Rx) para finalmente realizar la comunicación, que es el objetivo del proyecto, cabe mencionar que se plantea una etapa extra llamada actuador, sin embargo esta no es parte del desarrollo del sistema ya que ésta es solo un dispositivo prueba con fines de aplicación.

Para ello se ha desarrollado el diagrama funcional para la transmisión de datos Ver Fig.I.4.1 y la descripción de las partes que componen el diagrama se muestran en la Tabla I.4.1.

El diagrama se divide en 7 etapas principales, y es la forma en la cual se hace el análisis del proyecto, se describe el funcionamiento electrónico y se realiza el diagrama electrónico real para el prototipo final maestro y el prototipo final esclavo, cabe mencionar que el orden que se le da para la elaboración de cada una de las etapas, es de acuerdo a la dependencia de las mismas, es decir, se hacen, primero las etapas independientes, para después realizar las etapas dependientes de las etapas anteriores y de tal manera tener la herramienta para la elaboración del prototipo, y así sucesivamente hasta realizar la transmisión.

Las etapas que componen al dispositivo maestro son:

➤ **Adquisición de Datos.**

Es la parte del dispositivo maestro que interactúa con el usuario y es la encargada de recibir todos los datos necesarios para la transmisión a los diversos dispositivos (puede ser a un dispositivo esclavo o a otro dispositivo maestro).

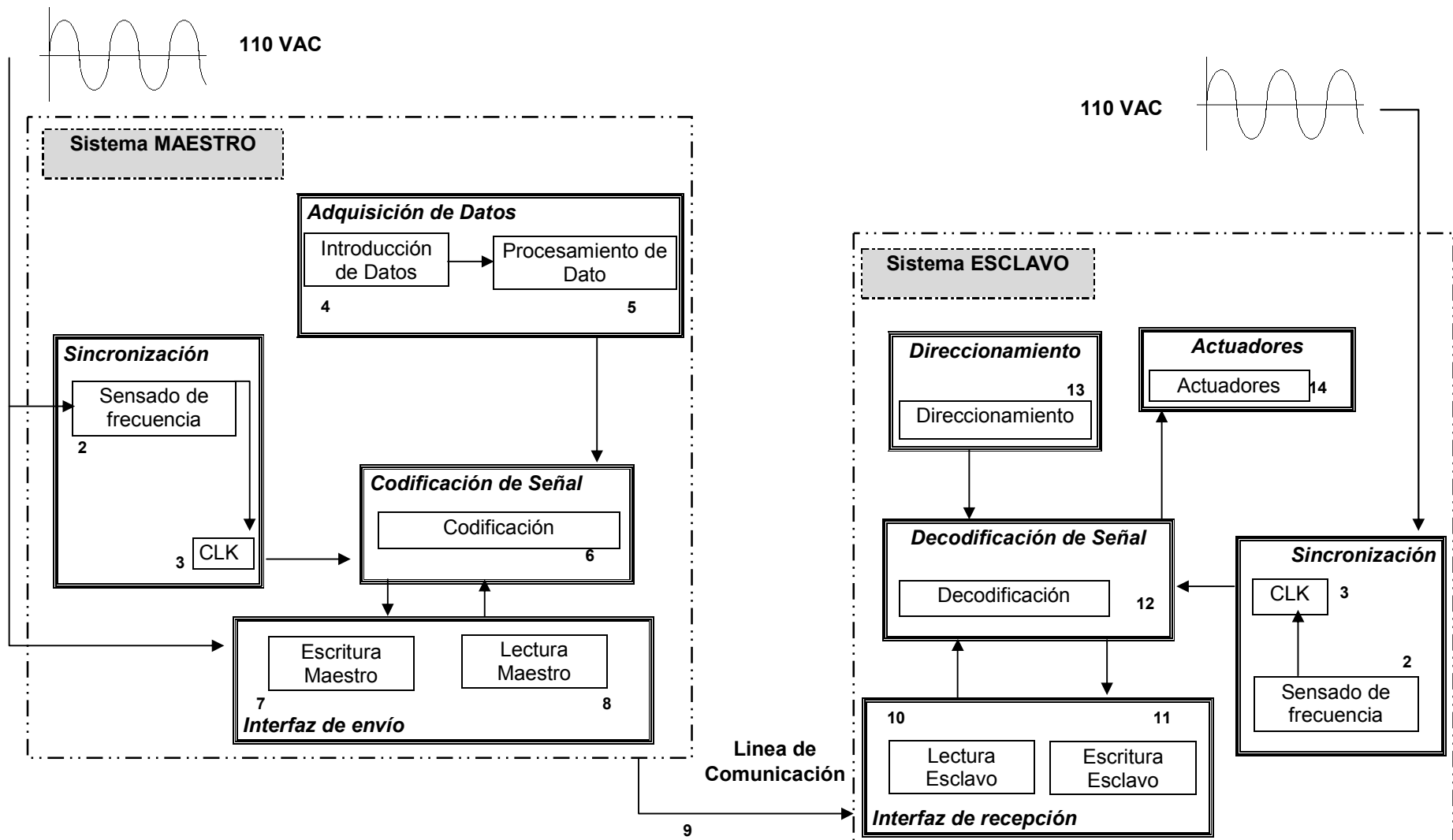


Fig. I.4.1 Diagrama funcional para la transmisión de datos

Tabla I.4.1 Descripción de las partes que componen las etapas del diagrama funcional

1.-	La alimentación es de 110VAC, y es necesario acoplarla a nuestro dispositivo electrónico el cual se alimenta a 5VDC.
2.-	La comunicación se basa a la frecuencia de trabajo (60HZ) de la señal de voltaje y es por ello que se necesita una parte electrónica que haga el sensado, para poder monitorear la frecuencia.
3.-	Etapa que se encuentra dentro de la programación del microcontrolador, y es la que se encarga de sincronizar la frecuencia de trabajo de la señal alterna con la frecuencia de trabajo para el procesamiento de datos.
4.-	La parte de “Introducción de Datos”, es una etapa periférica, ésta dependerá del dispositivo maestro al que se le adapte el sistema de comunicación. Para este caso práctico sólo se necesita un periférico que muestre y capture tanto los datos como las direcciones que se transmiten en el sistema.
5.-	Etapa que se encuentra dentro de la programación del microcontrolador, y es la que se encarga de procesar los datos y direcciones que el sistema periférico entrega para su envío, además de procesar, también, los datos que recibe.
6.-	Etapa que se encuentra dentro de la programación del microcontrolador, y es la que se dedica a sincronizar los datos a transferir con la frecuencia de trabajo para la transmisión.
7.-	Parte del modulo maestro en la cual se acopla la señal de la etapa Codificación de Señal para la transmisión, en ésta etapa se maneja la señal alterna de 12V.
8.-	Es la etapa de acoplamiento entre el dispositivo de control maestro y la línea de transmisión para la recepción de datos de la línea de comunicación.
9.-	Bus de transmisión de datos.
10.-	Es la etapa de acoplamiento entre el dispositivo de control esclavo y la línea de transmisión para la recepción de datos de la línea de comunicación.
11.-	Parte del modulo esclavo en la cual se acopla la señal de la etapa Decodificación de Señal para la transmisión, en ésta etapa se maneja la señal alterna de 12V.
12.-	Etapa que se encuentra en el microcontrolador, y es la etapa del dispositivo esclavo que decodifica la señal obtenida para su interpretación.
13.-	Etapa electrónica para direccionamiento manual del dispositivo periférico esclavo.
14.-	Dispositivo a controlar.

➤ **Sincronización.**

Esta etapa se encuentra tanto en el dispositivo maestro como en el dispositivo esclavo y es la encargada de la sincronización entre dispositivos en el momento de realizar la comunicación.

➤ **Interfaz de envío.**

En esta etapa del proyecto se acoplan las señales del dispositivo maestro para la lectura y escritura de datos con las de la señal de corriente alterna que transmite los datos.

➤ **Codificación de Señal.**

La etapa Codificación de Señal es la encargada de acoplar la interfaz de Adquisición de Datos del dispositivo que transmite con el protocolo de comunicación desarrollado para este sistema.

Las etapas que componen al dispositivo esclavo son (hay que aclarar que la etapa de Sincronización se incluye también en el dispositivo esclavo pero es el mismo desarrollo que en el dispositivo maestro, por lo cual no se repite como parte del diseño):

➤ **Interfaz de recepción.**

Realiza lo contrario a la etapa de Interfaz de envío, al acoplar la señal de corriente alterna que se utiliza para la transmisión de datos con la etapa de Decodificación de Señal para la lectura y escritura del dispositivo esclavo.

➤ **Direccionamiento.**

Es la parte del dispositivo esclavo que sirve como interfaz con el usuario para introducir la dirección fija del dispositivo a controlar.

➤ **Decodificación de Señal.**

Es la encargada de decodificar la señal recibida de corriente directa respetando el protocolo desarrollado.

➤ **Actuador.**

(Esta etapa no se incluye dentro del desarrollo del sistema ya que es independiente a los dispositivos que realizan la comunicación, solo se hace referencia a ella como un ejemplo de aplicación). Dispositivo externo manipulado por la información obtenida a distancia del dispositivo maestro.

CAPÍTULO. II.- DISEÑO DEL DISPOSITIVO ELECTRÓNICO MAESTRO

II.1.- Introducción.

Como parte de los objetivos de este proyecto es llevar los conocimientos adquiridos en las aulas y al plasmarlos en una aplicación de la vida cotidiana, en este capítulo nos adentramos en la relación existente entre el medio que proporciona la información y los dispositivos electrónicos que la manipulan para un fin concreto. El proyecto tiene un *usuario* como medio que proporciona la información a manipular, por lo tanto el dispositivo se desarrolla con características para ese uso, además debe incluir las características señaladas en las especificaciones del proyecto.

Se denomina *dispositivo maestro* al dispositivo que envía la información y que tiene el control principal del sistema, y es este *dispositivo maestro* el que se desarrolla en este capítulo.

II.2.- Adquisición de datos

En este capítulo se desarrollan las partes “introducción de Datos” y “Procesamiento de Dato” (partes 4 y 5 de la etapa Adquisición de Datos respectivamente) que se encuentran indicados en el *Diagrama funcional para la transmisión de datos* de la Fig. I.4.1.

El proyecto se inicia con la elaboración del dispositivo Maestro el cual consta de diversas etapas, *Adquisición de datos, Sincronización, Codificación de Señal e Interfaz de envío*. La etapa que se analiza a continuación es *Adquisición de datos* y tiene la función de introducir diferentes datos a diferentes direcciones de esclavos, ya que como dispositivo maestro el envío de datos debe ser a direcciones únicas para dispositivos esclavos específicos.

Introducción de Datos

Esta parte de la etapa “*Adquisición de datos*” está compuesta por una pantalla LCD con 16 caracteres y 2 líneas de visualización y un teclado hexadecimal, este último tiene la función de recibir la información (dato y dirección) que será enviada a la etapa de *Codificación de Señal*, mientras que

la pantalla LCD es un dispositivo visual en el cual se muestra dicha información antes de ser enviada, ambas partes son dispositivos electrónicos y en este capítulo se analizan y desarrollan esquemas electrónicos de los diversos componentes electrónicos y el funcionamiento de cada uno de ellos, sin embargo solo es a nivel componentes electrónicos ya que ambas se controlan por la parte “Procesamiento de Dato” y es en ésta donde se desarrolla la programación de los dispositivos para su control, como se muestra en la Fig. II.2.1

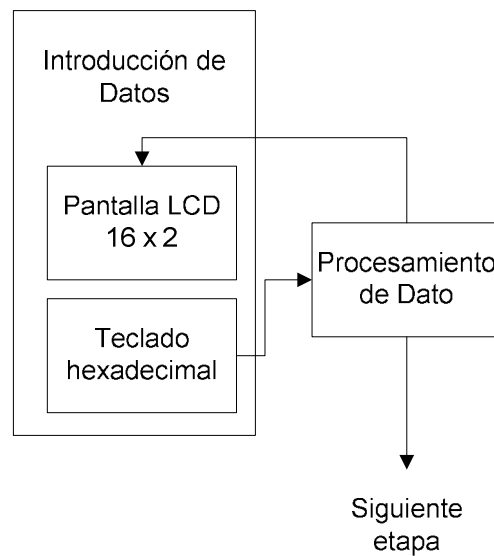


Fig. II.2.1 Diagrama de procesos para la elaboración de la etapa *Adquisición de Datos*

Pantalla LCD de 16 x 2, como dispositivo visual

Un módulo de visualización está basado en un Display de Cristal Líquido (por sus iniciales en ingles LCD), éste puede representar tanto información alfanumérica como símbolos, lo cual facilita la representación de la información que se genera en cualquier proyecto electrónico.

Normalmente los LCD se definen por el número de caracteres por línea y el número de líneas. Para la aplicación del LCD en este proyecto se utiliza un LCD de 16 x 2 (dos líneas de 16 caracteres cada una).

El LCD que se utiliza en este proyecto es el *JHD-162A STNG LEDB*, las características del dispositivo electrónico se muestran en la Tabla II.2.1.

Tabla II.2.1 Características generales del LCD JHD-162A

JHD-162A STNG LEDB	
JHD-162A	Número de Serie
STNG	Luz de respaldo con led
LEDB	Led Azul
Contenido del Display	16 caracteres por 2 líneas
Puntos por carácter	5 x 8
Consumo	7,5mW aprox. (sin luz de respaldo)
Caracteres	-ASCII -Japoneses Kan ji -Griegos -Símbolos matemáticos
Memoria por línea de pantalla	40 caracteres
Conexión	-Bus de 4 bits -Bus de 8 bits

Aunque los LCD son diferentes en cuanto a presentaciones físicas y fabricantes, comparten el mismo juego de instrucciones y de señales control en sus pines (el LCD JHD-162A incluye dos pines más para la alimentación del led de luz de respaldo) las cuales se describen en la Tabla II.2.2.

Tabla II.2.2 Función de los pines del LCD JHD-162A

SEÑAL	DEFINICIÓN	PIN
V_{SS}	Masa	1
V_{CC}	Tensión de alimentación	2
V_{EE}	Tensión para ajustar el contraste	3
RS	Selección de Registro	4
R/W	Lectura / Escritura	5
E	Habilitación del módulo	6
D0...D7	Bus de Datos	7...14
LED+	Alimentación del led	15
LED-	Alimentación del led	16

Descripción de las funciones de los pines del LCD (Display de Cristal Líquido)

V_{SS} y V_{CC} Terminales de alimentación del LCD y es de 5VCD

V_{EE} Terminal que regula el contraste de la pantalla del LCD y se realizará al dividir el voltaje de alimentación de 5VCD con una resistencia de 10KΩ

- RS** Registro de control o registro de datos, si **RS=0** la información presente en el bus de datos es un comando o señal de control del LCD.
- Si **RS=1** la información presente en el bus de datos es un caracter a escribir.
- R/W** Lectura/Escritura. Si **R/W=0** se activa la función escritura en el LCD.
- Si **R/W=1** se lee información del LCD.
- E** Terminal de habilitación del LCD. Si **E=0** el LCD no esta habilitado, por lo tanto no funcionan las señales restantes.
- Si **E=1** el LCD se habilita.
- D0:D7** Bus de datos bidireccional y es a través de estas líneas que se realiza la transferencia de datos y comandos entre el LCD y el sistema que lo gobierna.
- LED+ y LED-** Terminales de alimentación del led de luz de respaldo (se tiene que agregar la resistencia de carga para las características del led).

Memorias **DDRAM**, **CGRAM** y **CGROM**

El LCD posee tres tipos de memoria, dos de almacenamiento temporal: una denominada **DDRAM** (Data Display RAM) y la otra denominada **CGRAM** (Character Generator RAM), y la tercer memoria interna no volátil denominada **CGROM** (Character Generator ROM), para el estudio de esta parte del LCD, se toma como base las especificaciones del dispositivo electrónico HD44780U del fabricante Hitachi ya que es el sistema de control utilizado por los fabricantes para la mayoría de los LCD's.

La memoria **DDRAM** es donde se almacenan los caracteres (ASCII, Símbolos matemáticos, etc.) que se visualizan en la pantalla del LCD, de la cual se visualizan 16 bytes por línea (aunque almacena 40 bytes dentro de la memoria), se debe tener en cuenta que cada vez que un dato se escribe en la **DDRAM** automáticamente se apunta a la posición siguiente de la memoria para realizar la escritura del dato que continua.

El intervalo de las direcciones en la memoria **DDRAM** son:

- Línea 1** Desde la dirección 00h hasta la 27h, donde la dirección 00h indica el inicio de la línea 1.
- Línea 2** Desde la dirección 40h hasta la 67h, donde la dirección 40h indica el inicio de la línea 2.

Y una vez terminado el intervalo establecido de la memoria para cada línea, se repite cíclicamente pasando de la dirección 27h a la dirección 00h para la línea 1 y de la dirección 67h a la dirección 40h para la línea 2.

La memoria de almacenamiento temporal **CGRAM** esta diseñada para permitir la creación de ocho nuevos caracteres por el usuario (para este proyecto no se crearan nuevos caracteres, pero si se desea la creación de algún carácter (Ver Anexo) en las hojas de especificaciones del dispositivo HD44780U del fabricante Hitachi).

En la memoria **CGROM** se almacena una tabla no volátil con 192 caracteres que pueden ser visualizados en el LCD (Ver Tabla II.2.3), los cuales tienen su representación binaria de ocho bits y se transmiten por el bus de datos.

Tabla II.2.3 Tabla con los 192 caracteres definidos en la memoria **CGROM** de las hojas de especificaciones del dispositivo electrónico HD44780U del fabricante Hitachi.

CGROM	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000			0	1	P	'	P				-	△	≡	α	ρ	
xxxx0001	(2)		!	1	A	Q	a	q			□	ア	チ	△	ā	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	×	ρ	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	ε	∞	
xxxx0100	(5)		\$	4	D	T	d	t			、	エ	ト	μ	Ω	
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	1	σ	Ω
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			フ	キ	ヌ	ヲ	q	π
xxxx1000	(1)		<	8	H	X	h	x			イ	ウ	ネ	リ	フ	Σ
xxxx1001	(2)		>	9	I	Y	i	y			ウ	ケ	ル	ル	フ	γ
xxxx1010	(3)		*	:	J	Z	j	z			エ	コ	ハ	レ	j	κ
xxxx1011	(4)		+	;	K	L	k	l			オ	サ	ヒ	ロ	*	π
xxxx1100	(5)		,	<	L	¥	l	l			カ	シ	フ	ワ	φ	π
xxxx1101	(6)		=	=	M	J	m	}			ユ	ズ	ハ	ン	も	÷
xxxx1110	(7)		.	>	N	^	n	^			ヨ	セ	ホ	°	ñ	
xxxx1111	(8)		/	?	O	_	o	†			ツ	リ	マ	°	ö	■

Para determinar el modo de funcionamiento del LCD dispone de una serie de instrucciones o comandos que se describen en la Tabla II.2.4.

Tabla II.2.4 Tabla de instrucciones del LCD

Instrucción	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Descripción	t
Clear Display	0	0	0	0	0	0	0	0	0	1	Borra la pantalla del LCD y coloca el cursor en la primera posición	
Return Home	0	0	0	0	0	0	0	0	1	*	Cursor a dirección origen (dirección 00h)	1,7ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Modo Entrada I/D=1 se incrementa cursor I/D=0 se decrementa S=1 se desplaza la visualización cada vez que se escribe un dato	40µs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Control de la pantalla D=0 Display apagado D=1 Display encendido C=0 Cursor apagado C=1 Cursor encendido B=0 Parpadeo del cursor apagado B=1 Parpadeo del cursor encendido	40µs
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Control de desplazamientos del cursor y el Display R/L=0 Desplazamiento a la izquierda R/L=1 Desplazamiento a la derecha S/C=0 Desplazamiento solo en el cursor S/C=1 Desplazamiento en todo el Display	40µs
Function Set	0	0	0	0	1	DL	N	F	*	*	Características de control del hardware F=0 Fuente de 5x8 puntos F=1 Fuente de 5x10 puntos N=0 Display de 1 línea N=1 Display de 2 líneas DL=0 Comunicación con 4 bits DL=1 Comunicación con 8 bits	40µs
Set CGRAM Address	0	0	0	1	CGRAM Address					Los datos en CGRAM son enviados y recibidos en la dirección señalada	40µs	
Set DDRAM Address	0	0	1	DDRAM Address					Los datos en DDRAM son enviados y recibidos en la dirección señalada	40µs		
Read Busy Flag	0	1	BF	DDRAM Address					Indica con el bit BF si la dirección en la que apunta esta ocupada con una proceso interno, para leer dicha dirección	0µs		
Write RAM	1	0	Write Data					Escribe datos dentro de DDRAM o CGRAM	40µs			
Read RAM	1	1	Read Data					Lee datos desde DDRAM o CGRAM	40µs			

Inicialización del LCD por instrucciones con conexión de bus de 4 bits

El diagrama de flujo que se muestra en la siguiente figura, realiza paso a paso el proceso de inicialización y configuración del LCD con la conexión de bus de 4 líneas, se utiliza esta configuración para tener la mayor parte de pines disponibles en la parte de control, ya que el control del LCD será compartido con el teclado hexadecimal (como se explica más adelante dentro del mismo capítulo). Primero se hace una explicación de la forma de configurar el LCD por medio de diagrama de flujo, después se hace la programación necesaria que se ocupa en el proyecto y se termina con el diagrama electrónico que se emplea para el mismo.

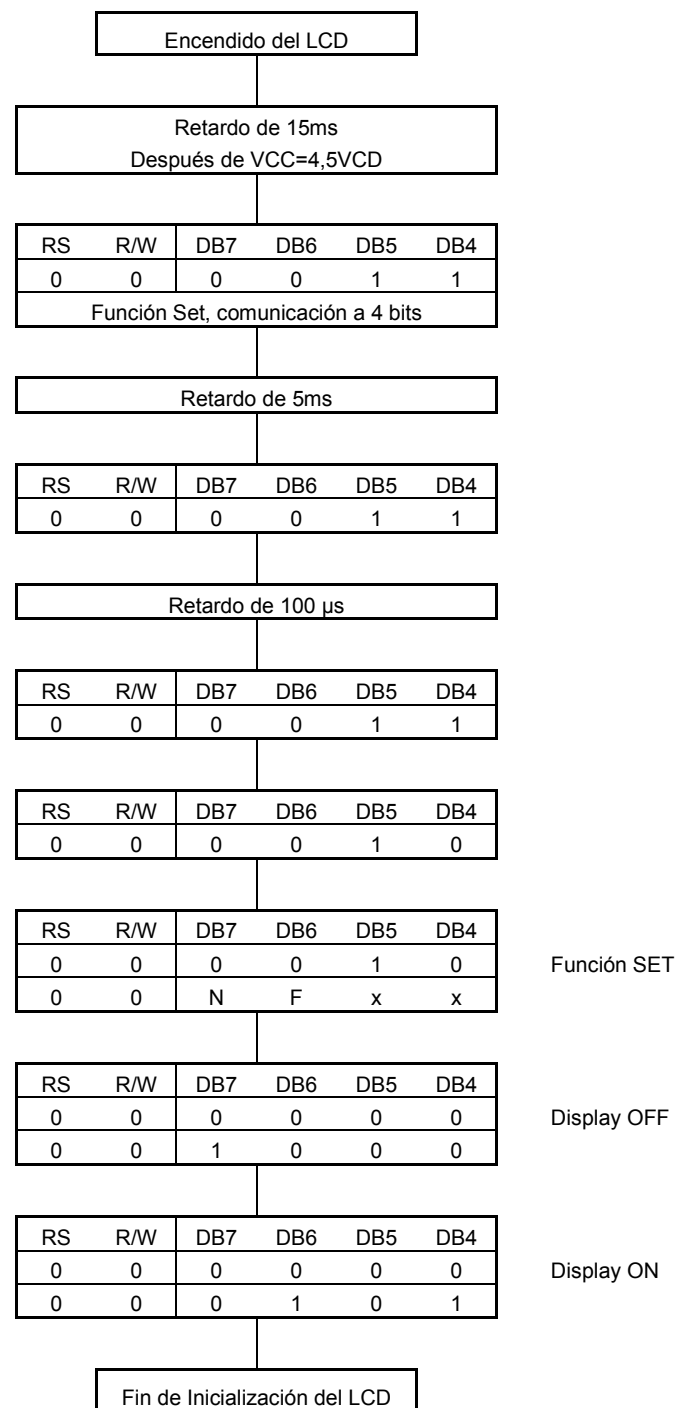


Fig. II.2.2 Diagrama de procesos para configurar e inicializar el LCD.

El dispositivo electrónico que controla al LCD es un microcontrolador PIC16F84A cuyas características del dispositivo se describen más adelante en este capítulo, ya que dicho microcontrolador será el dispositivo base en diversas partes de control en el transcurso del desarrollo de todo el proyecto. Por el momento sólo se aclara el tipo de componente a utilizar para realizar las librerías de programación para el LCD.

Las librerías¹ son un conjunto de subrutinas en los sistemas de programación para funciones específicas y repetitivas, que se utilizan constantemente para un lenguaje de programación y en su conjunto construyen un sistema de control, es decir, si en el programa se necesita hacer que un dispositivo realice un retardo en el tiempo de ejecución para continuar su proceso, se ejecuta una subrutina que se denomina Retardo y se especifica el tiempo que se desea para dicha subrutina, por ejemplo *Retardo_1s* y la forma en la que se llama la subrutina en el lenguaje de programación será:

```
call Retardo_1s
```

Así el programa llamará a la subrutina *Retardo_1s* que se encuentra en la librería *Retardos.INC* (Ver Anexo).

Ya que el estudio del microcontrolador no es el objetivo principal del proyecto, sino sólo es una herramienta para su desarrollo, en esta parte del capítulo se realiza la programación de las subrutinas y el lenguaje de programación necesarios para el funcionamiento del LCD enfocado hacia el uso predeterminado en el desarrollo del sistema de control para el proyecto propuesto y para el análisis del microcontrolador sólo se detallarán aspectos específicos que algún dispositivo electrónico lo amerite.

La librería que se usa para la programación en el inicio del LCD de acuerdo al diagrama de proceso de la Fig. II.2.2 es la siguiente:

```

;
;***** Librería "LCD.INC" *****
;
;
; Pin R/W del LCD a masa
; Se utilizan llamadas a subrutinas de la librería RETARDOS.INC
;
;
; ZONA DE DATOS *****
;
; CBLOCK
; LCD_Dato
; LCD_GuardaDato
; LCD_GuardaTRISB
; LCD_Auxiliar1
; LCD_Auxiliar2
; ENDC

LCD_CaracteresPorLinea EQU .16 ; Número de caracteres por línea de la pantalla.
#define LCD_PinRS PORTA,0 ; Pin RS del LCD a la línea RA0 del PIC
#define LCD_PinEnable PORTA,2 ; Pin Enable del LCD a la línea RA2 del PIC
#define LCD_BusDatos PORTB ; Las 4 líneas <DB7:DB4> del módulo LCD
; Se conectan a las 4 <RB7:RB4> del Puerto B del PIC

; Subrutina "LCD_EscribeLCD" -----
; El dato a enviar debe estar en W

LCD_EscribeLCD
    andlw b'11110000' ; Se queda con el nibble alto del dato que es el
    movwf LCD_Dato ; que hay que enviar y lo guarda.
    movf LCD_BusDatos,W ; Lee la información actual de la parte baja
    andlw b'00001111' ; del Puerto B, que no se debe alterar.
    iorwf LCD_Dato,F ; Enviar la parte alta del dato de entrada
; y en la parte baja lo que había antes.
    bsf STATUS,RP0 ; Acceso al Banco 1.
    movf TRISB,W ; Guarda la configuración que tenía antes TRISB.
    movwf LCD_GuardaTRISB

```

¹ Es importante aclarar que el sistema de programación utiliza una gran variedad de subrutinas o librerías y es por ello que el lenguaje de programación de cada una de las mismas se encuentra en un apartado dentro del ANEXO. La librería "RETARDOS.INC" es parte de la programación del libro "Microcontrolador PIC16F84A. Desarrollo de proyectos., E. Palacios, F. Remiro y L. López., Editorial Ra-Ma"

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos, basado en un protocolo propio"

movlw	b'00001111'	; Las 4 líneas inferiores del Puerto B se dejan
andwf	PORTB,F	; como estaban y las 4 superiores como salida.
bcf	STATUS,RP0	; Acceso al Banco 0.
;		
movf	LCD_Dato,W	; Recupera el dato a enviar.
movwf	LCD_BusDatos	; Envía el dato al módulo LCD.
bsf	LCD_PinEnable	; Permite funcionamiento del LCD mediante un pequeño
bcf	LCD_PinEnable	; pulso y termina impidiendo el funcionamiento del LCD.
bsf	STATUS,RP0	; Acceso al Banco 1. Restaura el antiguo valor en
movf	LCD_GuardaTRISB,W	; la configuración del Puerto B.
movwf	PORTB	; Realmente es TRISB.
bcf	STATUS,RP0	; Acceso al Banco 0.
return		
;Subrutina "LCD_Envia" -----		
; El dato a enviar debe estar en W		
LCD_EnviaComando		
bcf	LCD_PinRS	; Activa el Modo Comando, poniendo RS=0.
goto	LCD_Envia	
LCD_Caracter		
bsf	LCD_PinRS	; Activa el "Modo Dato", poniendo RS=1.
call	LCD_CodigoCGROM	; Obtiene el código para correcta visualización.
LCD_Envia		
movwf	LCD_GuardaDato	; Guarda el dato a enviar.
call	LCD_EscribeLCD	; Primero envía el nibble alto.
swapf	LCD_GuardaDato,W	; Ahora envía el nibble bajo. Para ello pasa el
		; nibble bajo del dato a enviar a parte alta del byte.
call	LCD_EscribeLCD	; Se envía al visualizador LCD.
btfs	LCD_PinRS	; Debe garantizar una correcta escritura manteniendo
call	Retardo_2ms	; 2 ms en modo comando y 50 µs en modo carácter.
call	Retardo_50micros	
return		
;Subrutina "LCD_CodigoCGROM;" -----		
LCD_CodigoCGROM		
movwf	LCD_Dato	; Guarda el valor del carácter y comprueba si es
LCD_EnheMinuscula		
sublw	'ñ'	; un carácter especial.
btfs	STATUS,Z	; ¿Es la "ñ"?
goto	LCD_EnheMayuscula	; No es "ñ".
movlw	b'11101110'	; Código CGROM de la "ñ".
movwf	LCD_Dato	
goto	LCD_FinCGROM	
LCD_EnheMayuscula		
movf	LCD_Dato,W	; Recupera el código ASCII de entrada.
sublw	'Ñ'	; ¿Es la "Ñ"?
btfs	STATUS,Z	
goto	LCD_Grado	; No es "Ñ".
movlw	b'11101110'	; Código CGROM de la "ñ". (No hay símbolo para
movwf	LCD_Dato	; la "Ñ" mayúscula en la CGROM).
goto	LCD_FinCGROM	
LCD_Grado		
movf	LCD_Dato,W	; Recupera el código ASCII de entrada.
sublw	'º'	; ¿Es el símbolo "º"?
btfs	STATUS,Z	
goto	LCD_FinCGROM	; No es "º".
movlw	b'11011111'	; Código CGROM del símbolo "º".
movwf	LCD_Dato	
LCD_FinCGROM		
movf	LCD_Dato,W	; En (W) el código buscado.
return		
; Subrutina "LCD_Inicializa" -----		
; Inicialización del módulo LCD de acuerdo al Diagrama de Procesos de la Fig. 4.2		
LCD_Inicializa		
bsf	STATUS,RP0	; Configura las líneas conectadas al pines RS,
bcf	LCD_PinRS	; R/W y E.
bcf	LCD_PinEnable	
bcf	STATUS,RP0	
bcf	LCD_PinRW	; En caso de que esté conectado le indica
		; que se va a escribir en el LCD.
bcf	LCD_PinEnable	; Impide funcionamiento del LCD poniendo E=0.
bcf	LCD_PinRS	; Activa el Modo Comando poniendo RS=0.

Elaboración del proyecto “Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio”

```

call    Retardo_20ms
movlw  b'00110000'
call    LCD_EscribeLCD           ; Escribe el dato en el LCD.
call    Retardo_5ms
movlw  b'00110000'
call    LCD_EscribeLCD
call    Retardo_200micros
movlw  b'00110000'
call    LCD_EscribeLCD
movlw  b'00100000'           ; Interface de 4 bits.
call    LCD_EscribeLCD

movlw  b'00101000'           ; Configura LCD de 2 líneas y caracteres de 5x7 puntos.
bcf    LCD_PinRS             ; Activa el Modo Comando, poniendo RS=0.
goto   LCD_Envia

movlw  b'00000001'           ; Pantalla encendida y limpia. Cursor al principio de la línea 1
bcf    LCD_PinRS             ; Activa el Modo Comando, poniendo RS=0.
goto   LCD_Envia

movlw  b'00001100'           ;Pantalla encendida y cursor apagado.
bcf    LCD_PinRS             ; Activa el Modo Comando, poniendo RS=0.
goto   LCD_Envia

movlw  b'00000110'           ; Cursor en modo incrementar.
bcf    LCD_PinRS             ; Activa el Modo Comando, poniendo RS=0.
goto   LCD_Envia
return

; Subrutinas diversas para control del LCD -----
LCD_Linea1
movlw  b'10000000'           ; Cursor al principio de la Línea 1.
goto   LCD_EnviaComando      ; Dirección 00h de la DDRAM

LCD_Linea2
movlw  b'11000000'           ; Cursor al principio de la Línea 2.
goto   LCD_EnviaComando      ; Dirección 40h de la DDRAM

LCD_Borra
movlw  b'00000001'           ; Borra toda la pantalla, memoria DDRAM y pone el
goto   LCD_EnviaComando      ; cursor a principio de la línea 1.

```

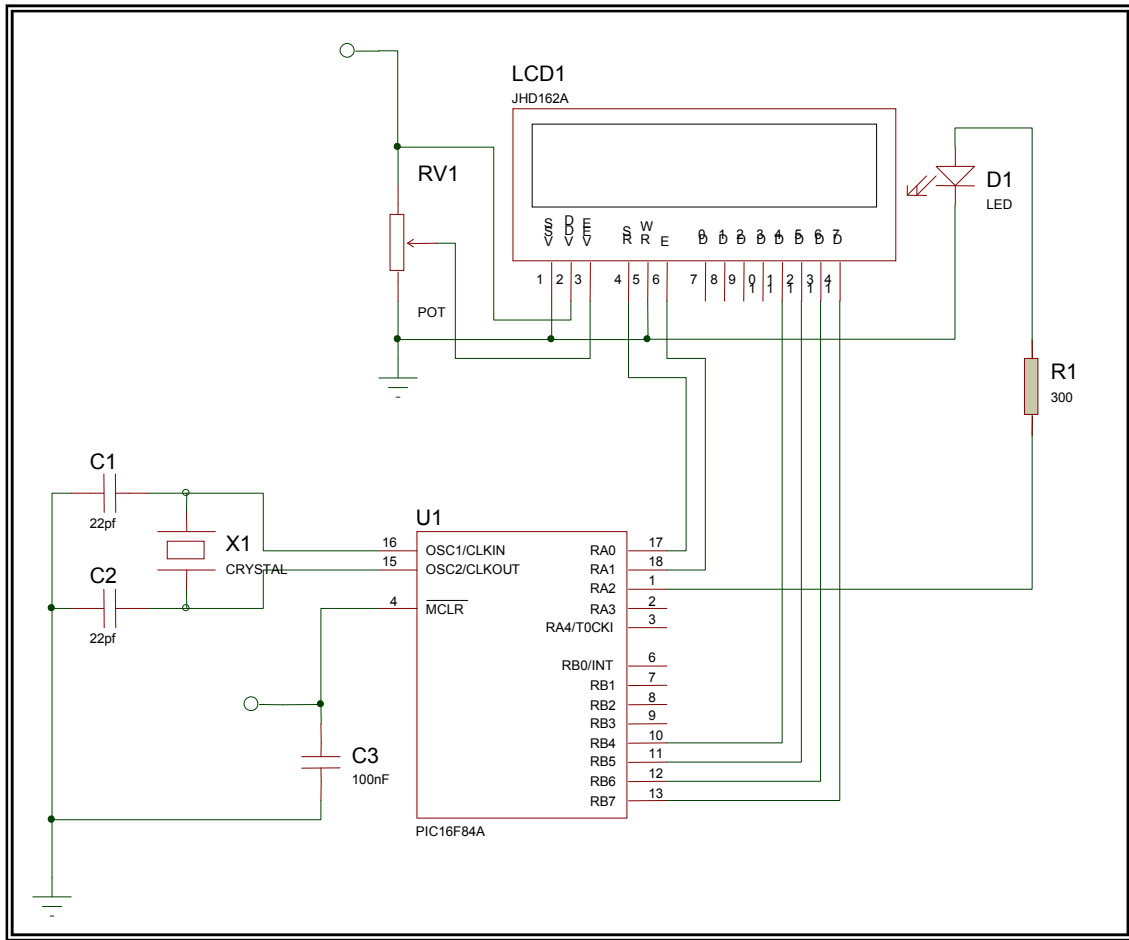


Fig. II.2.3 Diagrama electrónico para la conexión del LCD JHD162A con el PIC16F84A.

Teclado hexadecimal como dispositivo para ingreso de datos

El teclado es un periférico de entrada, el cual tiene la función de acoplar información del usuario con algún módulo de control, es decir, el usuario introduce la información necesaria para que el sistema de control pueda establecer el proceso que el usuario desea. El funcionamiento básico del teclado es activar interruptores (*push button*) en forma independiente donde cada uno tiene una función específica (Ver Fig. II.2.4).

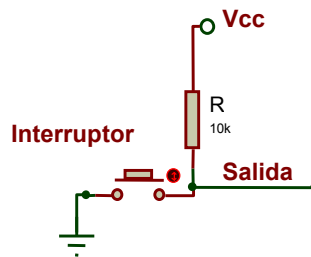


Fig. II.2.4 Configuración típica de un interruptor.

El reducir el número de componentes y la simplificación de la programación es aspecto importante en la realización de cualquier diseño, si se utiliza un interruptor y una salida para cada función, el nivel de componentes aumenta al igual que la complejidad en la programación para el control de cada interruptor, por ejemplo un celular tiene aproximadamente 21 teclas y si se utiliza una señal de entrada independiente para cada tecla se necesita de 21 pines de control, es por esto que es necesario reducir el número de señales de entrada, pero obtener las mismas señales independientes de control, de forma más clara, en la Fig. II.2.4 se tiene una señal de control para cada interruptor, pero al combinar las señales de entrada se tiene las señales de control mostradas en la Tabla II.2.5.

Tabla II.2.5 Comparación de estados de control con los mismas señales de entrada

Fig. 2.1			Combinación de Señales		
Señales de Control	Interruptor A	Interruptor B	Señales de Control	Interruptor A	Interruptor B
1	Encendido	Apagado	1	Apagado	Apagado
2	Apagado	Encendido	2	Encendido	Apagado
			3	Apagado	Encendido
			4	Encendido	Encendido

En el dispositivo Maestro se utiliza un teclado matricial el cual minimiza el número pines de control, pero realiza las mismas funciones independientes obedeciendo a una combinación matricial, agregando a esto la multifuncionalidad de los interruptores dependiendo del proceso en el que se encuentre.

Descripción de un teclado matricial

Hay que recordar que el teclado matricial es un dispositivo electrónico digital, por lo que la forma en que opera depende de valores binarios "1" y "0", el funcionamiento del teclado es reflejo de una matriz y el número de combinaciones son las señales de control, por ejemplo si es una matriz de 3x3 el número de combinaciones es 9 y las mismas son las señales de control, si se utiliza una matriz de 6x6 el total de combinaciones será de 36. El dispositivo Maestro utiliza (en su desarrollo de diseño) 12 interruptores, por lo que una matriz de 4x4 entrega las combinaciones de control necesarias.

El teclado matricial de 4x4 funciona haciendo dos bloques de cuatro bits cada uno, los primeros cuatro son de escritura y los cuatro restantes son de lectura (Ver Fig. II.2.5), al unirlos se hace un bloque de ocho bits, si fuera una matriz de 5x5 se hacen dos bloques de cinco bits cada uno.

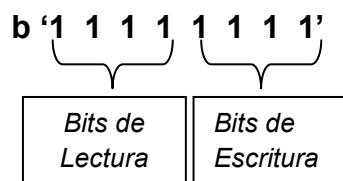


Fig. II.2.5 Distribución de bits para la etapa de control

En el bloque de escritura se entrega una serie de combinaciones las cuales se interpretan como un bit por cada columna identificando la columna cuando el bit marca un cero lógico, el bloque de bits de lectura tiene la función de reflejar los primeros cuatro bits de escritura, donde al leer el dato se identifica la fila cuando existe un cero lógico, además estos bits nos indican si se ha oprimido alguna tecla, es decir, en un teclado donde no se ha oprimido algún interruptor, al leer el dato en los bits de lectura siempre será de 1111 (Ver Fig. II.2.6 y Tabla II.2.6).

Fig. II.2.6 El teclado físico consta de 15 teclas y 8 pines de control

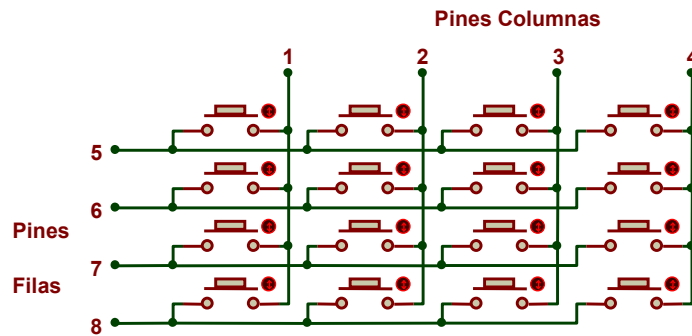


Tabla II.2.6 Representación de un teclado matricial de 4x4 sin presionar interruptores

		Columna 1	Columna 2	Columna 3	Columna 4	
		Bits de Escritura				
		1	1	1	1	
Fila 1	Bits de Lectura	1	0	1	2	3
Fila 2		1	4	5	6	7
Fila 3		1	8	9	10	11
Fila 4		1	12	13	14	15

El bloque de escritura tiene la función de monitorear el teclado haciendo un barrido de bits interpretando a cada columna, se empieza por la columna 1 hasta llegar a la columna 4, haciendo el recorrido mostrado en la Tabla II.2.7.

Tabla II.2.7 Recorrido para monitoreo del teclado

Dato binario a escribir	Columna			
	1	2	3	4
0111	0	1	1	1
1011	1	0	1	1
1101	1	1	0	1
1110	1	1	1	0

El sistema de control para el dispositivo Maestro tiene que establecer dos rutinas para el control del teclado, la primera es para escribir el dato binario y monitorear la columna que se desee y la segunda rutina para leer el dato en el bloque de lectura y determinar si está en funcionamiento el teclado o saber con exactitud la tecla presionada, por ejemplo si se monitorea la columna 1 y no se ha presionado tecla alguna, el bloque de lectura de ocho bits en binario es **b'11110111'**; así mismo si se monitorea la columna 1 y se encuentra presionada la tecla 8, entonces el bloque de lectura de ocho bits en binario es **b'11010111'**.

Es importante destacar que para el control completo del teclado el sistema de control principal debe hacer un recorrido por cada una de las cuatro columnas y realizar las dos rutinas (escritura y lectura) por cada columna para así poder monitorear las 16 teclas.

El sistema de control en la etapa **Procesamiento de Dato**, tiene la capacidad de transmitir datos basado en el bus serie I2C, este bus no tiene la velocidad que tiene una interfase de comunicación en paralelo pero la velocidad que proporciona es suficiente, además una de las ventajas que proporciona es el poco cableado que necesita (SDA, SCL, y en algunos casos un pin de interrupción).

Si se utiliza el teclado matricial, se necesitan 8 pines de control, si se utiliza un componente de acoplamiento a BUS I2C el número de pines de control se reduce a 3, además el bus I2C tiene la posibilidad de conectar hasta 255 dispositivos periféricos ya sea como entrada o como salida.

El dispositivo de acoplamiento a utilizar es el PCF8574P (*Ver Anexo*), es un expansor de bus I2C fabricado por *Philips Semiconductors*, ver *Fig. II.2.7*.

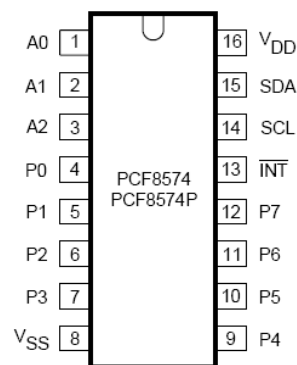


Fig. II.2.7 Diagrama físico del PCF8574

Descripción del PCF8574P

- ⇒ El PCF8574P puede configurarse como entrada o como salida en un puerto de 8 pines
- ⇒ Tiene un direccionamiento de 8 bits, con tres pines programables
- ⇒ Cuando se configura como salida se mantiene latched²
- ⇒ Voltaje de alimentación de 5VCD³
- ⇒ Está diseñado con una salida para interrupciones (INT) cuando se configura como entrada

El funcionamiento básico del expansor PCF8574P, es transformar datos de 8 bits en paralelo a serie y transmitirlos con el protocolo del bus serie I2C de *Philips* (Ver Fig. II.2.8), es decir, si se tienen 8 bits para transmitir un dato en un microcontrolador u otro dispositivo (memorias, GAL, etc.) se necesitan 8 pines específicos para el envío del dato (un bit por cada pin), el bus serie I2C tiene la capacidad de transmitir el mismo dato de 8 bits con tan sólo 2 pines (cuando se configura como salida) SDA y SCL, haciendo de esto un aspecto importante para disminuir el número de pines de control ya que se tiene una mayor posibilidad de conectar diversos periféricos con un mínimo de pines, además de la posibilidad de conectar hasta 16 dispositivos por su direccionamiento por hardware (A0, A1 y A2).

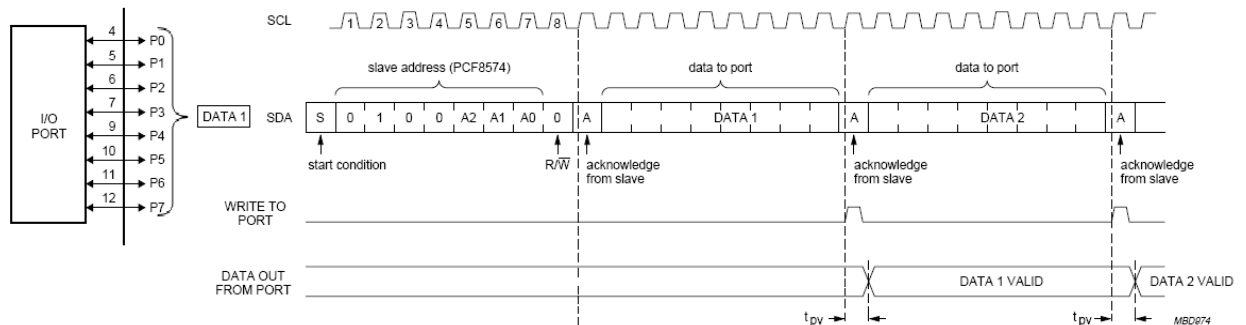


Fig. II.2.8 Diagrama de tiempo para el funcionamiento del PCF8574P

La configuración final para el teclado con el módulo de control de esta etapa, consta de los bloques mostrados en la Fig.II.2.9.

² Mantiene la información en la salida hasta nuevo dato a escribir

³ Valor típico. Ver Anexo.

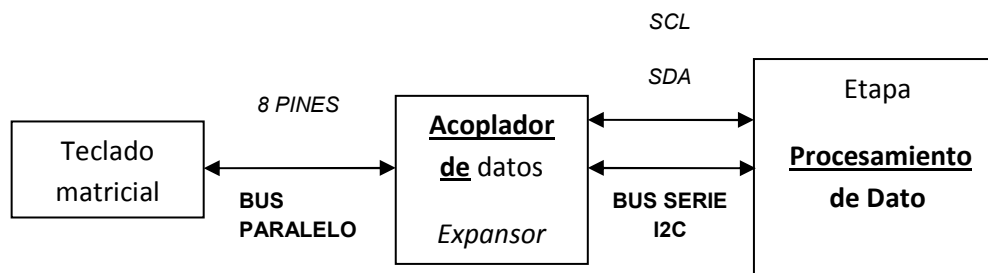


Fig. II.2.9 Diagrama a bloques del Teclado matricial al control a la etapa **Procesamiento de Dato**

La programación del teclado matricial con la etapa de control para este módulo, esta compuesto con subrutinas para cada bloque que conforma el control, es decir, la comunicación del teclado matricial con el dispositivo PCF8574P esta controlado por subrutinas que se encuentran dentro de la librería “TECLADO.INC” y el dispositivo PCF8574P tiene a su vez una librería denominada “PCF8574P.INC⁴ y éste se comunica con el sistema de control que se encuentra en la etapa *Procesamiento de Dato* por medio de un protocolo de comunicación denominado BUS I2C y para ello se necesita una librería denominada “BUS_I2C.INC⁵”.

Las librerías anteriores en conjunto con la librería “RETARDOS.INC” hacen una sistematización del programa quedando las siguientes llamadas a subrutinas:

- Subrutina “Teclado_Inicializa” (configura la dirección del teclado y realiza las configuraciones correspondientes para el correcto funcionamiento)
- Subrutina “Teclado_LeeOrdenTecla” (lee el dato obtenido por el PCF8574P al oprimir una tecla)
- Subrutina “Teclado_EsperaDejePulsar” (Espera a que se estabilicen los niveles de tensión en los pulsadores del teclado)

Estas subrutinas se utilizan para la programación en la etapa *Procesamiento de Dato* para el control del teclado hexadecimal.

⁴ La librería se encuentra en el ANEXO y es parte de la programación del libro “Microcontrolador PIC16F84A. Desarrollo de proyectos., E. Palacios, F. Remiro y L. López., Editorial Ra-Ma”

⁵ Ídem

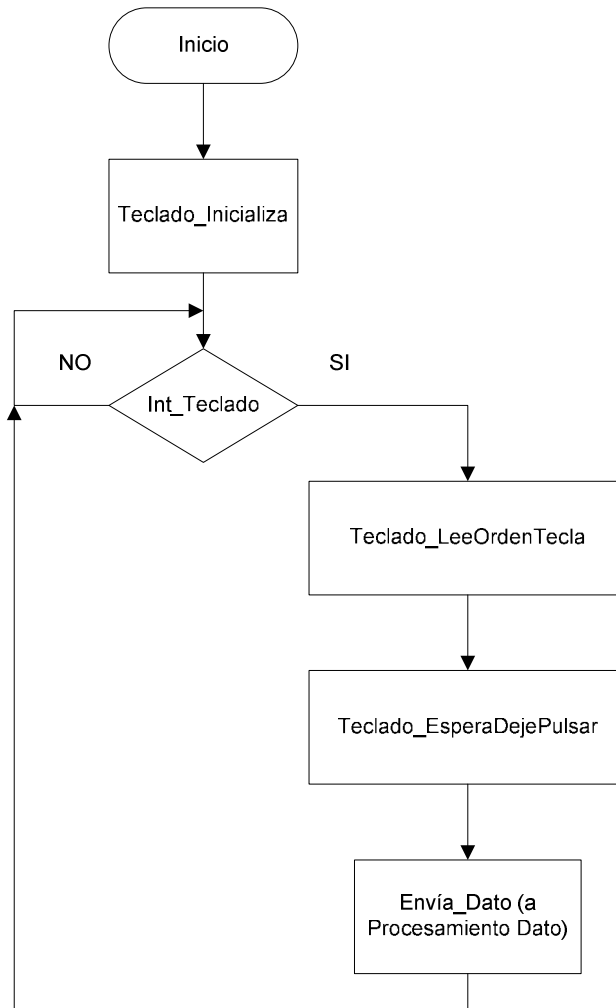


Fig. II.2.10 Diagrama de flujo para el control del Teclado matricial.

El lenguaje de programación utilizado para el *Diagrama de flujo para el control del Teclado matricial* que se muestra en la Fig. II.2.10., se desarrolla a continuación, en la librería TECLADO.INC:

```

;
;
;***** Librería "TECLADO.INC" *****
;
;
CBLOCK
    Tecl_TeclaOrden          ; Orden de la tecla a chequear.
    ENDC

PCF8574_DireccionLectura EQU  b'01000001'
PCF8574_DireccionEscritura EQU b'01000000'

Tecl_NumeroTeclas      EQU  d'16'      ; Número de pulsadores del teclado.

#DEFINE PCF8574_INT    PORTB,0        ; Línea donde se conecta la línea de
                                        ; interrupción del expander de bus I2C.

; Subrutina "Teclado_Inicializa" -----
;
Teclado_Inicializa
    bsf    STATUS,RP0                ; Acceso al Banco 1.
    bsf    PCF8574_INT               ; La línea RB0/INT se configura como entrada.
    bcf    OPTION_REG,NOT_RBPU       ; Habilita las resistencias de Pull-Up del Puerto B.
    bcf    OPTION_REG,INTEDG         ; Interrupción INT activa por flanco de bajada.
    bcf    STATUS,RP0                ; Acceso al Banco 0.
    call   Teclado_EsperaDejePulsar
    return
    
```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio"

```

; Subrutina "Teclado_EsperaDejePulsar" -----
;
Teclado_Comprobacion EQU b'11110000'

Teclado_EsperaDejePulsar:
    movlw Teclado_Comprobacion ; Pone a cero las cuatro líneas de salida del puerto
    call PCF8574_Escribe ; para que se produzca una interrupción cuando se
Teclado_SigueEsperando ; presione alguna tecla.
    call Retardo_20ms
    call PCF8574_Lee ; Lee el puerto.
    sublw Teclado_Comprobacion ; Si es lo mismo que lanzó es que ya no pulsa
    btfss STATUS,Z ; tecla alguna.
    goto Teclado_SigueEsperando
    return

; Subrutina "Teclado_LeeOrdenTecla" -----
;
; Lee el teclado, obteniendo el orden de la tecla pulsada.
;
; La Salida se ubica en (W) y es el número de orden de la tecla pulsada.

    CBLOCK
    Tecl_I2C_DatoLeido ; Guarda el dato leído.
    Tecl_I2C_FilaChequear ; Fila que va a ser chequeada.
    ENDC

Teclado_LeeOrdenTecla:
    clrf Tecl_TeclaOrden ; Todavía no ha empezado a chequear el teclado.
    movlw b'11111110' ; Va a chequear primera fila.
    movwf Tecl_I2C_FilaChequear
Tecl_ChequeaFila ; (Ver esquema de conexión).
    call PCF8574_Escribe ; Activa la fila correspondiente.
    call PCF8574_Lee ; Lee el puerto.
    movwf Tecl_I2C_DatoLeido ; Guarda la lectura efectuada.
Tecl_Columna1
    btfss Tecl_I2C_DatoLeido,4 ; Chequea 1ª columna buscando un cero.
    goto Tecl_GuardaValor ; Si, es cero. Salta a guardar su valor y sale.
    incf Tecl_TeclaOrden,F ; No es cero. Va a chequear la siguiente tecla.
Tecl_Columna2 ; Repite proceso para las siguientes
    btfss Tecl_I2C_DatoLeido,5 ; columnas.
    goto Tecl_GuardaValor
    incf Tecl_TeclaOrden,F
Tecl_Columna3
    btfss Tecl_I2C_DatoLeido,6
    goto Tecl_GuardaValor
    incf Tecl_TeclaOrden,F
Tecl_Columna4
    btfss Tecl_I2C_DatoLeido,7
    goto Tecl_GuardaValor
    incf Tecl_TeclaOrden,F
;
; Comprueba si ha chequeado la última tecla en cuyo caso sale. Para ello comprueba si
; el contenido del registro Tecl_TeclaOrden es igual al número de teclas del teclado.
;
Tecl_TerminaColumnas
    movlw Tecl_NumeroTeclas
    subwf Tecl_TeclaOrden,W
    btfsc STATUS,Z
    goto Tecl_NoPulsada ; Sí, se ha llegado al final del chequeo.
    bsf STATUS,C ; No. Va a chequear la siguiente fila.
    rlf Tecl_I2C_FilaChequear,F ; Apunta a la siguiente fila.
    movf Tecl_I2C_FilaChequear,W
    goto Tecl_ChequeaFila
Tecl_NoPulsada
    bcf STATUS,C ; Posiciona C=0 (carry=0), indicando que no ha pulsado
    goto Tecl_FinTecladoLee ; tecla alguna y sale.
Tecl_GuardaValor
    movf Tecl_TeclaOrden,W ; El orden de la tecla pulsada en (W).
    bsf STATUS,C ; Como hay tecla pulsada pone C=1 (carry=1).
Tecl_FinTecladoLee
    return
    
```

El diagrama en la Fig. II.2.11 muestra como se conecta el teclado matricial con la etapa de control

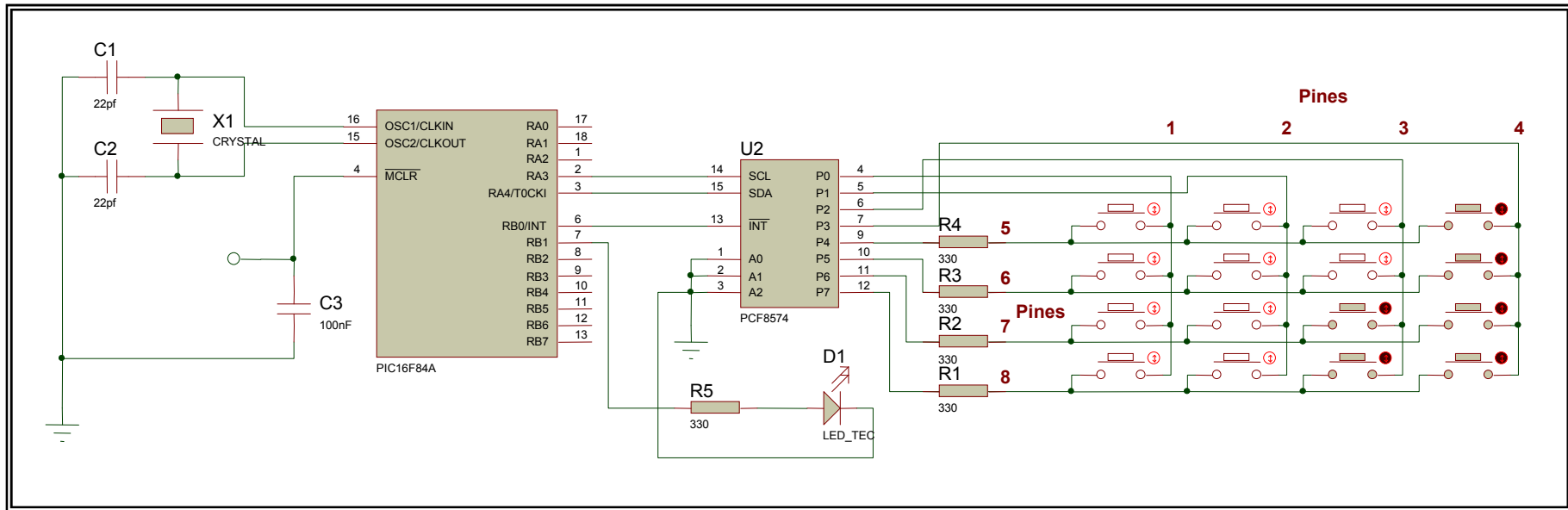


Fig. II.2.11 Diagrama electrónico para la conexión del Teclado hexadecimal con el PIC16F84A.

Procesamiento de Dato

En esta parte se procesa la información recibida por medio del Teclado hexadecimal para enviar los datos a la pantalla LCD y al PCF8574P, éste último sirve como medio de acoplamiento en paralelo de 8 bits con la Etapa de *Codificación de Señal* (como se muestra en la Fig. II.2.12)

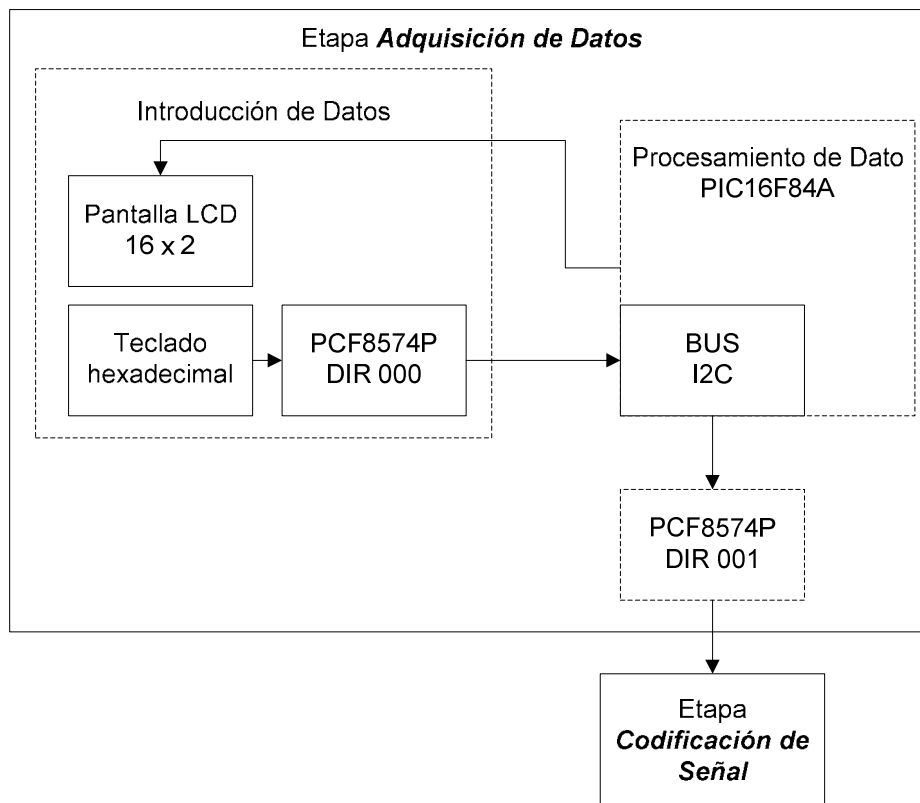


Fig. II.2.12 Diagrama de procesos completo de la Etapa **Adquisición de Datos**.

La distribución del teclado para la programación del dispositivo maestro es la siguiente:

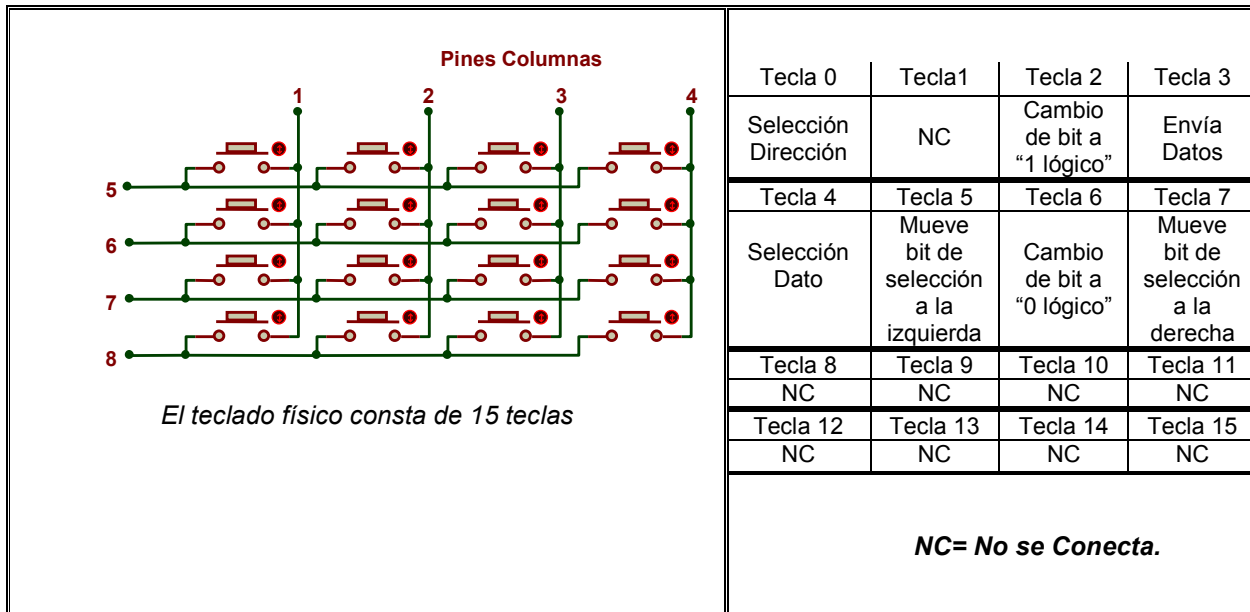


Fig. II.2.13 Muestra la distribución y funciones del teclado matricial, que se utiliza para la programación en la etapa **Procesamiento de Dato**.

El diagrama de flujo a seguir en la programación del microcontrolador para que realice las funciones del diagrama de procesos se muestra en la Fig. II.2.13.

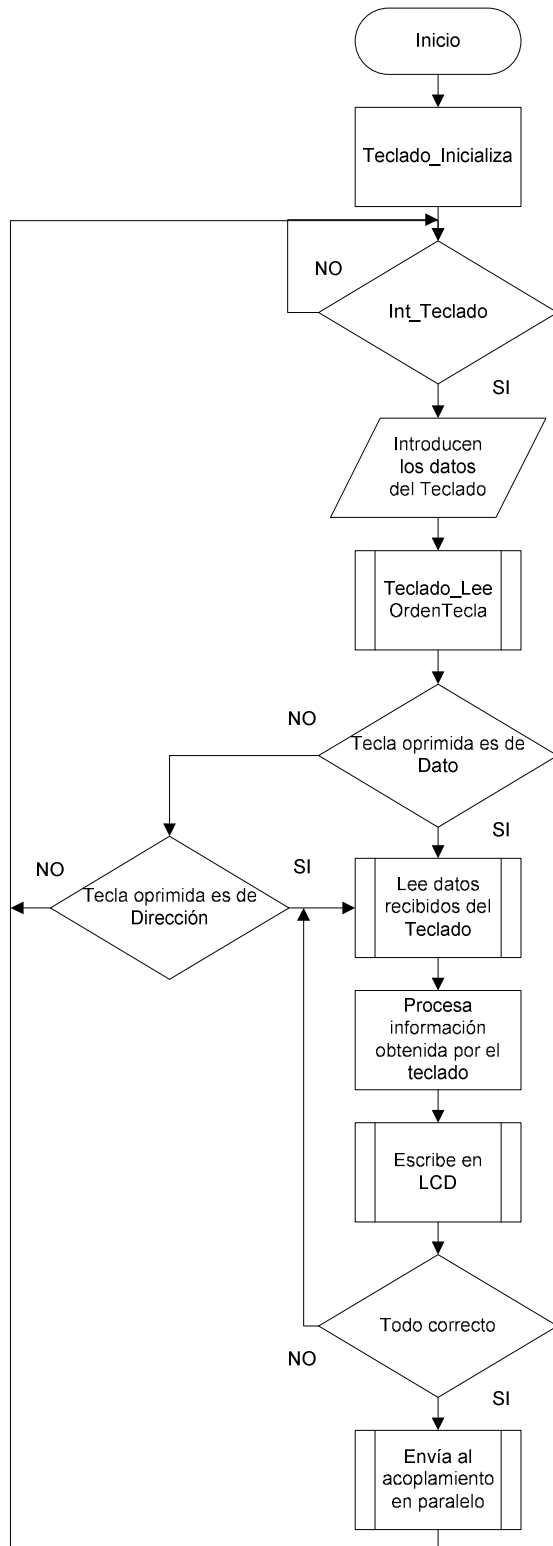


Fig. II.2.14 Diagrama de flujo para la programación del microcontrolador que se usa para el Procesamiento de Dato de la Etapa Adquisición de Datos.

La programación para el microcontrolador que cumple con el diagrama de flujo anterior se explica en el programa *Procesamiento_Dato.ASM*

```

;
;***** Librería "Procesamiento_Dato.ASM" *****
;
;
; ZONA DE DATOS *****
;
;   _CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
;   LIST     P=16F84A
;   INCLUDE <P16F84A.INC>
;
;   CBLOCK 0x0C
;   LCD_Auxiliar_1
;   TEC_Auxiliar_1
;   Com_DIR
;   Com_DIR_Co
;   Com_DATO
;   Com_DATO_Co
;   Com_AUX
;   LCD_Con
;   LCD_Con_Aux
;   ENDC
;
; ZONA DE CÓDIGOS *****
;
;   ORG      0
;   goto    Inicio
;   ORG      4
;   goto    ServicioInterrupcion
;
Inicio
;   bsf     STATUS,RP0
;   bcf     PORTB,3
;   bcf     PORTB,2
;   bcf     STATUS,RP0
;
;   bcf     PORTB,3
;   clrf   TEC_Auxiliar_1
;   clrf   LCD_Auxiliar_1
;   movlw  b'00000001'
;   movwf  Com_DATO_Co
;   movwf  Com_DIR_Co
;   clrf   Com_DIR
;   clrf   Com_DATO
;   clrf   Com_AUX
;
;   call   LCD_Inicializa
;   call   LCD_Actualiza
;   call   Teclado_Inicializa           ; Prepara al PCF8574 para interrumpir.
;   movlw b'10010000'
;   movwf INTCON
;
Principal
;   sleep
;   goto   Principal
;
;
; Subrutina "ServicioInterrupcion" -----
;
; Lee el teclado conectado al PCF8574 y el resultado se visualiza en la pantalla.
;
ServicioInterrupcion
;   call   Teclado_LeerOrdenTecla
;   btfs  STATUS,C
;   return

```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos, basado en un protocolo propio"

```

        andlw    b'00000111'
        addwf   PCL,F
;Es importante recalcar que el número de teclas que hay en un teclado hexadecimal .
Tabla
        goto    Tecla_0
        goto    Tecla_1
        goto    Tecla_2
        goto    Tecla_3
        goto    Tecla_4
        goto    Tecla_5
        goto    Tecla_6
        goto    Tecla_7

;
Tecla_0
        bsf     LCD_Con,0
        goto    Fin_Teclado_T
;
Tecla_1
        goto    Fin_Teclado_T
;
Tecla_2
        bsf     LCD_Con,1
        call    Cambio_de_BIT
        call    LCD_Actualiza
        goto    Fin_Teclado_T
;
Tecla_3
        call    Envia_Datos
        goto    Fin_Teclado_T
;
Tecla_4
        bcf     LCD_Con,0
        goto    Fin_Teclado_T
;
Tecla_5
        btfs   LCD_Con,0
        goto    RLF_I_DIR

RLF_I_DATO
        btfs   Com_DATO_Co,7
        goto    DATO_RLF
        rif    Com_DATO_Co,F
        goto    Fin_Teclado_T
DATO_RLF
        movlw  b'00000001'
        movwf  Com_DATO_Co
        goto    Fin_Teclado_T

RLF_I_DIR
        btfs   Com_DIR_Co,7
        goto    DIR_RLF
        rif    Com_DIR_Co,F
        goto    Fin_Teclado_T
DIR_RLF
        movlw  b'00000001'
        movwf  Com_DIR_Co
        goto    Fin_Teclado_T
;
Tecla_6
        bcf     LCD_Con,1
        call    Cambio_de_BIT
        call    LCD_Actualiza
        goto    Fin_Teclado_T
;
Tecla_7
        btfs   LCD_Con,0

```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos, basado en un protocolo propio"

```

    goto    RRF_D_DIR
RRF_D_DATO
    btfsc  Com_DATO_Co,0
    goto   DATO_RRF
    rrf    Com_DATO_Co,F
    goto   Fin_Teclado_T
DATO_RRF
    movlw  b'10000000'
    movwf  Com_DATO_Co
    goto   Fin_Teclado_T
RRF_D_DIR
    btfsc  Com_DIR_Co,0
    goto   DIR_RRF
    rrf    Com_DIR_Co,F
    goto   Fin_Teclado_T
DIR_RRF
    movlw  b'10000000'
    movwf  Com_DIR_Co
    goto   Fin_Teclado_T
;
Cambio_de_BIT
    btfsc  Com_DIR_Co,0
    goto   Cam_B0
    btfsc  Com_DIR_Co,1
    goto   Cam_B1
    btfsc  Com_DIR_Co,2
    goto   Cam_B2
    btfsc  Com_DIR_Co,3
    goto   Cam_B3
    btfsc  Com_DIR_Co,4
    goto   Cam_B4
    btfsc  Com_DIR_Co,5
    goto   Cam_B5
    btfsc  Com_DIR_Co,6
    goto   Cam_B6
    btfsc  Com_DIR_Co,7
    goto   Cam_B7
Cam_B0
    btfss  LCD_Con,0
    goto   Cam_B0_DATO
Cam_B0_DIR
    bcf    Com_DIR,0
    btfss LCD_Con,1
    return
    bsf    Com_DIR,0
    return
Cam_B0_DATO
    bcf    Com_DATO,0
    btfss LCD_Con,1
    return
    bsf    Com_DATO,0
    return
;
Cam_B1
    btfss  LCD_Con,0
    goto   Cam_B1_DATO
Cam_B1_DIR
    bcf    Com_DIR,1
    btfss LCD_Con,1
    return
    bsf    Com_DIR,1
    return
Cam_B1_DATO
    bcf    Com_DATO,1
    btfss LCD_Con,1
    return

```

Elaboración del proyecto “Diseño de un sistema electrónico de envío y recepción de datos, basado en un protocolo propio”

	bsf	Com_DATO,1
	return	
;		
Cam_B2	btfs	LCD_Con,0
	goto	Cam_B2_DATO
Cam_B2_DIR	bcf	Com_DIR,2
	btfs	LCD_Con,1
	return	
	bsf	Com_DIR,2
	return	
Cam_B2_DATO	bcf	Com_DATO,2
	btfs	LCD_Con,1
	return	
	bsf	Com_DATO,2
	return	
;		
Cam_B3	btfs	LCD_Con,0
	goto	Cam_B3_DATO
Cam_B3_DIR	bcf	Com_DIR,3
	btfs	LCD_Con,1
	return	
	bsf	Com_DIR,3
	return	
Cam_B3_DATO	bcf	Com_DATO,3
	btfs	LCD_Con,1
	return	
	bsf	Com_DATO,3
	return	
Cam_B4	btfs	LCD_Con,0
	goto	Cam_B4_DATO
Cam_B4_DIR	bcf	Com_DIR,4
	btfs	LCD_Con,1
	return	
	bsf	Com_DIR,4
	return	
Cam_B4_DATO	bcf	Com_DATO,4
	btfs	LCD_Con,1
	return	
	bsf	Com_DATO,4
	return	
Cam_B5	btfs	LCD_Con,0
	goto	Cam_B5_DATO
Cam_B5_DIR	bcf	Com_DIR,5
	btfs	LCD_Con,1
	return	
	bsf	Com_DIR,5
	return	
Cam_B5_DATO	bcf	Com_DATO,5
	btfs	LCD_Con,1
	return	
	bsf	Com_DATO,5
	return	
Cam_B6	btfs	LCD_Con,0
	goto	Cam_B6_DATO

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos, basado en un protocolo propio"

```

Cam_B6_DIR
    bcf    Com_DIR,6
    btfss  LCD_Con,1
    return
    bsf    Com_DIR,6
    return
Cam_B6_DATO
    bcf    Com_DATO,6
    btfss  LCD_Con,1
    return
    bsf    Com_DATO,6
    return

Cam_B7
    btfss  LCD_Con,0
    goto   Cam_B7_DATO
Cam_B7_DIR
    bcf    Com_DIR,7
    btfss  LCD_Con,1
    return
    bsf    Com_DIR,7
    return
Cam_B7_DATO
    bcf    Com_DATO,7
    btfss  LCD_Con,1
    return
    bsf    Com_DATO,7
    return

;
; Subrutina "Fin_Teclado" -----
;
Fin_Teclado_T
    call   LCD_Actualiza
    call   Teclado_EsperaDejePulsar
    bcf    INTCON,INTF
    retfie

;
;
;-----Subrutinas para el LCD-----
;

LCD_Actualiza
    call   LCD_Borra
    movf   LCD_Con,W
    movwf  LCD_Con_Aux
Visualiza_DIR
    call   LCD_Linea1
    movlw  'D'
    call   LCD_Character
    movlw  'I'
    call   LCD_Character
    movlw  'R'
    call   LCD_Character
    movlw  '1'
    call   LCD_Character
    bsf    LCD_Con,0
    call   Visualiza_Contenido
    bcf    LCD_Con,0

Visualiza_DATO
    call   LCD_Linea2
    movlw  'D'
    call   LCD_Character
    movlw  'A'
    call   LCD_Character
    movlw  'T'
    call   LCD_Character
    movlw  'O'

```

Elaboración del proyecto “Diseño de un sistema electrónico de envío y recepción de datos, basado en un protocolo propio”

```
call LCD_Caracter
call Visualiza_Contenido
return

Visualiza_Contenido
    btfscl LCD_Con,0
    goto Viz_DIR
    movf Com_DATO,W
    movwf Com_AUX
    goto Vizualiza_en_LCD

Viz_DIR
    movf Com_DIR,W
    movwf Com_AUX

Vizualiza_en_LCD
    movlw ''
    call LCD_Caracter
    movlw '<'
    call LCD_Caracter

    movlw '0'
    btfscl Com_AUX,7
    movlw '1'
    call LCD_Caracter

    movlw '0'
    btfscl Com_AUX,6
    movlw '1'
    call LCD_Caracter

    movlw '0'
    btfscl Com_AUX,5
    movlw '1'
    call LCD_Caracter

    movlw '0'
    btfscl Com_AUX,4
    movlw '1'
    call LCD_Caracter

    movlw '0'
    btfscl Com_AUX,3
    movlw '1'
    call LCD_Caracter

    movlw '0'
    btfscl Com_AUX,2
    movlw '1'
    call LCD_Caracter

    movlw '0'
    btfscl Com_AUX,1
    movlw '1'
    call LCD_Caracter

    movlw '0'
    btfscl Com_AUX,0
    movlw '1'
    call LCD_Caracter

    movlw '>'
    call LCD_Caracter

    movf LCD_Con_Aux,W
    movwf LCD_Con

return
;
```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos, basado en un protocolo propio"

```
;  
;-----Subrutinas para el Envío de Datos-----  
;  
Envia_Datos  
    movf    Com_DIR,W  
    call   PCF8574_Escribe  
    bsf    PORTB,2  
    bsf    PORTB,3  
    call   Retardo_100ms  
    bcf    PORTB,3  
  
    movf    Com_DATO,W  
    call   PCF8574_Escribe  
    bcf    PORTB,2  
    bsf    PORTB,3  
    call   Retardo_100ms  
    bcf    PORTB,3  
  
    Return  
  
    INCLUDE <PCF8574_Teclado.INC>  
    INCLUDE <BUS_I2C.INC>  
    INCLUDE <PCF8574.INC>  
    INCLUDE <RETARDOS.INC>  
    INCLUDE <LCD_4BIT.INC>  
    END
```

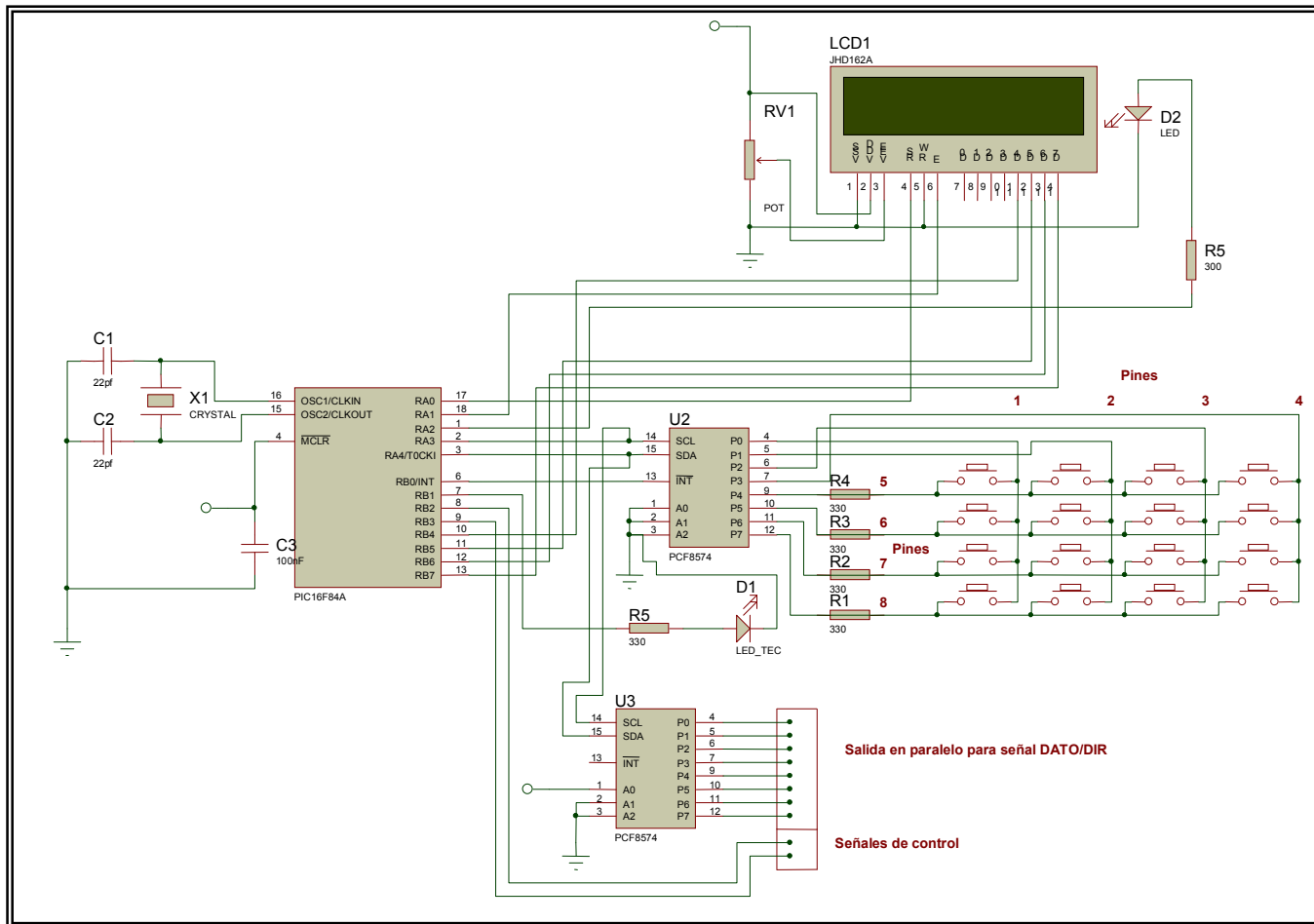


Fig. II.2.15 Diagrama electrónico (completo) de la etapa **Adquisición de Datos** (es importante aclarar que la forma de las teclas que se utilizan para este proyecto, se especifican en el Capítulo IV: Construcción y Pruebas del sistema)

II.3.- Sincronización

En esta etapa se desarrollan las partes "Sensado de frecuencia" y "CLK" (2 y 3 respectivamente) que componen al sistema de comunicación indicados en la Fig. I.4.1.

La etapa de "Sensado de frecuencia" está diseñada electrónicamente mientras que la etapa "CLK" es parte de la programación del microcontrolador principal en el sistema de comunicación, es decir, en la primera se analiza teóricamente y posteriormente se diseña con dispositivos electrónicos para poder elaborar el prototipo, mientras que en la segunda parte se utiliza la señal obtenida (en la etapa anterior) para la manipulación de la misma por medio de un lenguaje de programación para el microcontrolador, en esta etapa se elabora la subrutina necesaria para el programa general.

Sensado de frecuencia

El Sensado de la frecuencia es necesario para poder tener la sincronización entre dispositivos en la transmisión de datos como se muestra en la Fig. II.3.1

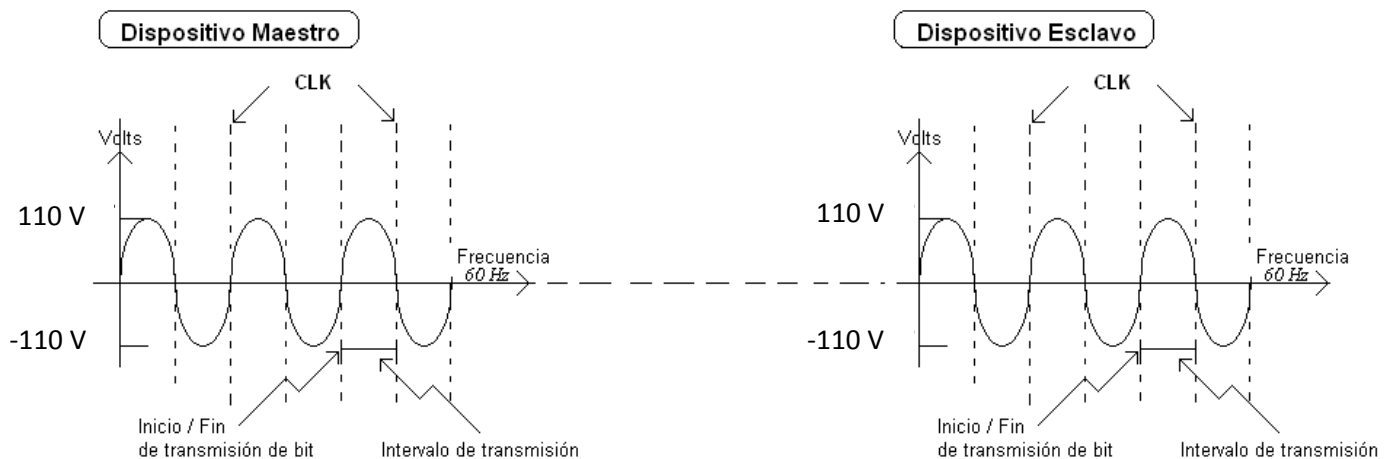


Fig. II.3.1 La señal de CLK debe ser la misma en el dispositivo emisor como en el dispositivo receptor.

La señal que se utiliza para la transmisión de datos es de 110VCA a 60Hz (Ver Fig. II.3.2), analizando la parte de la señal que sirve como CLK es el intervalo de transición entre ciclos, sin importar si éste se encuentra en la parte del semiciclo positivo o en la parte del semiciclo negativo, ya que en la transmisión se utilizan ambos.

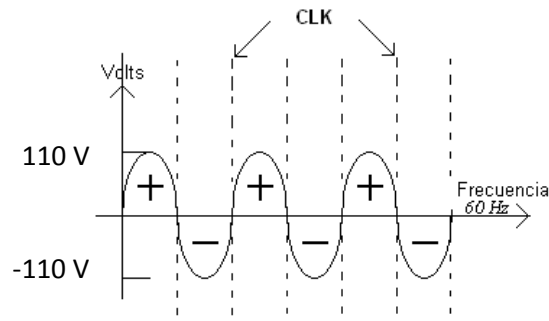


Fig. II.3.2 Onda senoidal 12VCA.

Si la señal de transmisión es de 60 ciclos/seg, y el dispositivo necesita cada intersección entre el voltaje igual a cero, por lo que se tiene que el sensado de frecuencia es de 120 muestras por segundo, además la señal a entregar al CLK es de corriente directa ya que esta etapa es interna al microcontrolador.

El planteamiento electrónico utilizado que satisface las características anteriores es el siguiente:

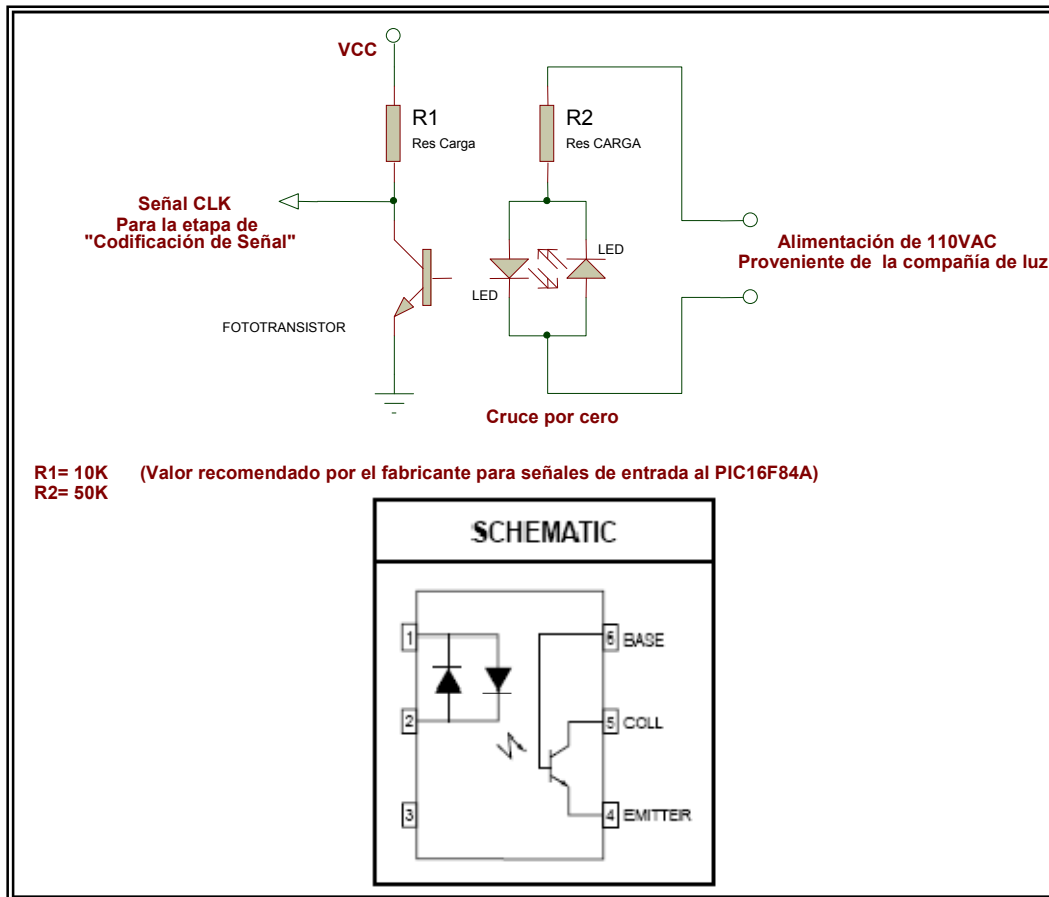


Fig. II.3.3 Diagrama electrónico que se utiliza en el sistema de comunicación para el sensado de frecuencia.

En la Fig. II.3.3, se muestra un diagrama electrónico para la realización de esta etapa, el cual está compuesto por un cruce por cero (para la etapa de

sensado en alterna) con una resistencia de carga R1 y un foto-transistor (para la etapa de sensado en directa) con una resistencia de carga en el colector R2.

El cruce por cero con salida a foto transistor se encuentra en el circuito integrado de 6 pines H11AA1 (Ver Anexo para sus especificaciones), la resistencia R1 se calcula para un voltaje de 110VCA y la resistencia R2 es de 10kΩ (es el valor recomendado por el fabricante) y el valor de VCC es de 5VCD ya que es el voltaje de alimentación para la parte electrónica.

```
;
;
;***** Librería "Sincronización.INC" *****
;
;
CBLOCK 0x0C
    ENDC

#DEFINE CLK          PORTA,0; Línea de la señal CLK.

; ZONA DE CÓDIGOS *****

Inicio      ORG      0
            bsf      STATUS,RP0
            bsf      CLK
            bcf      STATUS,RP0

Sincroniza_CLK          ;Sindroniza la señal CLK para q inicie en 1
            btfscl  CLK
            goto    Sincroniza_CLK

Inicia_0
            btfscl  CLK
            goto    Inicia_0
            Return

Inicia_TRX          ;Sincroniza la señal para que inicie la transferencia
            btfscl  CLK          ;en estado bajo (la periodo mas larga en tiempo de la
            goto    Inicia_TRX  ;señal CLK
            return

Termina_TRX        ;Sincroniza la señal para que termine el periodo
            btfscl  CLK          ;largo en el que se transmite la señal, para no
            goto    Termina_TRX ;no enviar otro bit en el mismo periodo
            return

            return
```

II.4.- Interfaz de envío

Esta parte del proyecto desarrolla la interfaz de envío, para la transmisión de la información, lo cual es el objetivo planteado del proyecto en desarrollo, se estableció un protocolo de comunicación con características previas (definidas en el capítulo I) donde se establece que la comunicación será de 110VCA con un bus de dos hilos (sin alimentación), con esa característica parte el análisis.

El medio de transmisión es en alterna ya que se pueden manejar grandes distancias para su transmisión y la pérdida de energía es mínima, la forma de transmitir es mediante “pulsos” de voltaje (en el caso del voltaje en alterna es mediante ciclos, como se explica en la Fig. II.3.1), para ello se hace la interrupción de voltaje por cada semiciclo de la onda de voltaje con la que se transmite, dicha interrupción se encuentra dentro de esta etapa (Interfaz de envío) en una parte modular que se denomina Escritura maestro, y su funcionamiento es acoplar al sistema electrónico de la etapa Codificación de señal que maneja un voltaje de directa con la señal que transmite datos con características de 110VAC ver Fig. II.4.2.

Es importante aclarar que para que esta parte de Escritura maestro funcione, es necesario que su contraparte en el dispositivo esclavo (Escritura esclavo) siempre este en “1” lógico o activada, ya que eso permite que la señal de envío tenga el complemento de referencia, necesario para una caída de voltaje.

La parte Escritura maestro, está compuesta por un dispositivo interruptor de señal en alterna controlado con un dispositivo acoplador con señales de voltaje en directa.

El dispositivo que hace la función de interrumpir la señal de envío es el triac (Ver Fig. II.4.1), éste se comporta como un interruptor de corriente alterna (las terminales se conectan en A1 y A2), y cuyo control depende de la corriente que fluye por su terminal G.

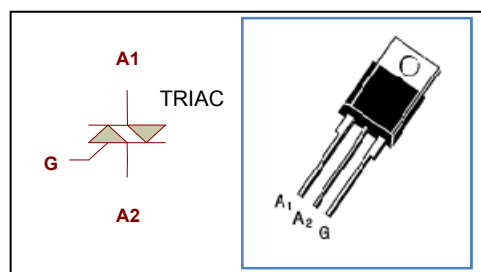


Fig. II.4.1 Diagrama electrónico del triac (el modelo que se usa para el proyecto es el BTA08-600BW).

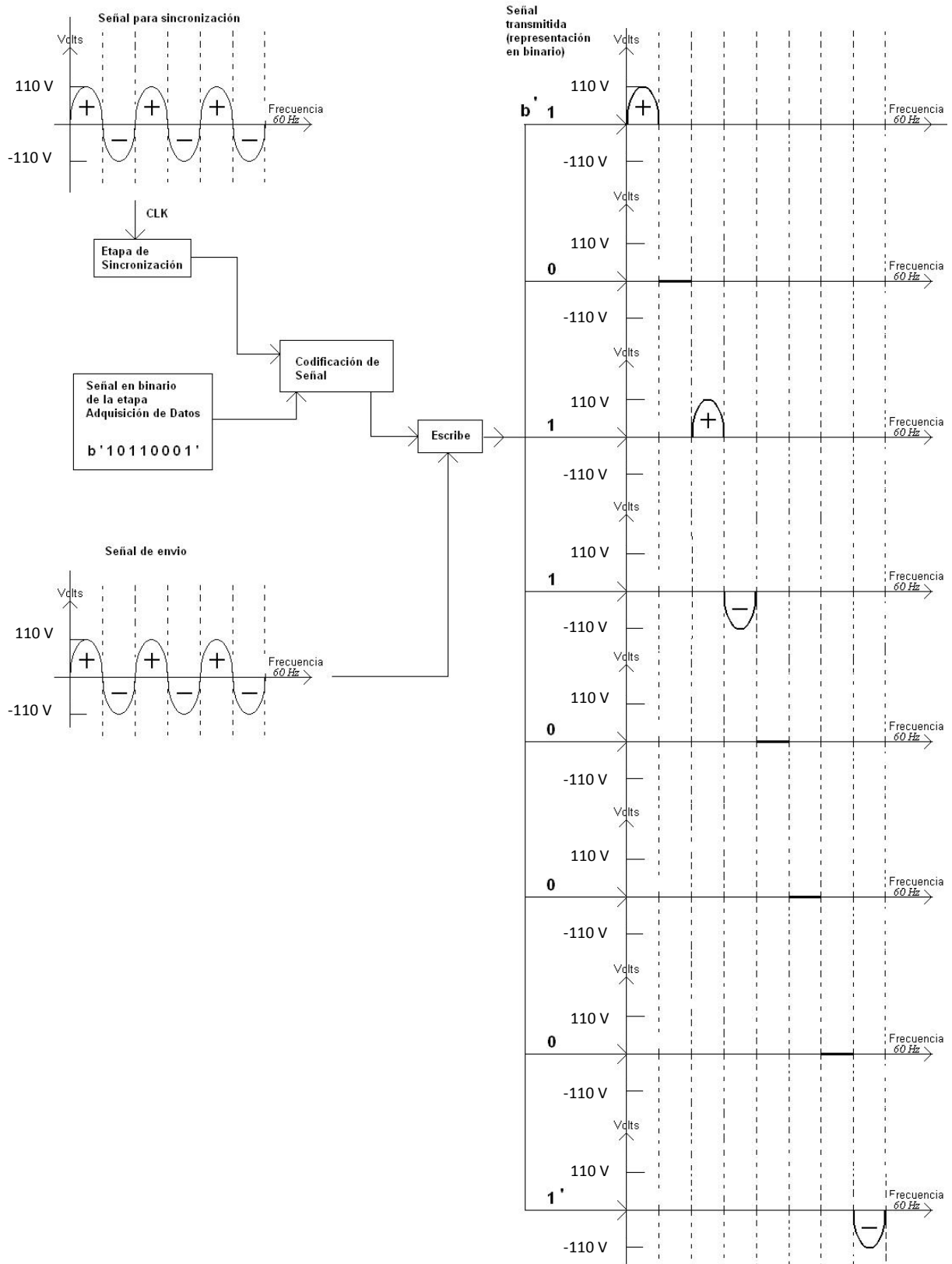


Fig. II.4.2 Diagrama transmisión de datos por parte del dispositivo maestro en su etapa Escribe.

La terminal G del triac debe ser controlada por la etapa Codificación de Señal y el dispositivo electrónico que acopla dicha señal con la de control es un opto-acoplador con componente de salida fototriac (como se muestra en la Fig. II.4.3), el cual su función es proteger de voltajes mayores o diferentes a los dispositivos electrónicos de control.

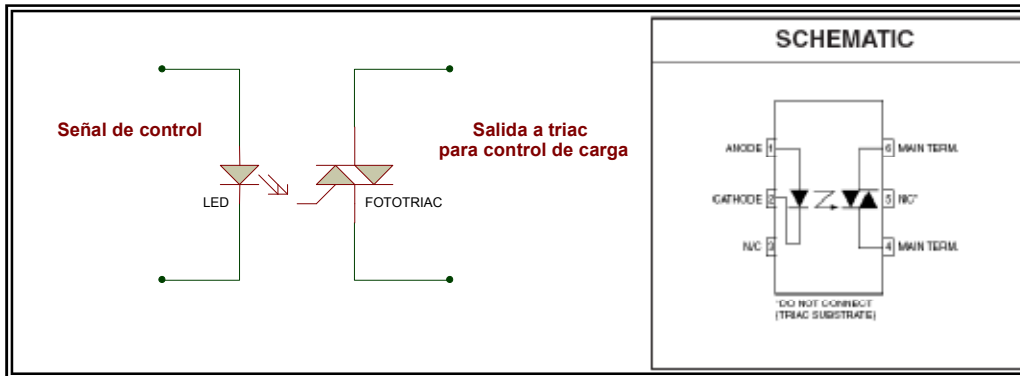


Fig. II.4.3 Diagrama electrónico del opto-acoplador con dispositivo de salida un fototriac (el modelo que se usa para el proyecto es el MOC3011).

El diseño electrónico completo para la transmisión de datos es el que se muestra en la Fig. II.4.4.

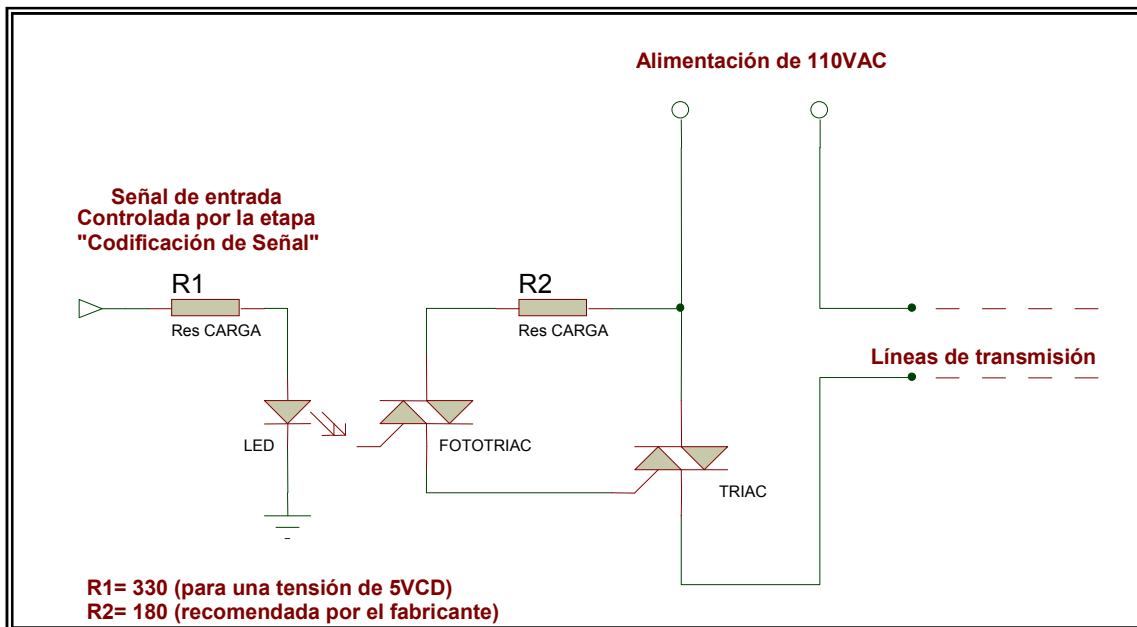


Fig. II.4.4 Diagrama electrónico de la parte Escritura para la transmisión de datos

La otra parte modular que compone la etapa Interfaz envío es Lectura maestro, es en ésta donde se recibe la respuesta proporcionada por el dispositivo esclavo (o por otro dispositivo maestro, lo cual se insta como propuesta para estudio y mejoramiento de este trabajo a futuras propuestas de desarrollo) para ello se utiliza la misma señal de envío.

La manera en la cual la parte modular Lectura maestro funciona es como se muestra en la Fig. II.4.5.

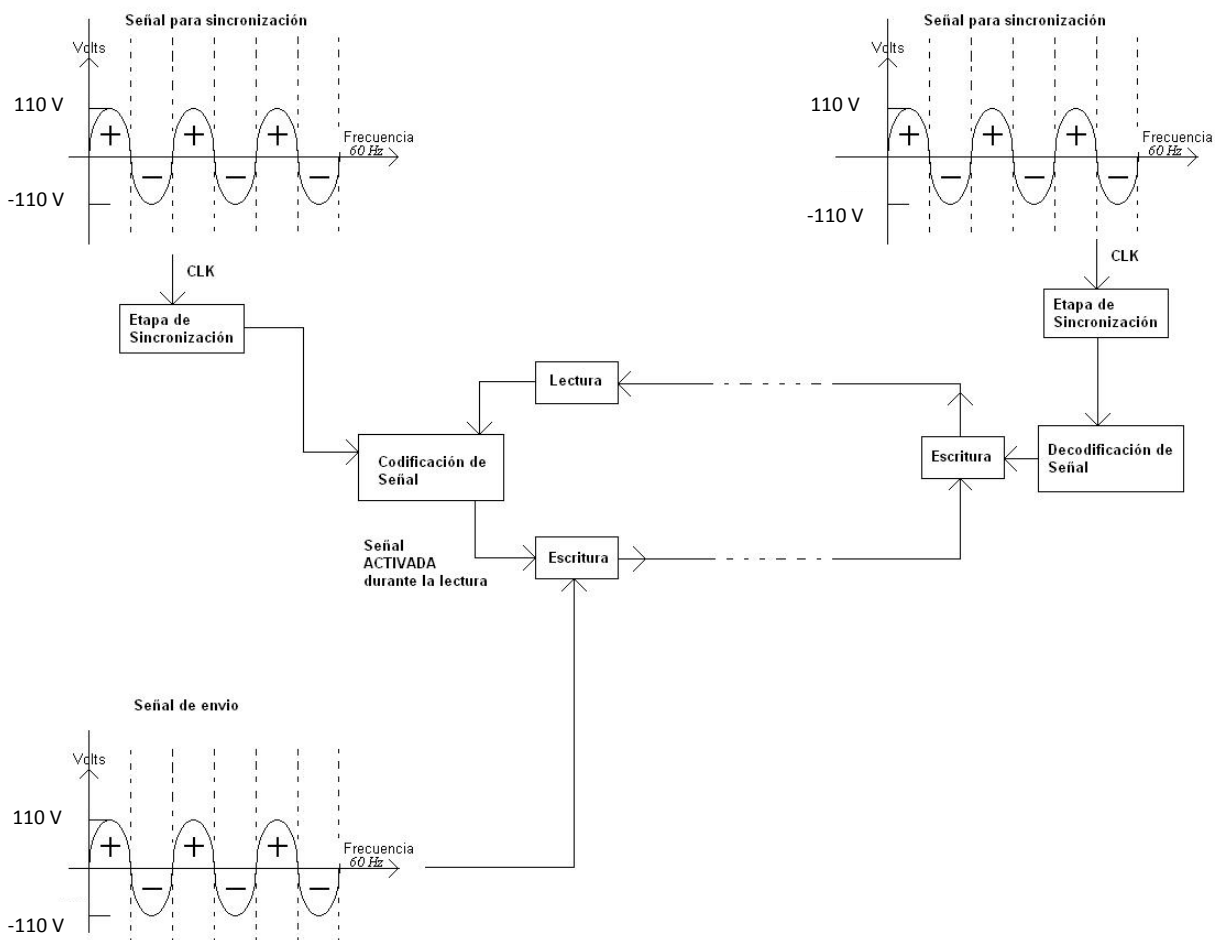


Fig. II.4.5 Diagrama de la recepción de datos por parte del dispositivo maestro en su etapa Lectura.

El funcionamiento de la parte modular Lectura depende de la señal de envío que el mismo dispositivo maestro maneja, es por ello que la etapa Codificación de señal siempre mantiene activada la parte modular Escritura para permitir que la señal pase íntegramente y quien la manipule sea el dispositivo esclavo (como se muestra en la Fig. II.4.5).

La parte Lectura de la etapa Interfaz de envío está compuesta por el mismo circuito que se utiliza para la sincronización (ver etapa Sincronización) mientras una señal (la señal de sincronización) indica a que determinado lapso de tiempo se debe leer o escribir un bit, la otra en paralelo controlada por el dispositivo esclavo envía por medio de cada mitad de ciclo de la señal, la representación del dato en binario (ver Fig. I.3.2)

En la Fig. II.4.6 se muestra el diagrama electrónico correspondiente a la etapa Interfaz de envío con sus dos módulos que la componen, Escritura y Lectura.

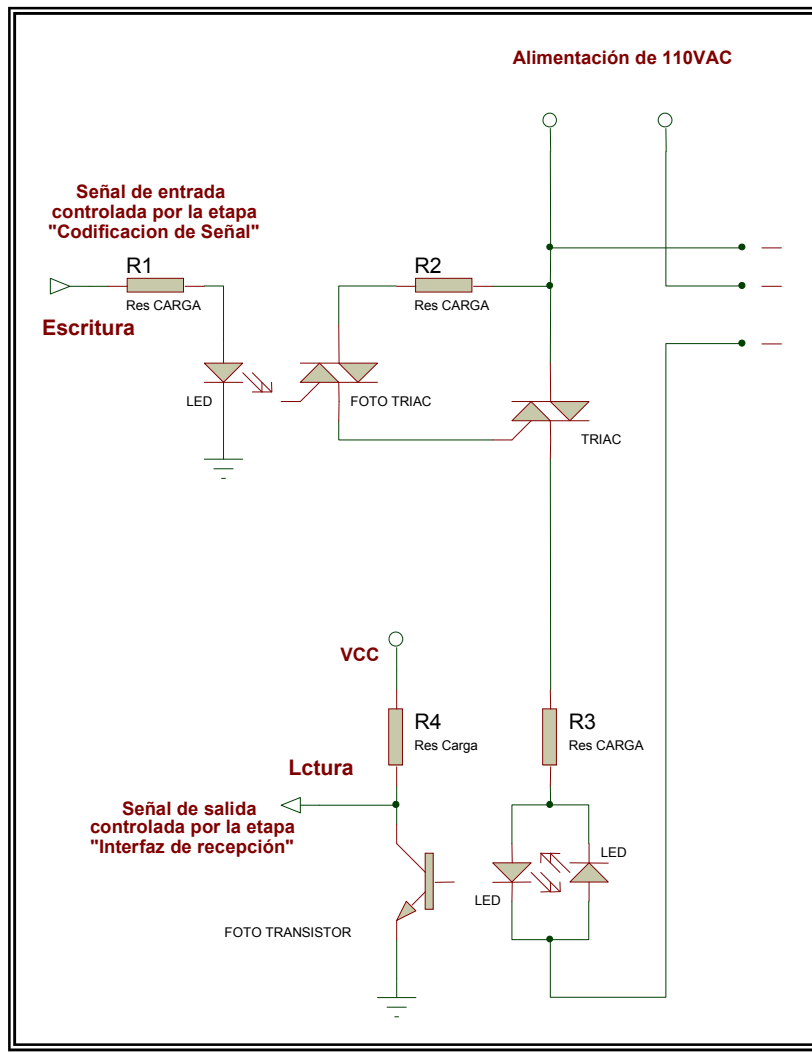


Fig. II.4.6 Diagrama electrónico completo de la parte Escritura para la transmisión de datos.

II.5.- Codificación de Señal

En esta parte del capítulo se hace el control principal del dispositivo maestro, y es por ello que todas las etapas anteriores llegan a esta parte del dispositivo maestro, la etapa Adquisición de Datos recibe la información que se va a transmitir y la envía a esta etapa de control, es aquí donde se procesa la información recibida y la armoniza con la etapa de Sincronización para poderla transmitir de acuerdo al protocolo establecido y finalmente envía cada uno de los bits a la etapa Interfaz de envío.

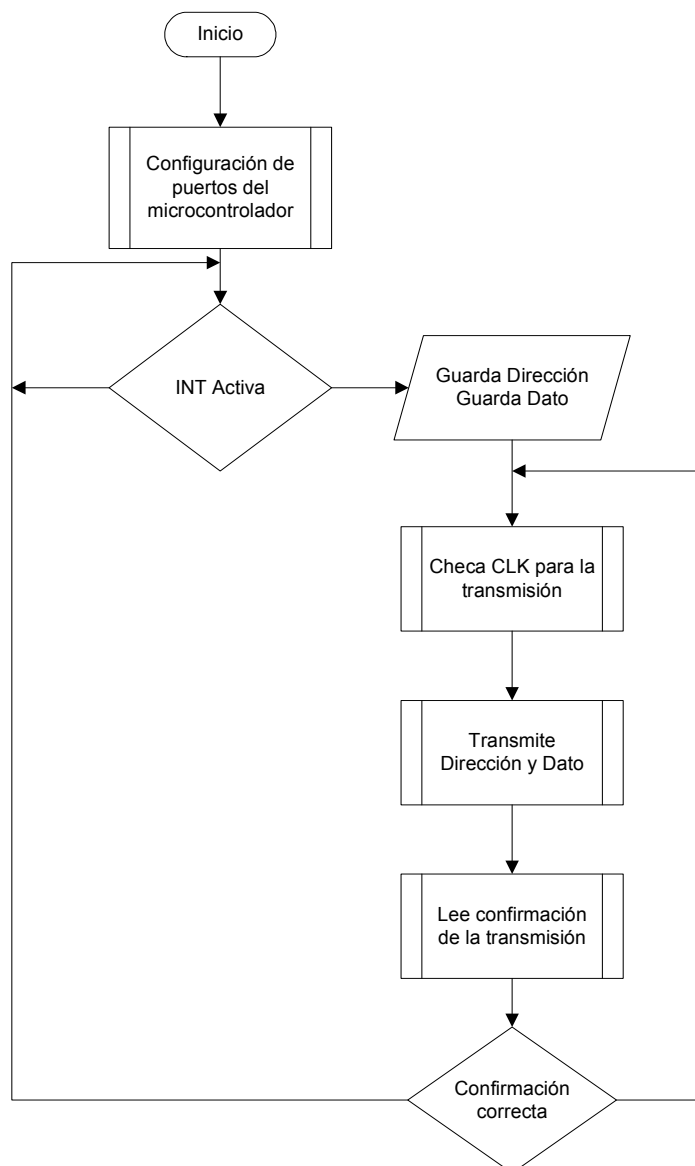


Fig. II.5.1 Diagrama de flujo para la programación del sistema principal del dispositivo maestro.

El diagrama de flujo a seguir para la programación del microcontrolador de esta etapa se muestra en la Fig. II.5.1

```

;
;
;***** Librería "Tr_Datos.ASM" *****
;
;
; ZONA DE DATOS *****
;
;   _CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
LIST      P=16F84A
INCLUDE  <P16F84A.INC>
;
;   CBLOCK 0x0C
;   DATO
;   DIRECCION
;   ENDC
;
;#DEFINE SINC          PORTA,0; Línea de Sincronizacion
;#DEFINE TRX          PORTA,1; Línea de CONTROL para la Transferencia de Datos
;#DEFINE DATO_DIR     PORTA,3; Línea de Dato/Dir
;#DEFINE Trx_ACT      PORTA,4; Línea de Transferencia Activa
;
; ZONA DE CÓDIGOS *****
;
; Inicio
;
;   bsf    STATUS,RP0
;   bsf    DATO_DIR
;   bsf    Trx_ACT
;   bsf    SINC
;   bcf    TRX
;   bcf    STATUS,RP0
;   clrf   PORTB
;   clrf   PORTA
;   call   Retardo_2s
;
; Principal
;
;   btfss  Trx_ACT
;   goto  Principal
;
;   movf   PORTA,W
;   movwf  DIRECCION
;
; Regresa_01
;   btfs  Trx_ACT
;   goto  Regresa_01
;
; Regresa_02
;   btfss  Trx_ACT
;   goto  Regresa_02
;
;   movf   PORTA,W
;   movwf  DATO
;
;   call   Envia_TRX
;   goto  Principal
;
; Envia_TRX
;
;   call   Sincroniza_CLK
;
; TRX_TA
;
;   call   Inicia_TRX
;   bsf    TRX
;   call   Termina_TRX
;   bcf    TRX
;
;   ;Indica al dispositivo esclavo que se inicia la transferencia
;   ;enviando una señal de 1 lógico al dispositivo
;
; TRX_CT
;
;   call   Inicia_TRX
;   bcf    TRX
;   call   Termina_TRX
;   bcf    TRX
;
;   ;Comunicacion unidireccional
;   ;enviando dos señales de 0 lógico al dispositivo
;
; TRX_CT_1
;
;   call   Inicia_TRX
;   bcf    TRX

```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio"

call	Termina_TRX	
bcf	TRX	
TRX_TA_0		;Indica al dispositivo esclavo que se inicia la transferencia ;enviando una señal de 1 lógico al dispositivo
call	Inicia_TRX	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
TRX_DD		;Indica al dispositivo esclavo que es una DIRECCION la que envía ;enviando una señal de 1 lógico al dispositivo
call	Inicia_TRX	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
DIRECCION_B7		
call	Inicia_TRX	
btfs	DIRECCION,7	
goto	C_DIR7_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B6	
C_DIR7_0		
bcf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B6	
DIRECCION_B6		
call	Inicia_TRX	
btfs	DIRECCION,6	
goto	C_DIR6_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B5	
C_DIR6_0		
bcf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B5	
DIRECCION_B5		
call	Inicia_TRX	
btfs	DIRECCION,5	
goto	C_DIR5_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B4	
C_DIR5_0		
bcf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B4	
DIRECCION_B4		
call	Inicia_TRX	
btfs	DIRECCION,4	
goto	C_DIR4_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B3	
C_DIR4_0		
bcf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B3	
DIRECCION_B3		
call	Inicia_TRX	
btfs	DIRECCION,3	

goto	C_DIR3_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B2	
C_DIR3_0		
bcf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B2	
DIRECCION_B2		
call	Inicia_TRX	
btfs	DIRECCION,2	
goto	C_DIR2_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B1	
C_DIR2_0		
bcf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B1	
DIRECCION_B1		
call	Inicia_TRX	
btfs	DIRECCION,1	
goto	C_DIR1_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B0	
C_DIR1_0		
bcf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DIRECCION_B0	
DIRECCION_B0		
call	Inicia_TRX	
btfs	DIRECCION,0	
goto	C_DIR0_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	TRX_TA_2	
C_DIR0_0		
bcf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	TRX_TA_2	
TRX_TA_2		
call	Inicia_TRX	;Indica al dispositivo esclavo que se inicia la transferencia
bsf	TRX	;enviando una señal de 1 lógico al dispositivo
call	Termina_TRX	
bcf	TRX	
TRX_DD_2		
call	Inicia_TRX	;Indica al dispositivo esclavo que es un DATO lo que envía
bcf	TRX	;enviando una señal de 1 lógico al dispositivo
call	Termina_TRX	
bcf	TRX	
DATO_B7		
call	Inicia_TRX	
btfs	DATO,7	
goto	C_DATO7_0	
bsf	TRX	
call	Termina_TRX	
bcf	TRX	
goto	DATO_B6	
C_DATO7_0		
bcf	TRX	

call	Termina_TRX
bcf	TRX
goto	DATO_B6
DATO_B6	
call	Inicia_TRX
btfs	DATO,6
goto	C_DATO6_0
bsf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B5
C_DATO6_0	
bcf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B5
DATO_B5	
call	Inicia_TRX
btfs	DATO,5
goto	C_DATO5_0
bsf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B4
C_DATO5_0	
bcf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B4
DATO_B4	
call	Inicia_TRX
btfs	DATO,4
goto	C_DATO4_0
bsf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B3
C_DATO4_0	
bcf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B3
DATO_B3	
call	Inicia_TRX
btfs	DATO,3
goto	C_DATO3_0
bsf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B2
C_DATO3_0	
bcf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B2
DATO_B2	
call	Inicia_TRX
btfs	DATO,2
goto	C_DATO2_0
bsf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B1
C_DATO2_0	
bcf	TRX
call	Termina_TRX
bcf	TRX
goto	DATO_B1
DATO_B1	

```
call      Inicia_TRX
btfss    DATO,1
goto     C_DATO1_0
bsf      TRX
call     Termina_TRX
bcf      TRX
goto     DATO_B0
C_DATO1_0
bcf      TRX
call     Termina_TRX
bcf      TRX
goto     DATO_B0

DATO_B0
call     Inicia_TRX
btfss    DATO,0
goto     C_DATO0_0
bsf      TRX
call     Termina_TRX
bcf      TRX
goto     TRX_Fin
C_DATO0_0
bcf      TRX
call     Termina_TRX
bcf      TRX
goto     TRX_Fin

TRX_Fin
bcf      TRX
goto     Principal

INCLUDE <RETARDOS.INC>
INCLUDE <Sincronizacion.INC>

END
```

Para la elaboración del circuito electrónico de esta etapa del proyecto, se incorporan, como se mencionó anteriormente, todas las etapas anteriores (Adquisición de Datos, Sincronización e Interfaz de envío) y es por ello que el circuito consta del diagrama Sensado de frecuencia (ver Fig. II.3.3), llega aquí también la señal en paralelo que envía la etapa de Adquisición de Datos (ver Fig. II.2.13, en la parte “Salida de Control”) y ésta a su vez controla el circuito Interfaz de envío y todas ellas están controladas por un microcontrolador PIC16F84A (se utiliza este microcontrolador ya que no se necesita mayor capacidad de memoria para el caso práctico de este proyecto, pero se recomienda ampliar la capacidad de los dispositivos maestros con un microcontrolador de mayor potencia como el 16F877)

CAPÍTULO. III.- DISEÑO DEL DISPOSITIVO ELECTRÓNICO ESCLAVO

III.1.- Introducción

La elaboración del dispositivo electrónico esclavo es más sencilla que la del dispositivo electrónico maestro, ya que la primera está diseñada para hacer un multicontrol de dispositivos esclavos, además de que es la que interactúa con el medio que introduce los datos, que en este proyecto es un usuario, y por ello debe ser de uso sencillo y amigable, todo lo contrario con el dispositivo esclavo ya que éste solo envía un dato como respuesta sin direccionar y el dato que procesa (enviado por el dispositivo maestro) no tiene un procesamiento especial para su entendimiento ya que solo expresa el dato recibido en binario, para un dispositivo electrónico llamado Actuador (independiente de este proyecto).

III.2.- Interfaz de recepción

La Interfaz de recepción está compuesta, al igual que el dispositivo maestro, de dos partes modulares una denominada Lectura y la otra Escritura.

La etapa modular denominada Lectura, al igual que su contra parte en el dispositivo maestro, se utiliza un opto-acoplador electrónico que detecte la señal que se envía desde el dispositivo maestro en la línea de transmisión de alterna. (Ver Fig. III.2.1)

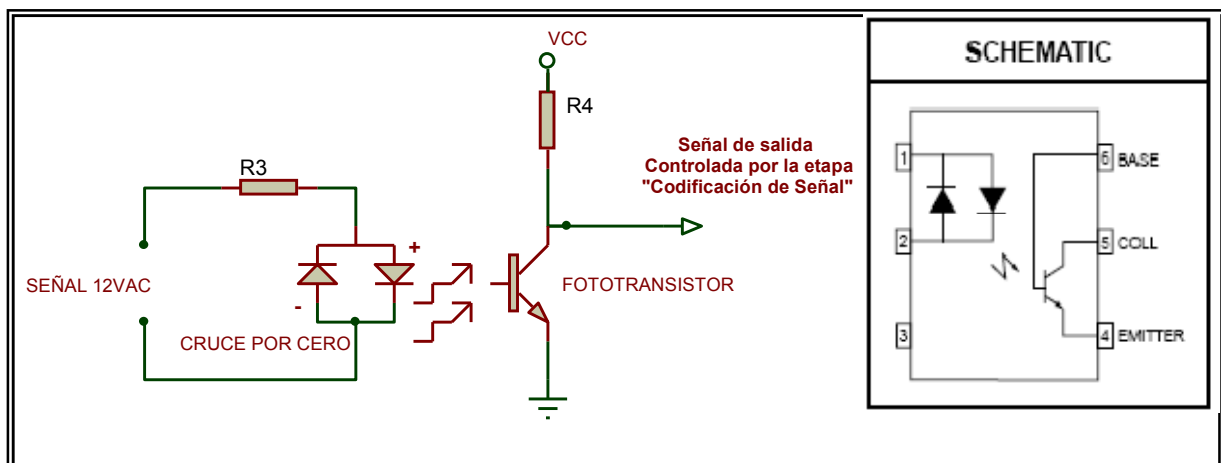


Fig. III.2.1 Diagrama electrónico del opto-acoplador que utiliza como dispositivo de salida un fototransistor.

El diagrama electrónico que realiza la comunicación completa tanto la interfaz de envío como la Interfaz de recepción se muestra en la siguiente Fig. III.2.2.

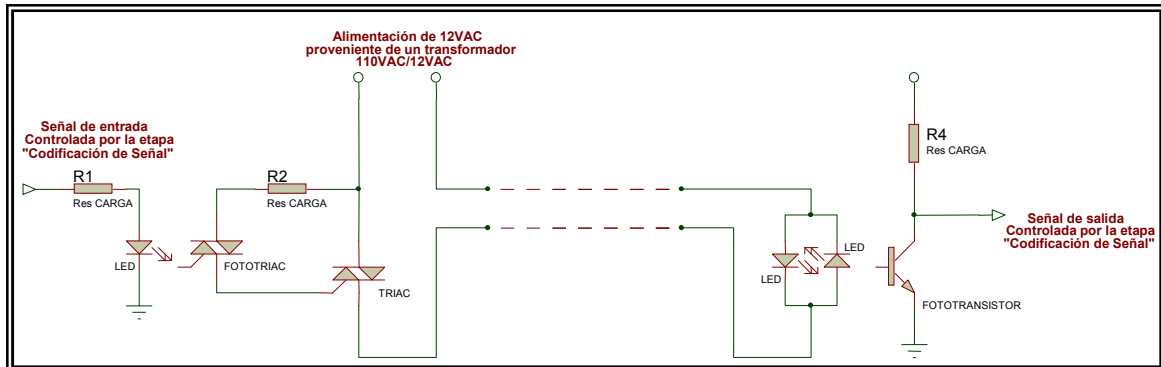


Fig. III.2.2 Diagrama electrónico para la transmisión de datos (R4= 10K, resistencia recomendada por el fabricante para señales de entrada al PIC16F84A).

La etapa completa “Interfaz recepción” del dispositivo esclavo es un espejo de la etapa “Interfaz envío” del dispositivo esclavo, es decir, mientras el dispositivo maestro escribe el dispositivo esclavo lee y viceversa, es importante recalcar que mientras el dispositivo está activo en forma “Lectura”, la parte de “Escritura” debe estar activada para dejar pasar la señal completa de envío, ya que el dispositivo Lectura esta en paralelo al de Escritura.

El circuito electrónico que cumple con toda la etapa Interfaz de recepción se muestra en la Fig. III.2.3., la figura se muestra con el complemento de la parte del dispositivo maestro para tener un panorama claro del ciclo completo de la comunicación.

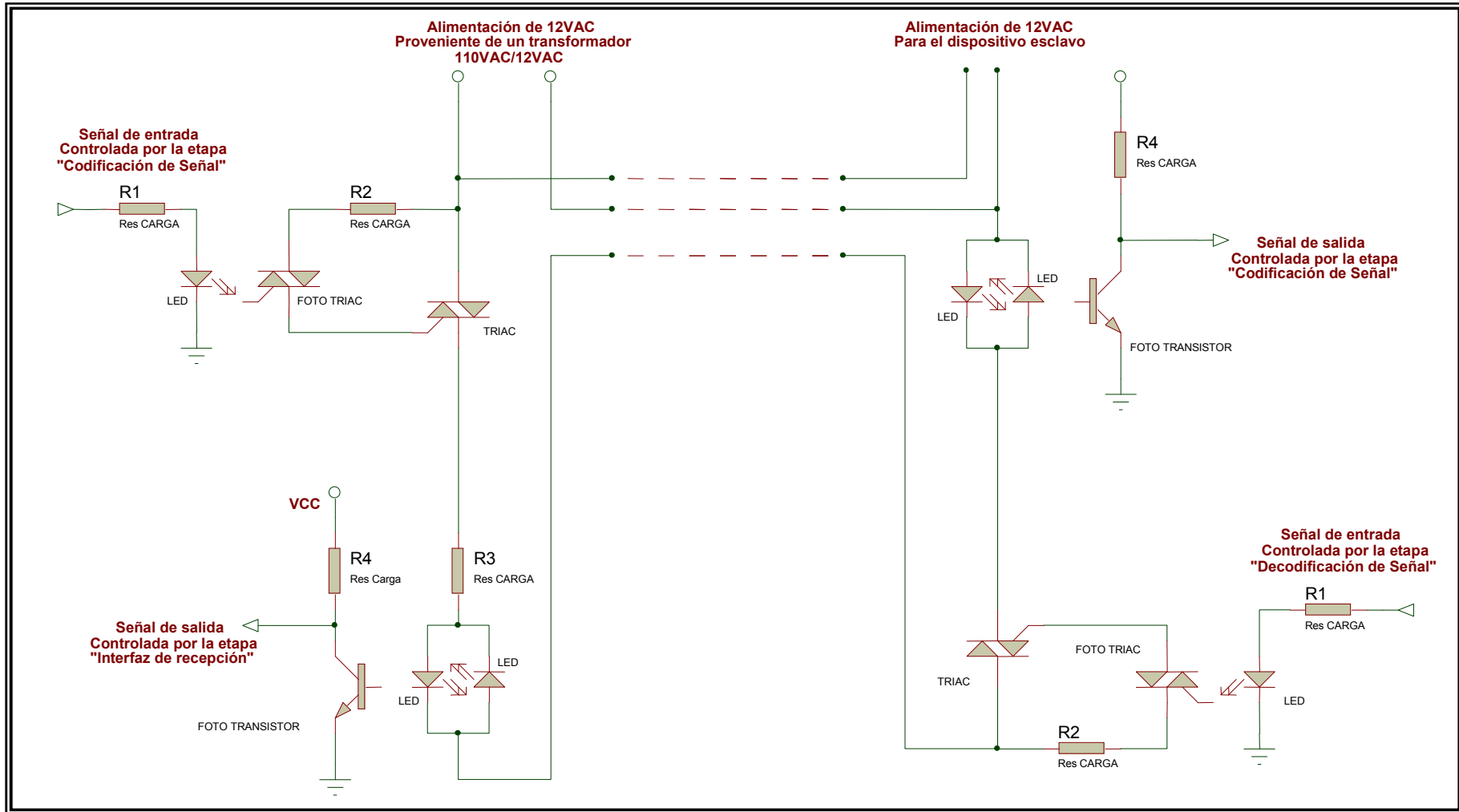


Fig. III.2.3 Diagrama electrónico bidireccional para la transmisión de datos con alimentación de 12VAC para dispositivos esclavos.

III.3.- Sincronización

Esta parte del dispositivo esclavo es idéntica a la del dispositivo maestro ya que es la dedicada a sincronizar la transmisión entre ambos dispositivos ver Fig. III.3.1.

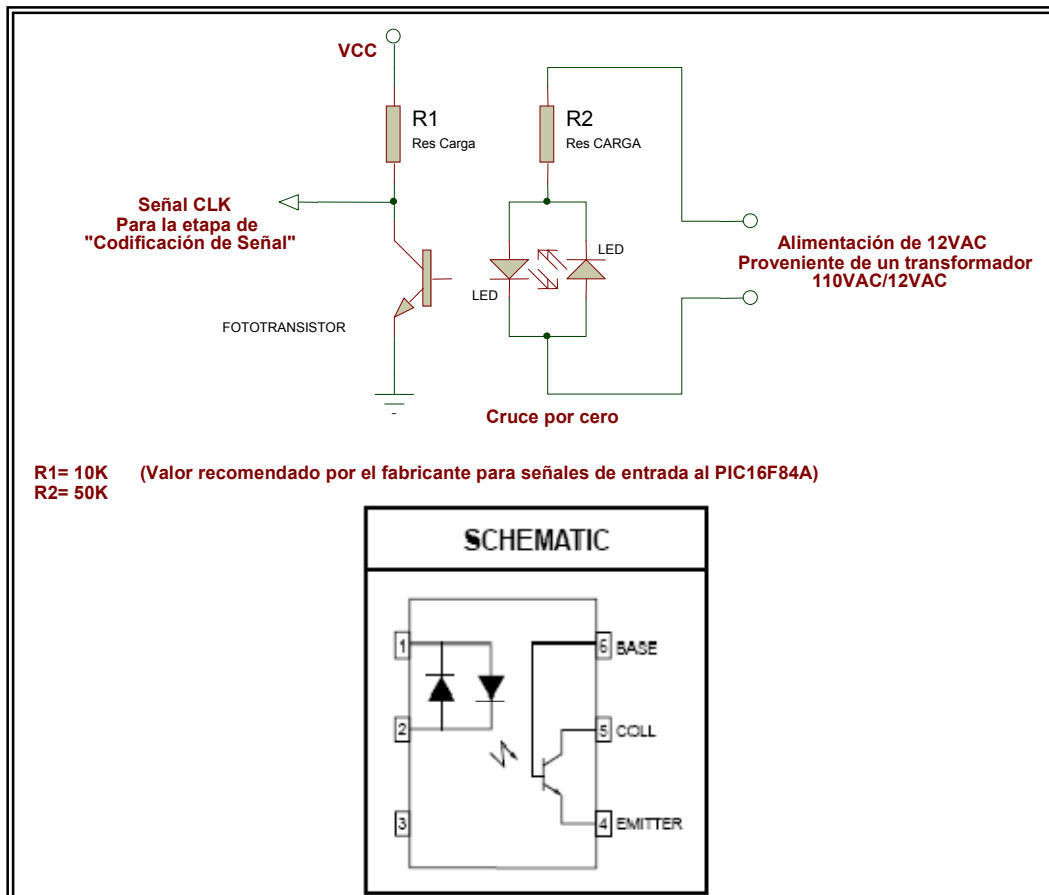


Fig. III.3.1 Diagrama electrónico bidireccional para la transmisión de datos con alimentación de 12VAC para dispositivos esclavos

III.4.- Direcccionamiento

El sistema de comunicación es multidireccional, es por ello que el dispositivo esclavo tiene que tener incluida una sección en el diseño para el direccionamiento del dispositivo, esta forma de ingreso de dirección es manual para mayor alcance del dispositivo, ya que así se garantiza un mayor manejo en la adaptabilidad de aplicaciones, es decir, un dispositivo podría tener la misma dirección que otros dispositivos similares, aplicándolo al ejemplo que se viene manejando desde el inicio, los dispositivos esclavos podrían ser el encendido y apagado del alumbrado de un espacio determinado en una casa, por lo que el funcionamiento es idéntico ya que el direccionamiento que se ingresa a los dispositivos esclavos podría ser el mismo y así cada que se envíe una acción específica a esa dirección, todos los dispositivos esclavos que estén con esa dirección realizarán determinado trabajo, en la Fig. III.4.1 se muestra la selección manual de dirección para un dispositivo esclavo.

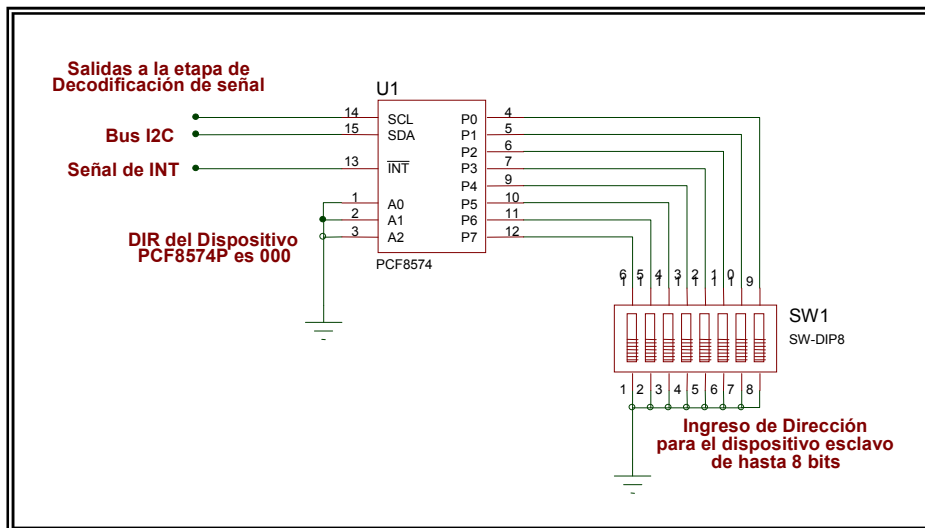


Fig. III.4.1 Diagrama electrónico para el direccionamiento del dispositivo esclavo por medio de una barra de ocho interruptores.

Con la barra de ocho interruptores se puede hacer una combinación binaria de hasta 255 direcciones diferentes.

```

;
;
;***** Librería "Direcc_Esclavo.INC" *****
;
;
; Utiliza dos circuitos integrados PCF8574P uno para la lectura de un array y otro que se utiliza como
; actuador para indicar el dato que se transmite, sin embargo en este caso se utiliza para mostrar la dirección que
; se selecciona para el dispositivo esclavo (la dirección se mostrará 5s para después regresar a su lectura de dato
; anterior
    
```

```
;
;
; Dos PCF8574P:
; - Uno como entrada, leyendo un array de 8 interruptores. Su pin A0 se conecta a masa.
; - Otro como salida, visualizando los datos en un array de diodos LEDs. Su pin A0 se
;   conecta a Vcc.
;
;
; Este programa lee los switches conectados al PCF8574 que actúa como entrada y su valor se
; visualiza en los LEDs conectados al PCF8574 de salida.
;
;
; ZONA DE DATOS *****
PCF8574_DireccionLectura EQU    b'01000001'
PCF8574_DireccionEscritura EQU  b'01000010'
;
;
CBLOCK
Dir_Esclavo
ENDC

; ZONA DE CÓDIGOS *****
;
;
; Lee los interruptores conectados al PCF8574 que actúa como entrada y el resultado lo visualiza
; en los diodos LEDs conectados al PCF8574 configurado como salida.

Lee_Direccion
    call    Retardo_20ms           ; Espera unos instante a que se estabilicen los
    call    PCF8574_Lee           ; niveles de tensión y lee la entrada.
    movwf  Dir_Esclavo            ; Complementa el dato leído porque los diodos
    comf   Dato,W                 ; se activan con nivel bajo, (ver esquema).
    call   PCF8574_Escribe        ; Lo visualiza en los diodos LEDs.
    call   Retardo_5s
    return
```

III.5.- Decodificación de señal

Es aquí donde se unen todas las etapas anteriores que componen al dispositivo esclavo hasta llegar con el actuador, quien realiza el proceso de control en ésta es el microcontrolador (PIC16f84A), primero recibe la señal transmitida de la etapa Interfaz de recepción, de acuerdo con el protocolo establecido y la sincronización de dispositivos con ayuda de la etapa Sincronización, para así decodificar la señal y obtener un dato concreto, el primer dato recibido por cualquier dispositivo es la dirección o direcciones (la forma de ver si más de una dirección, es mediante el protocolo de comunicación establecido en el Capítulo I), después hace una comparación del primer dato recibido de la línea de comunicación con el dato recibido de la etapa de Direccionamiento para establecer si es el dispositivo al que se desea controlar por medio del dispositivo maestro, para finalmente recibir la información a transmitir.

El diagrama a bloques del proceso que realiza esta etapa se muestra en la Fig. III.5.1.

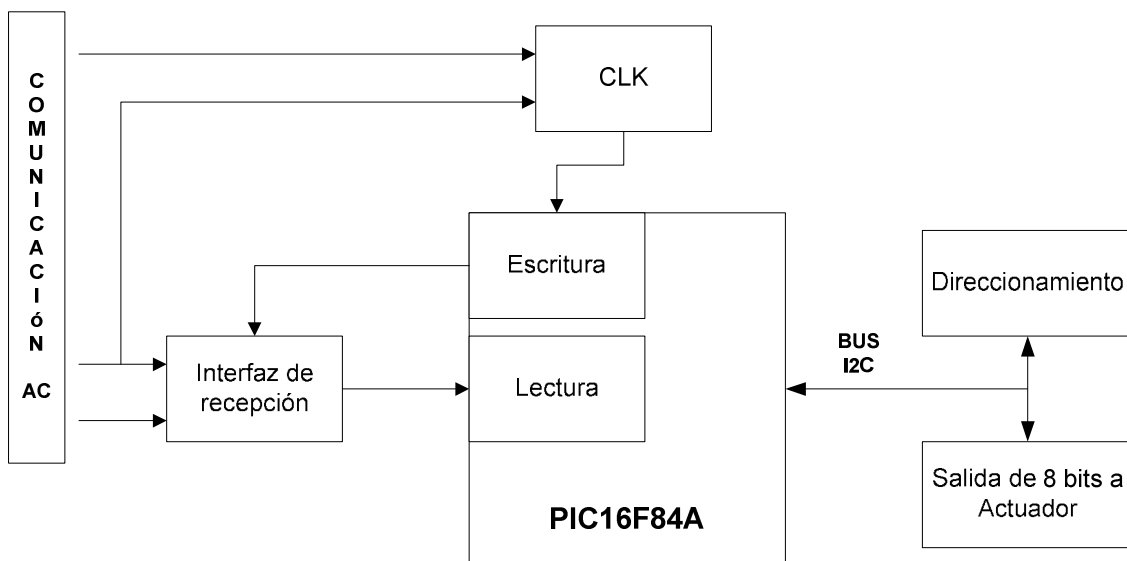


Fig. III.5.1 Diagrama a bloques de la etapa de Decodificación de Señal.

```

;
;
;***** Librería "RcX Datos.ASM" *****
;
;
; ZONA DE DATOS *****
;
;   _CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
;   LIST     P=16F84A
;   INCLUDE  <P16F84A.INC>
;
;   CBLOCK  0x0C
;   Guarda_DATO
;   Guarda_DIRECCION
;   ENDC
;
;#DEFINE SINC          PORTB,0; Línea de Sincronizacion
;#DEFINE TRX           PORTA,1; Línea de CONTROL para la Transferencia de Datos
;#DEFINE SCL           PORTA,3; CLK de bus I2C
;#DEFINE SDA           PORTA,4; Transmision de bus I2C
;
; ZONA DE CÓDIGOS *****
;
;   ORG      0
;
; Inicio
;   bsf      STATUS,RP0
;   bsf      SINC
;   bcf      TRX
;   bcf      STATUS,RP0
;   clrf    PORTB
;   clrf    PORTA
;   call     Retardo_2s
;
; Principal
;   btfss   TRX
;   goto    Principal
;
;   call    Sincroniza_CLK
;
; TRX_TA
;   call    Inicia_TRX
;   btfss   TRX
;   goto    Principal
;   call    Termina_TRX
;
; TRX_CT
;   call    Inicia_TRX
;   btfss   TRX
;   goto    Principal
;   call    Termina_TRX
;
; TRX_CT_1
;   call    Inicia_TRX
;   btfss   TRX
;   goto    Principal
;   call    Termina_TRX
;
; TRX_TA_0
;   call    Inicia_TRX
;   btfss   TRX
;   goto    Principal
;   call    Termina_TRX
;
; TRX_DD
;   call    Inicia_TRX
;   btfss   TRX
;   goto    DATO_B7
;   call    Termina_TRX
;
; DIRECCION_B7
;   call    Inicia_TRX
;   bcf     Guarda_DIRECCION,7
;   btfsc   TRX
;   bsf     Guarda_DIRECCION,7
;
;Indica al dispositivo esclavo que se inicia la transferencia
;enviando una señal de 1 lógico al dispositivo
;Comunicacion unidireccional
;enviando dos señales de 0 lógico al dispositivo
;Indica al dispositivo esclavo que se inicia la transferencia
;enviando una señal de 1 lógico al dispositivo
;Indica al dispositivo esclavo que es una DIRECCION la que envía
;enviando una señal de 1 lógico al dispositivo

```

Elaboración del proyecto “Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio”

call	Termina_TRX	
DIRECCION_B6		
call	Inicia_TRX	
bcf	Guarda_DIRECCION,6	
btfs	TRX	
bsf	Guarda_DIRECCION,6	
call	Termina_TRX	
DIRECCION_B5		
call	Inicia_TRX	
bcf	Guarda_DIRECCION,5	
btfs	TRX	
bsf	Guarda_DIRECCION,5	
call	Termina_TRX	
DIRECCION_B4		
call	Inicia_TRX	
bcf	Guarda_DIRECCION,4	
btfs	TRX	
bsf	Guarda_DIRECCION,4	
call	Termina_TRX	
DIRECCION_B3		
call	Inicia_TRX	
bcf	Guarda_DIRECCION,3	
btfs	TRX	
bsf	Guarda_DIRECCION,3	
call	Termina_TRX	
DIRECCION_B2		
call	Inicia_TRX	
bcf	Guarda_DIRECCION,2	
btfs	TRX	
bsf	Guarda_DIRECCION,2	
call	Termina_TRX	
DIRECCION_B1		
call	Inicia_TRX	
bcf	Guarda_DIRECCION,1	
btfs	TRX	
bsf	Guarda_DIRECCION,1	
call	Termina_TRX	
DIRECCION_B0		
call	Inicia_TRX	
bcf	Guarda_DIRECCION,0	
btfs	TRX	
bsf	Guarda_DIRECCION,0	
call	Termina_TRX	
TRX_TA_2		
call	Inicia_TRX	;Indica al dispositivo esclavo que se inicia la transferencia
bsf	TRX	;enviando una señal de 1 lógico al dispositivo
call	Termina_TRX	
bcf	TRX	
TRX_DD_2		
call	Inicia_TRX	;Indica al dispositivo esclavo que es un DATO lo que envía
bcf	TRX	;enviando una señal de 1 lógico al dispositivo
call	Termina_TRX	
bcf	TRX	
DATO_B7		
call	Inicia_TRX	
bcf	Guarda_DATO,7	
btfs	TRX	
bsf	Guarda_DATO,7	
call	Termina_TRX	
DATO_B6		
call	Inicia_TRX	
bcf	Guarda_DATO,6	
btfs	TRX	
bsf	Guarda_DATO,6	

```
call Termina_TRX

DATO_B5
call Inicia_TRX
bcf Guarda_DATO,5
btfsc TRX
bsf Guarda_DATO,5
call Termina_TRX

DATO_B4
call Inicia_TRX
bcf Guarda_DATO,4
btfsc TRX
bsf Guarda_DATO,4
call Termina_TRX

DATO_B3
call Inicia_TRX
bcf Guarda_DATO,3
btfsc TRX
bsf Guarda_DATO,3
call Termina_TRX

DATO_B2
call Inicia_TRX
bcf Guarda_DATO,2
btfsc TRX
bsf Guarda_DATO,2
call Termina_TRX

DATO_B1
call Inicia_TRX
bcf Guarda_DATO,1
btfsc TRX
bsf Guarda_DATO,1
call Termina_TRX

DATO_B0
call Inicia_TRX
bcf Guarda_DATO,0
btfsc TRX
bsf Guarda_DATO,0
call Termina_TRX

TRX_Fin
bcf TRX
goto Principal

INCLUDE <RETARDOS.INC>
INCLUDE <Sincronizacion.INC>

END
```

En la Fig. III.5.2 se muestra el circuito completo que cumple con la etapa de Decodificación de señal.

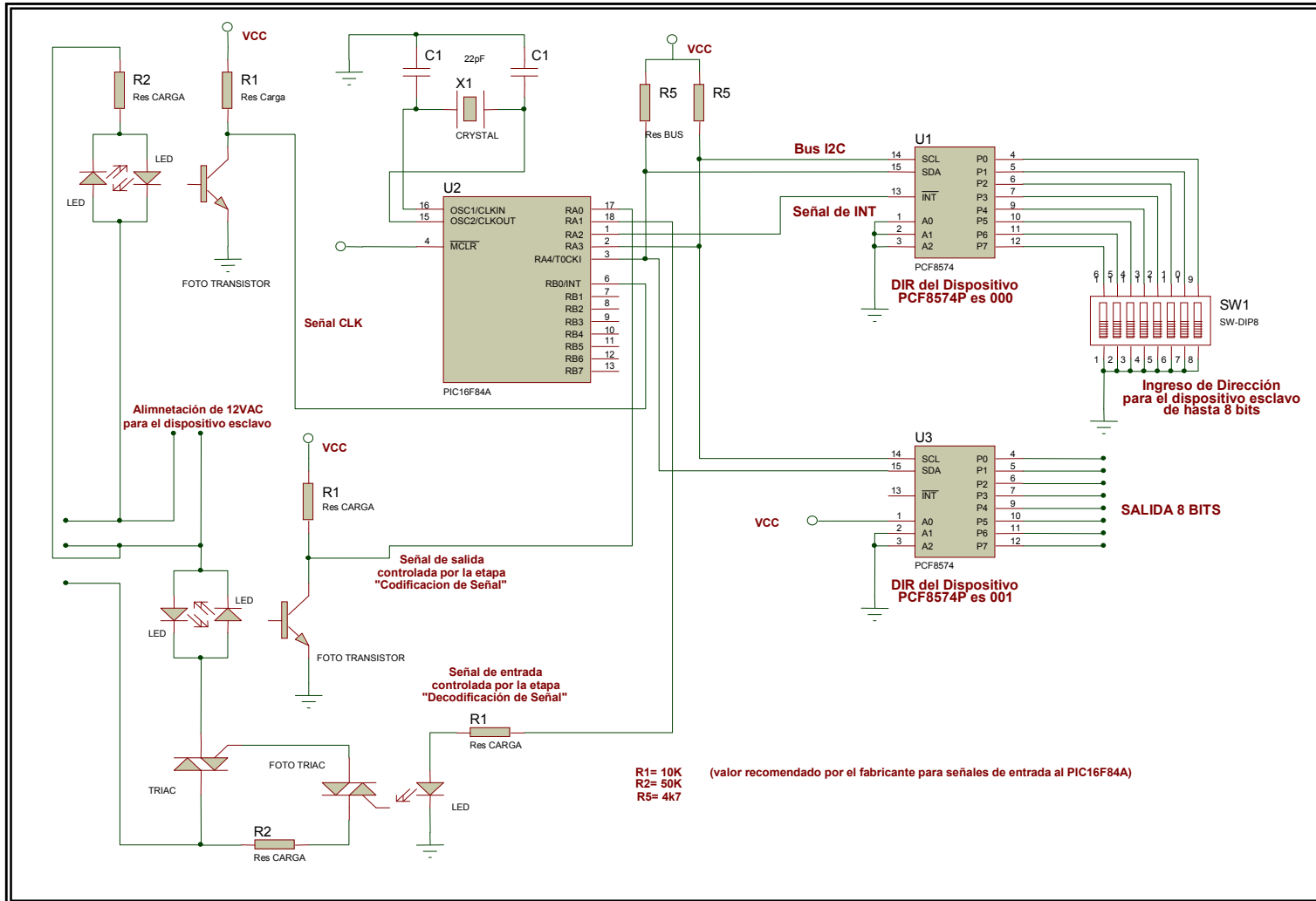


Fig. III.5.2 Circuito electrónico de la etapa “Decodificación de señal”.

CAP. IV.- CONSTRUCCIÓN Y PRUEBAS DEL SISTEMA

IV.1.- Introducción

En este último capítulo se desarrolla la construcción del dispositivo maestro y el dispositivo esclavo, con la finalidad de realizar los prototipos del sistema planteado en este trabajo y desarrollar de forma práctica lo aprendido.

IV.2.- Elaboración del prototipo maestro

El diagrama electrónico frontal del LCD, la conexión hacia el Puerto A (para las líneas PORTA,0; PORTA,1 y PORTA,2) y el Puerto B (para las líneas de PORTB,4 hasta PORTB,7) del microcontrolador además de los pines de alimentación para el LCD Ver Fig. IV.2.1.



Fig. IV.2.1 Circuito electrónico prototipo del LCD JHD162A y su conexión completa.

El diagrama electrónico que se utiliza para el teclado hexadecimal se muestra en la Fig. IV.2.2.

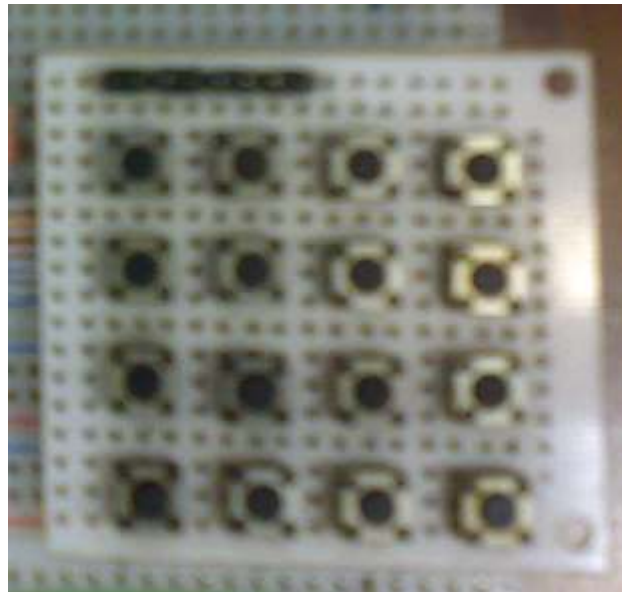


Fig. IV.2.2 Circuito electrónico prototipo para la colocación de los push button.

La capa siguiente (Ver Fig. IV.2.3) tiene los componentes electrónicos como resistencias, capacitores, integrados, etc., necesarios para hacer funcionar al teclado (con complemento de la primera capa)

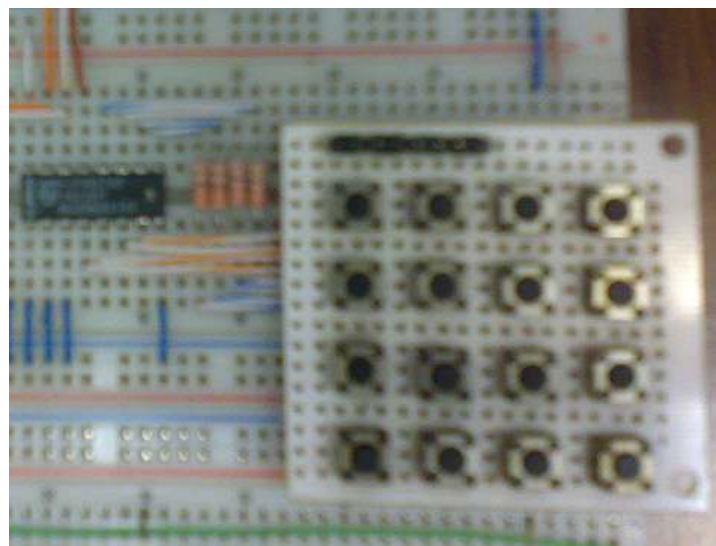


Fig. IV.2.3 Circuito electrónico prototipo donde muestra todos los componentes que forman la segunda capa para el teclado

La Fig. IV.2.4 muestra la imagen completa del LCD y el teclado hexadecimal

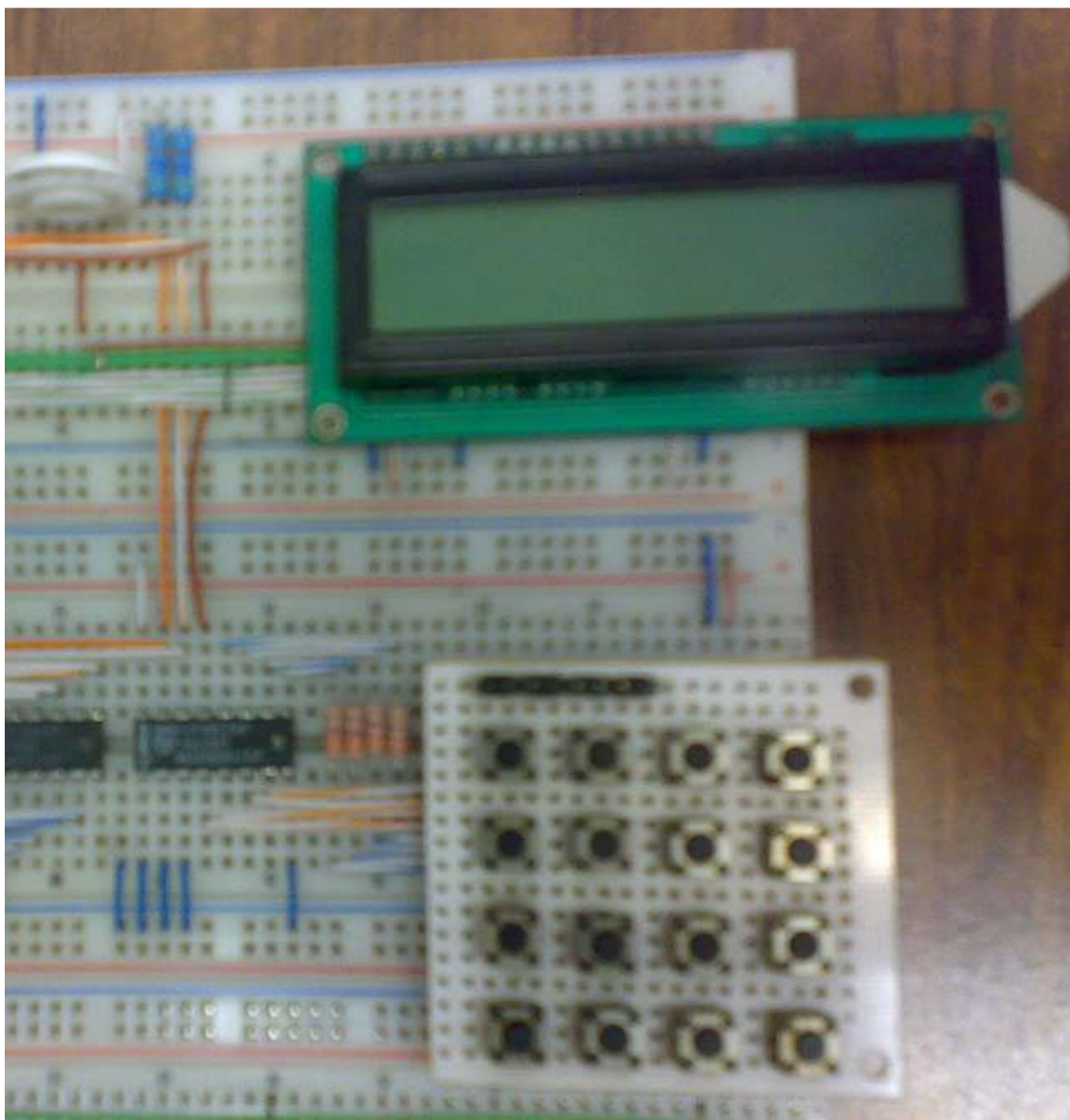


Fig. IV.2.4 Circuito electrónico *prototipo* de la etapa *Introducción de Datos*, en la que se utiliza una pantalla LCD y un teclado matricial.

El circuito electrónico que cumple con la Etapa **Adquisición de Datos** está compuesto por la parte de *Introducción de Datos* y la parte del *Procesamiento de Dato*, esta última se desarrolla a continuación y la forman: un microcontrolador PIC16F84A (con su configuración mínima para su funcionamiento) y el acoplador en paralelo con la Etapa de *Codificación de Señal* que consta de un PCF8574P comunicado en serie con el PIC usado para el Procesamiento de Datos, ver Fig. IV.2.5.

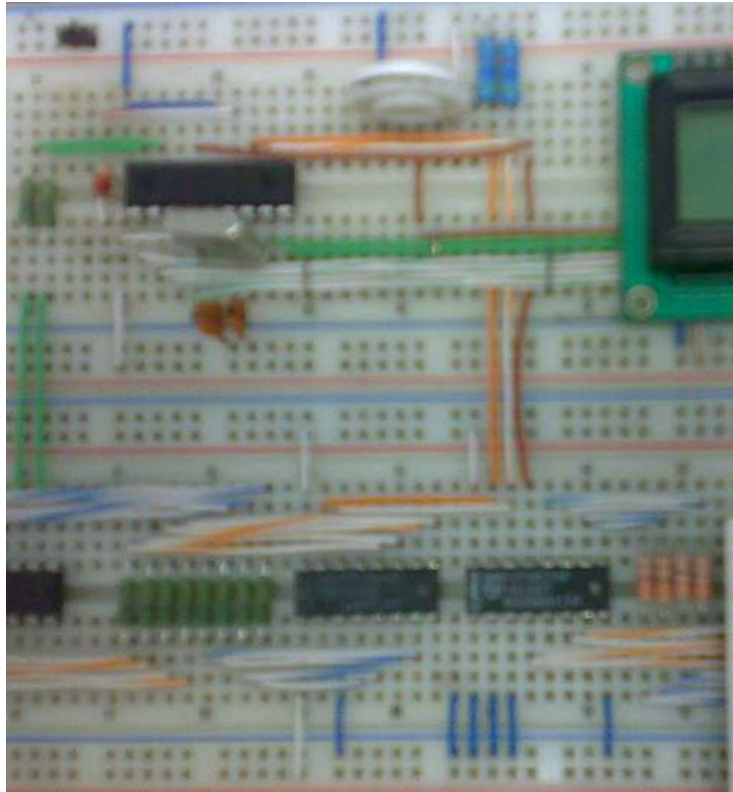


Fig. IV.2.5 Circuito electrónico propuesto para la parte de *Procesamiento de Datos* del proyecto, el control lo realiza el microcontrolador PIC16F84A.

El cruce por cero con salida a foto transistor se encuentra en el circuito integrado de 6 pines H11AA1 (Ver Anexo para sus especificaciones), la resistencia R1 se calcula para un voltaje de 12VCA y la resistencia R2 es de 10k Ω (es el valor recomendado por el fabricante) y el valor de VCC es de 5VCD ya que es el voltaje de alimentación para la parte electrónica, ver Fig. IV.2.6.



Fig. IV.2.6 Diagrama electrónico prototipo que se utiliza en el sistema de comunicación para el sensado de frecuencia.

El circuito electrónico para la etapa de Interfaz de envío que cumple con el diagrama anterior se muestra en la Fig. IV.2.7, la señal que controla la etapa Codificación de Señal está configurada por la línea 1 del Puerto A del microcontrolador (PORTA, 1 y para identificarla se rotula con el nombre “Escribe DATO”), mientras que la señal que envía el circuito electrónico de la etapa “Interfaz de recepción” que capta la etapa Codificación de Señal y ésta a su vez se encuentra configurada por la línea 2 del Puerto A del microcontrolador (PORTA, 2 y para identificarla se rotula con el nombre “Lee DATO”).

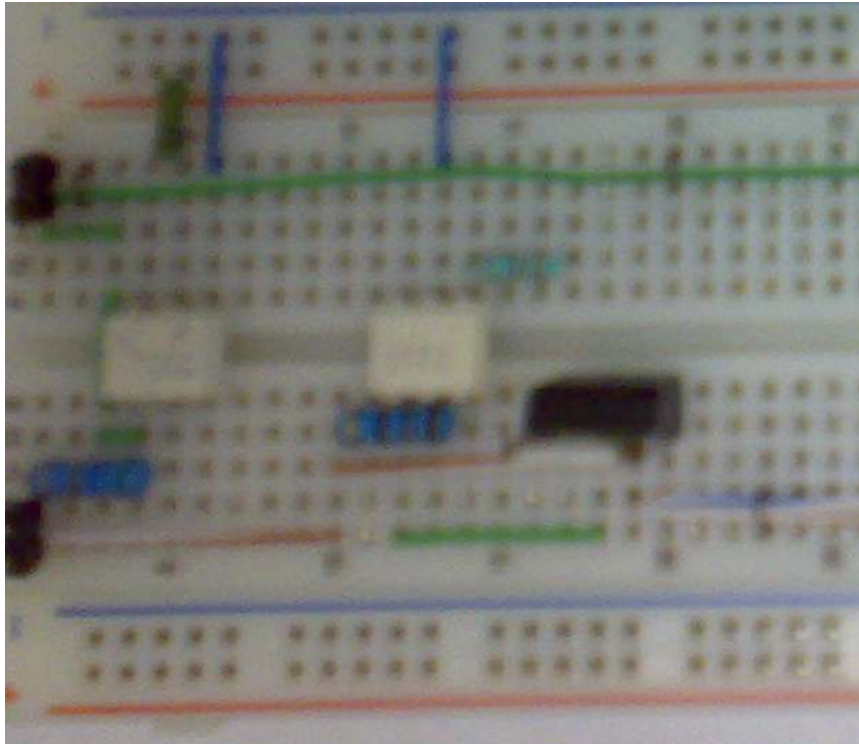


Fig. IV.2.7 Circuito electrónico de la etapa Interfaz de envío.

El circuito electrónico que cumple con las características del dispositivo maestro se muestra en la Fig. IV.2.8.

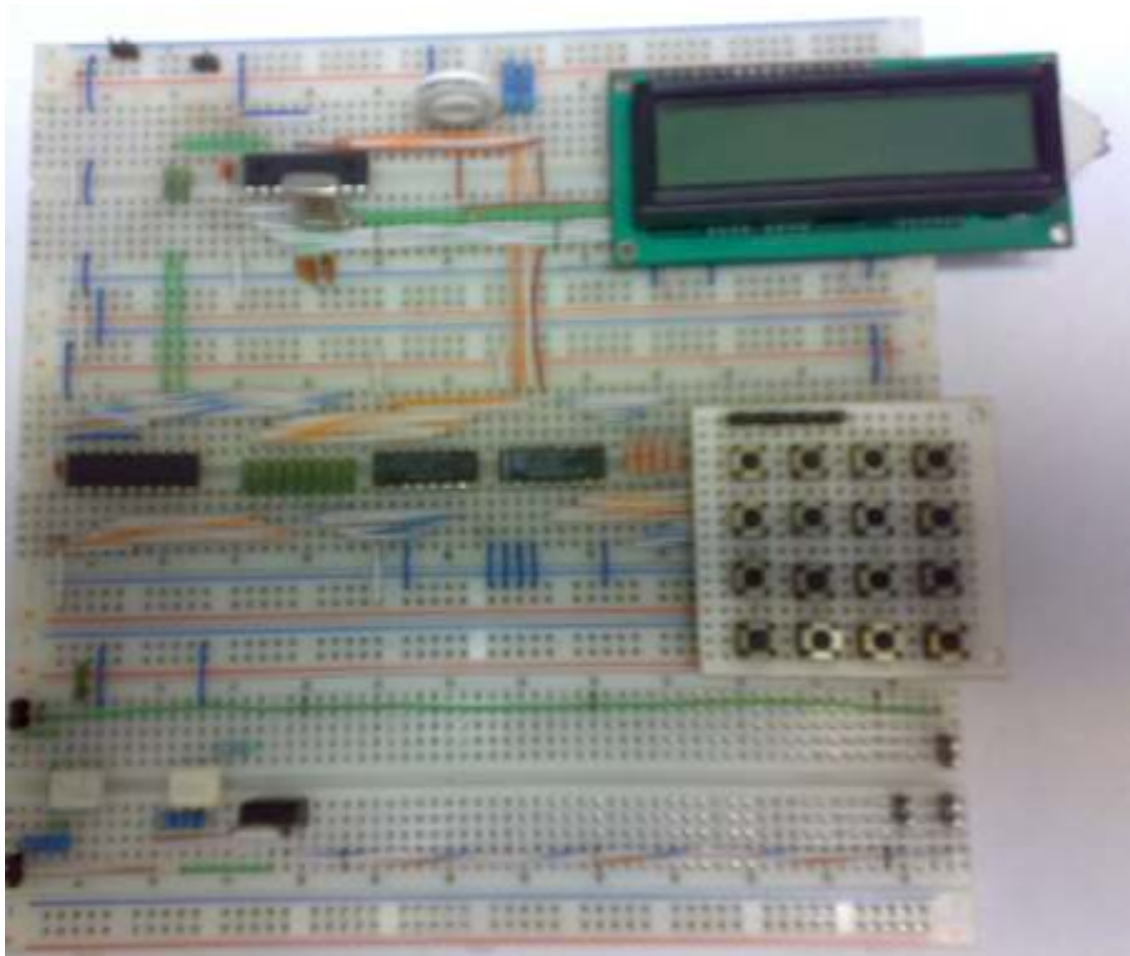


Fig. IV.2.8 Circuito electrónico prototipo completo que realiza las funciones del dispositivo maestro.

IV.3.- Elaboración del prototipo esclavo

El circuito electrónico para la etapa de Interfaz de recepción (Ver Fig. IV.3.1), tiene un funcionamiento muy similar que el de la etapa interfaz envío ya que al realizar una comunicación bidireccional ambos tienen los mismos dispositivos electrónicos solo que por ser un dispositivo esclavo hay algunos cambios en el diseño que se describen a continuación:

- La resistencia de carga R3 que el circuito de Interfaz de envío tiene se omite en el de Interfaz de recepción ya que se encuentra en serie y la R3 del circuito maestro (interfaz envío de datos), tiene la misma función para el circuito esclavo (interfaz recepción de datos).
- En el circuito maestro la alimentación para dispositivo llega directamente del transformador 110VAC/12VAC, mientras que el dispositivo esclavo es alimentado por el dispositivo maestro mediante las líneas de transmisión.
- Las señales de control de ambos dispositivos siempre serán opuestas, teniendo como preferencia la del dispositivo maestro, es decir, la señal de control en la etapa Decodificación de señal siempre estará activa en modo espera, mientras que las señal de control del dispositivo maestro estará siempre inactiva en modo espera hasta que el sistema de control Codificación de Señal ejecute una acción.

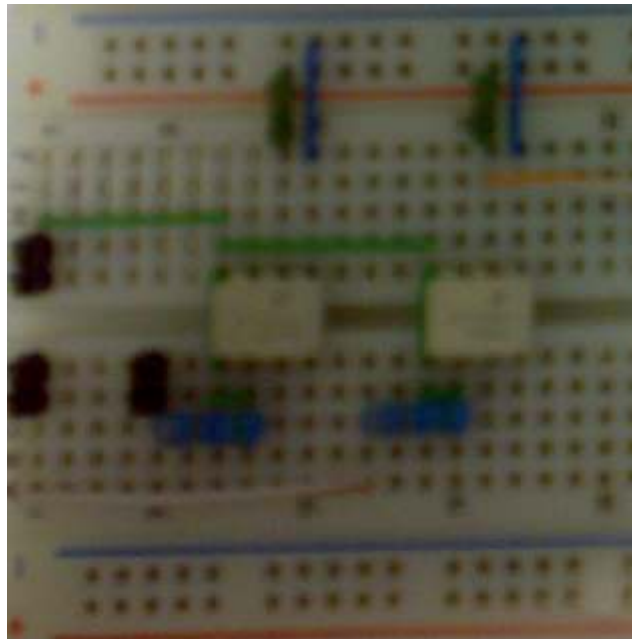


Fig. IV.3.1 Circuito electrónico prototipo de la etapa Interfaz de recepción.

El circuito electrónico propuesto para la etapa de direccionamiento se muestra en la Fig. IV.3.2, es en esta etapa donde se selecciona la dirección de manera manual, que utilizará el dispositivo esclavo a controlar.

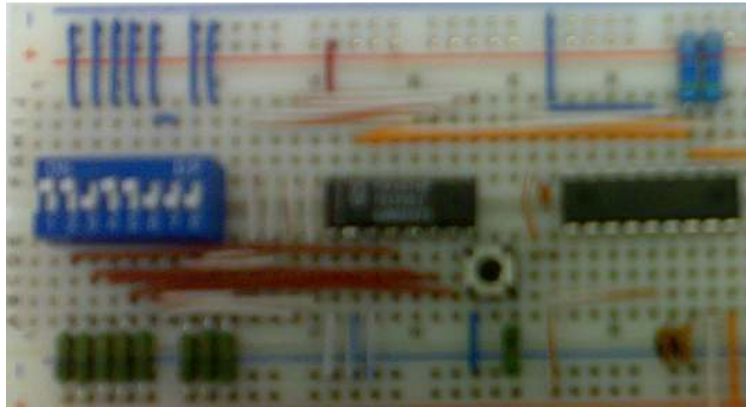


Fig. IV.3.2 Circuito electrónico prototipo de la etapa direccionamiento del dispositivo esclavo.

Se elabora un circuito electrónico actuador propuesto para identificar que la recepción de datos es correcta, Fig. IV.3.3, este circuito muestra en una barra de leds, el dato de ocho bits recibidos por dicho dispositivo.

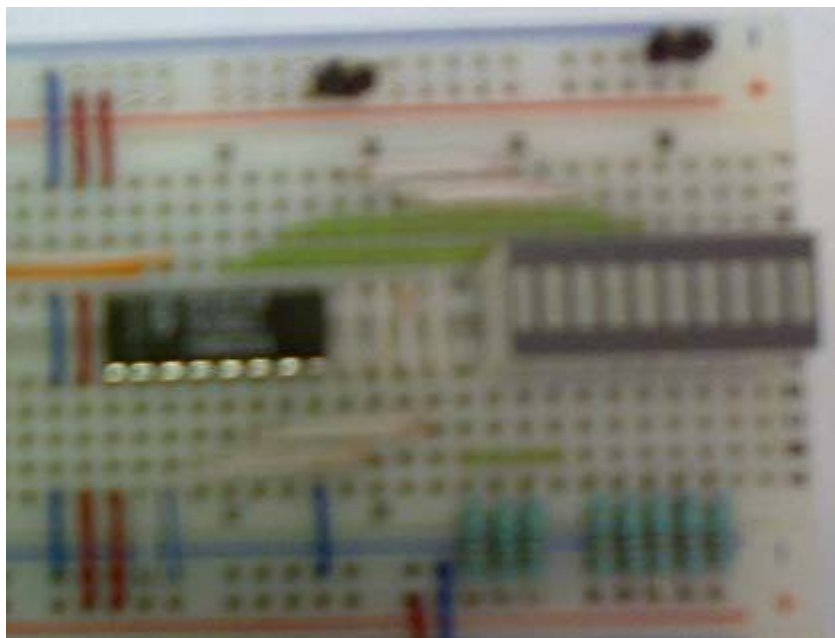


Fig. IV.3.3 Circuito electrónico actuador prototipo.

El circuito electrónico que cumple con la función del Dispositivo Esclavo se muestra en la Fig. IV.3.4.

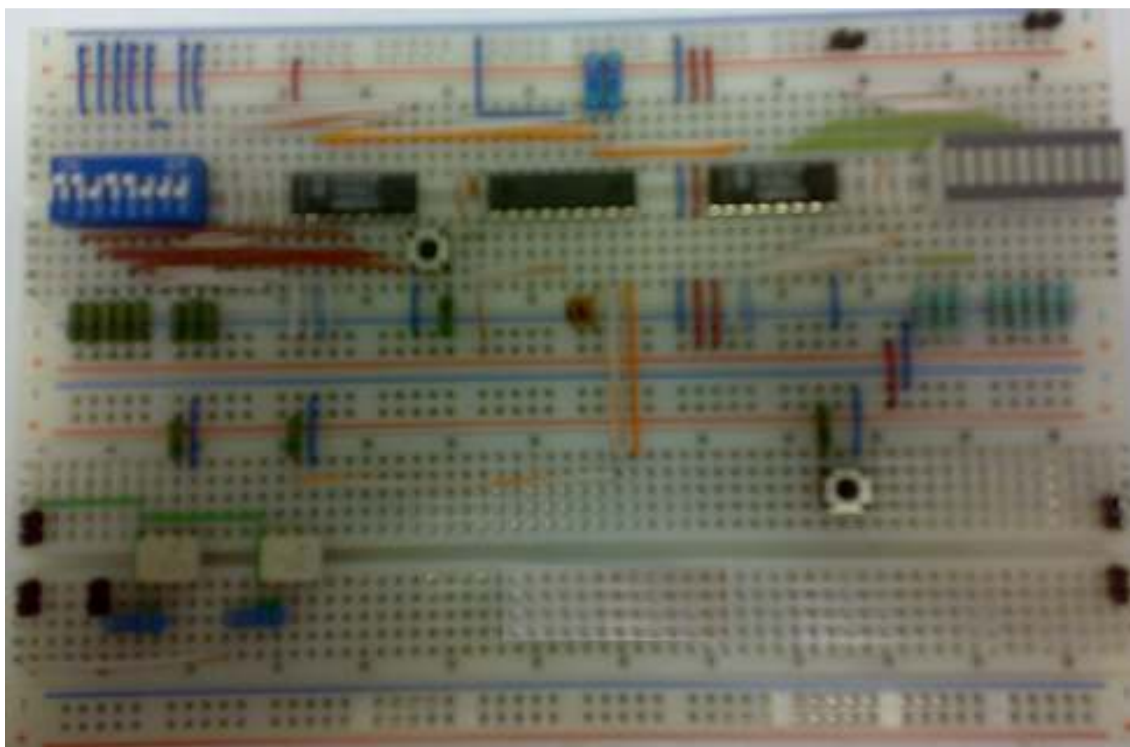


Fig. IV.3.4 Circuito electrónico prototipo completo que realiza las funciones del dispositivo esclavo.

Conclusiones

Se concluyó con el prototipo del proyecto planteado, partiendo de una necesidad de aplicar los conocimientos adquiridos durante el proceso de enseñanza-aprendizaje en la carrera de IME.

La diversidad del conocimiento aprendido en la facultad se incrementó y fundamentó a partir del ejercicio práctico en la elaboración de este proyecto, ya que se desarrollaron habilidades en el estudio de *análisis de proyectos, sistemas de control digital, análisis de señales, etc.*

El objetivo se cumplió satisfactoriamente para una comunicación unidireccional en alterna, partiendo de un protocolo propio que se propuso al inicio del mismo, con ello queda pendiente la necesidad de implementar para futuros proyectos la creación de diversos dispositivos esclavos para crear así, en una segunda etapa, una comunicación bidireccional compleja dando lugar al desarrollo e implantación en el sector productivo.

El presente trabajo cumplió con el objetivo de plantear un proyecto electrónico utilizando los conocimientos adquiridos durante la carrera de Ingeniero Mecánico Electricista, en su especialidad de Eléctrico Electrónico, dejando así la satisfacción y motivación para el ejercicio de una carrera profesional, partiendo de ser un estudiante de una de las instituciones más prestigiadas del país.

Se realizaron pruebas con los prototipos realizados, cumpliendo con las especificaciones iniciales planteados en el desarrollo del proyecto teniendo satisfactoriamente resultados positivos

Anexo

Tipos de comunicación sugeridos como parte de desarrollo posterior, para el mejoramiento de los prototipos: maestros y esclavo

La Comunicación Bidireccional Tipo Básica. En éste tipo de comunicación el dispositivo Tx envía un dato (o una serie de datos) a un único dispositivo Rx, el cual tiene la capacidad de enviar, como respuesta, un dato específico que el dispositivo Tx haya solicitado o como señal de un estado determinado, siguiendo en el campo de aplicación de la Domótica, en un área específica (como un pasillo de uso frecuente) a determinada hora es necesario un ahorro de energía en el alumbrado, se coloca un sensor de presencia (dispositivo Rx), el cual enviará una señal a el dispositivo Tx para que éste procese la variable y haga una configuración predeterminada (como el encendido de una luz auxiliar mientras el dispositivo Rx envíe una señal de presencia).

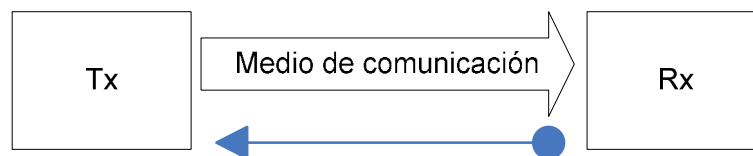


Fig. I.2.2 Diagrama a bloques de un sistema de comunicación bidireccional tipo básica.

La Comunicación Bidireccional Tipo Compleja (Se menciona como una característica de alcance, sin embargo este tipo de comunicación se propone como un trabajo de investigación posterior a la mejora del proyecto planteado).

Éste tipo de comunicación es la más completa y compleja, ya que es la intercomunicación de dispositivos Tx haciendo que entre éstos exista una transmisión mutua de datos, con la finalidad de controlar bloques de áreas específicas desde un dispositivo Tx cualquiera, siguiendo con la aplicación en el campo de la Domótica, en una casa habitación existen varios dispositivos Tx uno para cada zona de la casa (la recámara principal, la cocina, el estudio, etc.) los cuales tienen la finalidad de hacer un control inteligente para la zona específica en que se encuentren. Si los usuarios de esta casa habitación salen de ella, con un dispositivo Tx (cualquiera que se encuentre instalado) pueden hacer el control, por área, de apagado de contactos y control de diversos dispositivos Rx para éste fin “salí de casa”, es decir, el dispositivo Tx al que el usuario haya introducido la configuración “salí de casa”, éste enviará la información a cada uno de los dispositivos Tx que controlan áreas específicas para que realicen la configuración determinada para cada área (ya sea una configuración predeterminada por el dispositivo Tx o una que el usuario haya

configurado para este fin), por ejemplo, el dispositivo Tx que controla la zona de cocina, podría interrumpir la energía en todos los contactos de esta zona exceptuando el contacto que controla el refrigerador para evitar su descongelamiento, y así sucesivamente para cada área existe una configuración determinada.

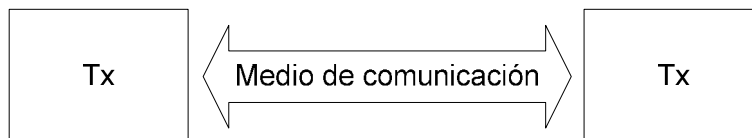


Fig. I.2.3 Diagrama a bloques de un sistema de comunicación bidireccional tipo compleja.

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio"

```

; - Retardo_100micros:      2 + 1 + 2 + (2 + 3K) = 100 cm = 100 μs. (para K= 31 y 4 MHz).
; - Retardo_50micros :     2 + 1 + 1 + 2 + (2 + 3K) = 50 cm = 50 μs. (para K= 14 y 4 MHz).
; - Retardo_20micros :     2 + 1 + (2 + 3K) = 20 cm = 20 μs. (para K= 5 y 4 MHz).
;
; RETARDOS de 1 ms hasta 200 ms. -----
;
Retardo_200ms
    movlw    d'200'                ; La llamada "call" aporta 2 ciclos máquina.
    goto    Retardos_ms           ; Aporta 1 ciclo máquina. Este es el valor de "M".
Retardo_100ms
    movlw    d'100'                ; Aporta 2 ciclos máquina.
    goto    Retardos_ms           ; La llamada "call" aporta 2 ciclos máquina.
Retardo_50ms
    movlw    d'50'                 ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto    Retardos_ms           ; Aporta 2 ciclos máquina.
Retardo_20ms
    movlw    d'20'                 ; La llamada "call" aporta 2 ciclos máquina.
    goto    Retardos_ms           ; Aporta 1 ciclo máquina. Este es el valor de "M".
Retardo_10ms
    movlw    d'10'                 ; Aporta 2 ciclos máquina.
    goto    Retardos_ms           ; La llamada "call" aporta 2 ciclos máquina.
Retardo_5ms
    movlw    d'5'                  ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto    Retardos_ms           ; Aporta 2 ciclos máquina.
Retardo_2ms
    movlw    d'2'                  ; La llamada "call" aporta 2 ciclos máquina.
    goto    Retardos_ms           ; Aporta 1 ciclo máquina. Este es el valor de "M".
Retardo_1ms
    movlw    d'1'                  ; Aporta 2 ciclos máquina.
    ; Aporta 1 ciclo máquina. Este es el valor de "M".
;
; El próximo bloque "Retardos_ms" tarda:
; 1 + M + M + KxM + (K-1)xM + Mx2 + (K-1)Mx2 + (M-1) + 2 + (M-1)x2 + 2 =
; = (2 + 4M + 4KM) ciclos máquina. Para K=249 y M=1 supone 1002 ciclos máquina
; que a 4 MHz son 1002 μs = 1 ms.
;
Retardos_ms
    movwf    R_ContB                ; Aporta 1 ciclo máquina.
R1ms_BucleExterno
    movlw    d'249'                ; Aporta Mx1 ciclos máquina. Este es el valor de "K".
    movwf    R_ContA                ; Aporta Mx1 ciclos máquina.
R1ms_BucleInterno
    nop                                ; Aporta KxMx1 ciclos máquina.
    decfsz   R_ContA,F              ; (K-1)xMx1 cm (cuando no salta) + Mx2 cm (al saltar).
    goto     R1ms_BucleInterno      ; Aporta (K-1)xMx2 ciclos máquina.
    decfsz   R_ContB,F              ; (M-1)x1 cm (cuando no salta) + 2 cm (al saltar).
    goto     R1ms_BucleExterno      ; Aporta (M-1)x2 ciclos máquina.
    return                            ; El salto del retorno aporta 2 ciclos máquina.
;
; En total estas subrutinas tardan:
; - Retardo_200ms: 2 + 1 + 2 + (2 + 4M + 4KM) = 200007 cm = 200 ms. (M=200 y K=249).
; - Retardo_100ms: 2 + 1 + 2 + (2 + 4M + 4KM) = 100007 cm = 100 ms. (M=100 y K=249).
; - Retardo_50ms : 2 + 1 + 2 + (2 + 4M + 4KM) = 50007 cm = 50 ms. (M= 50 y K=249).
; - Retardo_20ms : 2 + 1 + 2 + (2 + 4M + 4KM) = 20007 cm = 20 ms. (M= 20 y K=249).
; - Retardo_10ms : 2 + 1 + 2 + (2 + 4M + 4KM) = 10007 cm = 10 ms. (M= 10 y K=249).
; - Retardo_5ms  : 2 + 1 + 2 + (2 + 4M + 4KM) = 5007 cm = 5 ms. (M= 5 y K=249).
; - Retardo_2ms  : 2 + 1 + 2 + (2 + 4M + 4KM) = 2007 cm = 2 ms. (M= 2 y K=249).
; - Retardo_1ms  : 2 + 1 + (2 + 4M + 4KM) = 1005 cm = 1 ms. (M= 1 y K=249).
;
; RETARDOS de 0.5 hasta 20 segundos -----
;
Retardo_20s
    movlw    d'200'                ; La llamada "call" aporta 2 ciclos máquina.
    goto    Retardo_1Decima        ; Aporta 1 ciclo máquina. Este es el valor de "N".
Retardo_10s
    movlw    d'100'                ; Aporta 2 ciclos máquina.
    goto    Retardo_1Decima        ; La llamada "call" aporta 2 ciclos máquina.
Retardo_5s
    movlw    d'50'                 ; Aporta 1 ciclo máquina. Este es el valor de "N".
    goto    Retardo_1Decima        ; Aporta 2 ciclos máquina.
Retardo_2s
    movlw    d'20'                 ; La llamada "call" aporta 2 ciclos máquina.
    goto    Retardo_1Decima        ; Aporta 1 ciclo máquina. Este es el valor de "N".
Retardo_1s
    movlw    d'10'                 ; Aporta 2 ciclos máquina.
    goto    Retardo_1Decima        ; La llamada "call" aporta 2 ciclos máquina.

```


Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio"

```

Retardo_500ms                ; La llamada "call" aporta 2 ciclos máquina.
    movlw    d'5'             ; Aporta 1 ciclo máquina. Este es el valor de "N".
;
; El próximo bloque "Retardo_1Decima" tarda:
; 1 + N + N + MxN + MxN + KxMxN + (K-1)xMxN + MxNx2 + (K-1)xMxNx2 +
; + (M-1)xN + Nx2 + (M-1)xNx2 + (N-1) + 2 + (N-1)x2 + 2 =
; = (2 + 4M + 4MN + 4KM) ciclos máquina. Para K=249, M=100 y N=1 supone 100011
; ciclos máquina que a 4 MHz son 100011 µs = 100 ms = 0,1 s = 1 décima de segundo.
;
Retardo_1Decima
    movwf   R_ContC           ; Aporta 1 ciclo máquina.
R1Decima_BucleExterno2
    movlw   d'100'           ; Aporta Nx1 ciclos máquina. Este es el valor de "M".
    movwf   R_ContB         ; Aporta Nx1 ciclos máquina.
R1Decima_BucleExterno
    movlw   d'249'          ; Aporta MxNx1 ciclos máquina. Este es el valor de "K".
    movwf   R_ContA         ; Aporta MxNx1 ciclos máquina.
R1Decima_BucleInterno
    nop                ; Aporta KxMxNx1 ciclos máquina.
    decfsz  R_ContA,F   ; (K-1)xMxNx1 cm (si no salta) + MxNx2 cm (al saltar).
    goto    R1Decima_BucleInterno ; Aporta (K-1)xMxNx2 ciclos máquina.
    decfsz  R_ContB,F   ; (M-1)xNx1 cm (cuando no salta) + Nx2 cm (al saltar).
    goto    R1Decima_BucleExterno ; Aporta (M-1)xNx2 ciclos máquina.
    decfsz  R_ContC,F   ; (N-1)x1 cm (cuando no salta) + 2 cm (al saltar).
    goto    R1Decima_BucleExterno2 ; Aporta (N-1)x2 ciclos máquina.
    return                ; El salto del retorno aporta 2 ciclos máquina.
;
; En total estas subrutinas tardan:
; - Retardo_20s:  2 + 1 + 2 + (2 + 4N + 4MN + 4KMN) = 20000807 cm = 20 s.
;                 (N=200, M=100 y K=249).
; - Retardo_10s:  2 + 1 + 2 + (2 + 4N + 4MN + 4KMN) = 10000407 cm = 10 s.
;                 (N=100, M=100 y K=249).
; - Retardo_5s:   2 + 1 + 2 + (2 + 4N + 4MN + 4KMN) = 5000207 cm = 5 s.
;                 (N= 50, M=100 y K=249).
; - Retardo_2s:   2 + 1 + 2 + (2 + 4N + 4MN + 4KMN) = 2000087 cm = 2 s.
;                 (N= 20, M=100 y K=249).
; - Retardo_1s:   2 + 1 + 2 + (2 + 4N + 4MN + 4KMN) = 1000047 cm = 1 s.
;                 (N= 10, M=100 y K=249).
; - Retardo_500ms: 2 + 1 + (2 + 4N + 4MN + 4KMN) = 500025 cm = 0,5 s.
;                 (N= 5, M=100 y K=249).
;
; =====
; Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
; E. Palacios, F. Remiro y L. López.
; Editorial Ra-Ma. www.ra-ma.es
; =====

```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio"

```

***** Librería "PCF8574.INC" *****
;
;
;
;=====
; Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
; E. Palacios, F. Remiro y L. López.
; Editorial Ra-Ma. www.ra-ma.es
;=====
;
; Estas subrutinas permiten realizar las tareas de manejo del expansor de bus I2C PCF8574
; que convierte un bus de 8 líneas paralelo a bus serie I2C y viceversa.
;
; Como es posible que haya más de un expansor en el mismo circuito, se deja a la subrutina
; de llamada a éstas que determine las direcciones de escritura y lectura mediante las
; etiquetas "PCF8574_DireccionLectura" y "PCF8574_DireccionEscritura", atendiendo a las
; características del hardware utilizado.
;
; ZONA DE DATOS *****
;
; CBLOCK
; PCF8574_Dato ; Aquí guarda el resultado del dato leído o que
; ENDC ; se va a escribir.
;
; Subrutina "PCF8574_Lee" -----
;
; Lee el periférico conectado al bus paralelo de 8 líneas del PCF8574 que actúa como
; entrada y el resultado lo proporciona en el registro de trabajo W.
;
PCF8574_Lee
call I2C_EnviaStart ; Envía la condición de Start.
movlw PCF8574_DireccionLectura ; Apunta al expansor de lectura.
call I2C_EnviaByte
bsf I2C_UltimoByteLeer
call I2C_LeerByte ; Lee el puerto.
movwf PCF8574_Dato ; Lo guarda en el registro correspondiente.
call I2C_EnviaStop ; Acaba de leer.
movf PCF8574_Dato,W ; Recupera el valor del dato leído.
return
;
; Subrutina "PCF8574_Escribe" -----
;
; Escribe en el periférico conectado bus paralelo de 8 líneas del expansor PCF8574 que
; actúa como salida, el dato que le llega por el registro de trabajo W. El dato escrito
; es recuperado finalmente en el registro W.
;
PCF8574_Escribe
movwf PCF8574_Dato ; Guarda el dato a escribir.
call I2C_EnviaStart ; Envía condición de Start.
movlw PCF8574_DireccionEscritura ; Apunta al expansor que actúa como salida.
call I2C_EnviaByte
movf PCF8574_Dato,W ; Recupera el valor del dato a enviar.
call I2C_EnviaByte ; Envía el dato.
call I2C_EnviaStop ; Termina.
movf PCF8574_Dato,W ; Recupera el valor del dato escrito.
return
;
;=====
; Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
; E. Palacios, F. Remiro y L. López.
; Editorial Ra-Ma. www.ra-ma.es
;=====
;

```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio"

```

***** Librería "BUS_I2C.INC" *****
;
;
;
;=====
;
; Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"
; E. Palacios, F. Remiro y L. López.
; Editorial Ra-Ma. www.ra-ma.es
;=====
;
; Estas subrutinas permiten realizar las tareas básicas de control del bus serie I2C,
; por parte de un solo microcontrolador maestro.
;
; ZONA DE DATOS *****
;
; CBLOCK
; I2C_ContadorBits          ; Cuenta los bits a transmitir o a recibir.
; I2C_Dato                  ; Dato a transmitir o recibido.
; I2C_Flags                 ; Guarda la información del estado del bus I2C.
; ENDC

#define I2C_UltimoByteLeer   I2C_Flags,0
; - (I2C_UltimoByteLeer)=0, NO es el último byte a leer por el maestro.
; - (I2C_UltimoByteLeer)=1, SÍ es el último byte a leer por el maestro.

; La definición de las líneas SCL y SDA del bus I2C se puede cambiar según las
; necesidades del hardware.

#define SCL   PORTA,3          ; Línea SCL del bus I2C.
#define SDA   PORTA,4          ; Línea SDA del bus I2C.
;
; Subrutina "SDA_Bajo" -----
;
; SDA_Bajo
; bsf     STATUS,RP0          ; Configura la línea SDA como salida.
; bcf     SDA
; bcf     STATUS,RP0
; bcf     SDA                 ; SDA en bajo.
; return

;
; Subrutina "SDA_AltImpedancia" -----
;
; SDA_AltImpedancia
; bsf     STATUS,RP0          ; Configura la línea SDA entrada.
; bsf     SDA                 ; Lo pone en alta impedancia y, gracias a la Rp
; bcf     STATUS,RP0          ; de esta línea, se mantiene a nivel alto.
; return

;
; Subrutina "SCL_Bajo" -----
;
; SCL_Bajo
; bsf     STATUS,RP0          ; Configura la línea SCL como salida.
; bcf     SCL
; bcf     STATUS,RP0
; bcf     SCL                 ; La línea de reloj SCL en bajo.
; return

;
; Subrutina "SCL_AltImpedancia" -----
;
; SCL_AltImpedancia
; bsf     STATUS,RP0          ; Configura la línea SCL entrada.
; bsf     SCL                 ; Lo pone en alta impedancia y, gracias a la Rp
; bcf     STATUS,RP0          ; de esta línea, se mantiene a nivel alto.
;
; SCL_EsperaNivelAlto
; btfss   SCL                 ; Si algún esclavo mantiene esta línea en bajo
; goto    SCL_EsperaNivelAlto ; hay que esperar.
; return

;
; Subrutina "I2C_EnviaStart" -----
;
; Esta subrutina envía una condición de Start o inicio.
;
; I2C_EnviaStart
; call    SDA_AltImpedancia   ; Línea SDA en alto.
; call    SCL_AltImpedancia   ; Línea SCL en alto.
; call    Retardo_4micros     ; Tiempo tBUF del protocolo.
; call    SDA_Bajo            ; Flanco de bajada de SDA mientras SCL está alto.

```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio"

```

call Retardo_4micros ; Tiempo tHD;STA del protocolo.
call SCL_Bajo ; Flanco de bajada del reloj SCL.
call Retardo_4micros
return
;
; Subrutina "I2C_EnviaStop" -----
;
; Esta subrutina envía un condición de Stop o parada.
;
I2C_EnviaStop
call SDA_Bajo
call SCL_AltImpedancia ; Flanco de subida de SCL.
call Retardo_4micros ; Tiempo tSU;STO del protocolo.
call SDA_AltImpedancia ; Flanco de subida de SDA.
call Retardo_4micros ; Tiempo tBUF del protocolo.
return
;
; Subrutina "I2C_EnviaByte" -----
;
; El microcontrolador maestro transmite un byte por el bus I2C, comenzando por el bit
; MSB. El byte a transmitir debe estar cargado previamente en el registro de trabajo W.
; De la subrutina ejecutada anteriormente I2C_EnviaStart o esta misma I2C_EnviaByte,
; la línea SCL se debe encontrar a nivel bajo al menos durante 5 µs.
;
I2C_EnviaByte
movwf I2C_Dato ; Almacena el byte a transmitir.
movlw 0x08 ; A transmitir 8 bits.
movwf I2C_ContadorBits
I2C_EnviaBit
rfl I2C_Dato,F ; Chequea el bit, llevándolo previamente al Carry.
btfsc STATUS,C
goto I2C_EnviaUno
I2C_EnviaCero
call SDA_Bajo ; Si es "0" envía un nivel bajo.
goto I2C_FlancoSCL
I2C_EnviaUno
call SDA_AltImpedancia ; Si es "1" lo activará a alto.
I2C_FlancoSCL
call SCL_AltImpedancia ; Flanco de subida del SCL.
call Retardo_4micros ; Tiempo tHIGH del protocolo.
call SCL_Bajo ; Termina el semiperiodo positivo del reloj.
call Retardo_4micros ; Tiempo tHD;DAT del protocolo.
decfsz I2C_ContadorBits,F ; Lazo para los ocho bits.
goto I2C_EnviaBit
;
call SDA_AltImpedancia ; Libera la línea de datos.
call SCL_AltImpedancia ; Pulso en alto de reloj para que el esclavo
call Retardo_4micros ; pueda enviar el bit ACK.
call SCL_Bajo
call Retardo_4micros
return
;
; Subrutina "I2C_LeeByte" -----
;
; El microcontrolador maestro lee un byte desde el esclavo conectado al bus I2C. El dato
; recibido se carga en el registro I2C_Dato y lo envía a la subrutina superior a través
; del registro W. Se empieza a leer por el bit de mayor peso MSB.
; De alguna de las subrutinas ejecutadas anteriormente I2C_EnviaStart, I2C_EnviaByte
; o esta misma I2C_LeeByte, la línea SCL lleva en bajo al menos 5 µs.
;
I2C_LeeByte
movlw 0x08 ; A recibir 8 bits.
movwf I2C_ContadorBits
call SDA_AltImpedancia ; Deja libre la línea de datos.
I2C_LeeBit
call SCL_AltImpedancia ; Flanco de subida del reloj.
bcf STATUS,C ; En principio supone que es "0".
btfsc SDA ; Lee el bit
bsf STATUS,C ; Si es "1" carga 1 en el Carry.
rfl I2C_Dato,F ; Lo introduce en el registro.
call SCL_Bajo ; Termina el semiperiodo positivo del reloj.
call Retardo_4micros ; Tiempo tHD;DAT del protocolo.
decfsz I2C_ContadorBits,F ; Lazo para los 8 bits.
goto I2C_LeeBit
;
;

```

Elaboración del proyecto "Diseño de un sistema electrónico de envío y recepción de datos,
basado en un protocolo propio"

```
; Chequea si este es el último byte a leer para enviar o no el bit de reconocimiento  
; ACK en consecuencia.  
;
```

```
    btfss    I2C_UltimoByteLeer    ; Si es el último, no debe enviar  
                                     ; el bit de reconocimiento ACK.  
    call     SDA_Bajo              ; Envía el bit de reconocimiento ACK  
                                     ; porque todavía no es el último byte a leer.  
    call     SCL_AltImpedancia     ; Pulso en alto del SCL para transmitir el  
    call     Retardo_4micros       ; bit ACK de reconocimiento. Este es THIGH.  
    call     SCL_Bajo              ; Pulso de bajada del SCL.  
    call     Retardo_4micros  
    movf    I2C_Dato,W             ; El resultado se manda en el registro de  
    return                                     ; de trabajo W.
```

```
;  
;===== Del libro "MICROCONTROLADOR PIC16F84. DESARROLLO DE PROYECTOS"  
; E. Palacios, F. Remiro y L. López.  
; Editorial Ra-Ma. www.ra-ma.es  
;=====
```