



Universidad del Azuay

Facultad de Ciencia y Tecnología

Escuela de Ingeniería Electrónica

Diseño y construcción de un prototipo de telemetría y control en el sistema de medición en la EMPRESA ELÉCTRICA REGIONAL CENTRO SUR C.A.

Tesis previa a la obtención del título de

Ingeniero Electrónico

Autores

Juan Pablo Dávalos Ríos

Freddy German Montero Mariño

Director

Eduardo Rodrigo Sempértegui Cañizares

Cuenca - Ecuador

2013

DEDICATORIA

Germán Montero

Elsy Mariño

Marcelo Dávalos

Nancy Ríos

AGRADECIMIENTO

Al Ing. Xavier Gutiérrez e Ing. Eduardo Sempértegui, miembros de la Empresa Eléctrica Regional Centro Sur C.A., por sus valiosos aportes en la elaboración de este trabajo.

A nuestros padres y hermanos por el apoyo y el cariño de siempre.

A nuestros amigos y el resto de familiares por el soporte y el apoyo incondicional brindado para la elaboración de la tesis

INDICE DE CONTENIDOS

DEDICATORIA	i
INDICE DE CONTENIDOS	iii
ÍNDICE DE FIGURAS.....	viii
RESUMEN	xii
ABSTRACT.....	xiii
INTRODUCCIÓN.....	1
CAPITULO I: RED ELÉCTRICA INTELIGENTE Y REUTILIZACIÓN DE MEDIDORES	
1. Introducción	3
2. Sistemas AMR	3
2.1. Características	4
2.2. Estructura de un Sistema AMR	4
2.2.1. Medidor de energía digital AMR.....	5
2.2.2. Concentrador de datos	5
2.2.3. Servidor de datos.....	5
2.3. Aplicaciones.....	6
2.4. Ventajas de los AMR	6
3. Sistemas AMI.....	7
3.1. Características	8
3.2. Componentes de una Sistema AMI	8
3.2.1. Medidores	8
3.2.2. Red de comunicación	8
3.2.3. Sistema de recolección de Datos	9
3.2.4. Interoperabilidad	9

3.3. Ventajas de un sistema AMI.....	10
4. Smart Grids	11
4.1. Estructura de un Smart Grid	11
4.2. Ventajas de los Smart Grids	12
5. Reutilización de los medidores de la empresa eléctrica.....	14

CAPITULO II: DISEÑO Y CONSTRUCCIÓN DE HARDWARE

Introducción	15
1. Conceptos teóricos	15
1.1. Relé	15
1.1.1. Concepto de Relé	15
1.1.2. Tipos de Relés	15
1.1.3. Relé de estado sólido.....	16
1.2.1. Pic 18f4550	17
1.2.2. Distribución de pines del pic 18f4550	19
1.3. Módulo de comunicación Xbee.....	19
1.3.1. Concepto de Xbee	20
1.3.2. Tipos de Xbee	20
1.3.3. Características Xbee Serie 2B	20
1.3.4. Distribución de los pines del módulo XBee y tarjeta Xbee	21
1.4. Fuente de Corriente Continua.....	22
2. Diseño de hardware.....	23
2.1. Diseño de la fuente de tensión de 5 voltios	24
2.1.1. Materiales.....	24
2.1.2. Diseño	24
2.2. Conexión del microcontrolador	26
2.2.1. Materiales.....	26
2.2.2. Diseño	26
2.3. Conexión del módulo de comunicación xbee	28
2.3.1. Materiales.....	28
2.3.2. Diseño	29
2.4. Conexión del LCD	29
2.4.1. Materiales.....	29

2.4.2. Diseño	30
2.5. Conexión de Relé de Estado Solido	31
2.5.1. Materiales.....	31
2.5.2. Diseño	31
2.6. Conexión del contador de pulsos y pulsantes de sincronización.....	32
2.6.1. Materiales.....	32
2.6.2. Diseño	33
2.6.2.1. Contador de pulsos	33
2.6.2.2. Pulsantes de sincronización	34
2.7. Diseño del circuito Final	35
3. Construcción de la tarjeta	36
4. Esquema de montaje del circuito con el medidor	39

CAPITULO III: DISEÑO DE SOFTWARE

Introducción	40
1. Marco teórico	40
1.1. LabVIEW	41
1.2. Base de Datos.....	42
1.2.1. Marcas de Base de Datos.....	42
1.2.1.1. MySQL	42
1.3. Enlace de conexión a una Base de Datos con Labview	43
1.4. MikroC.....	43
2. Crear Base de Datos	44
3. Programas en LabVIEW	49
3.1. Ingresar Nuevo Cliente.....	49
3.2. Toma de lecturas	52
3.3. Ingresar código de medidor	55
3.4. Pago por servicio.....	56
3.5. Corte y reconexión del servicio	58
3.6. Visualización de la base de datos	60
3.7. Registro de datos por tiempo.....	60
3.8. Perfil de carga	62

4. Programación en el microcontrolador utilizando MikroC	63
4.1. Declaración de variables	63
4.2. Conexión del módulo LCD	64
4.3. Inicialización del programa principal e inicialización de variables	65
4.3.1. Inicialización de los puertos	65
4.3.2. Inicialización del Puerto Serial USART	66
4.3.3. Inicialización de las interrupciones	67
4.3.4. Inicialización del LCD	67
4.3.5. Inicialización del Variables	68
4.4. Lectura inicial de la EEPROM	69
4.4.1. Lectura de los dígitos de la medida	70
4.4.2. Lectura de los dígitos del medidor	71
4.4.3. Lectura del estado del relé de corte o reconexión	71
4.5. Construcción del bucle principal	72
4.5.1. Verificación del estado del relé	72
4.5.2. Operaciones para obtener los dígitos de la medida	73
4.5.3. Envío de datos hacia el concentrador	75
4.5.4 Visualización de los datos en el LCD	77
4.6. Interrupciones por cambio de estado en el puerto B	80
4.6.1. Interrupciones en el puerto RB0	80
4.6.2. Interrupciones en el puerto RB1	81
4.6.3. Interrupciones en el puerto RB2	83
4.7. Interrupción por recepción de datos	84
4.7.1. Acciones de corte y reconexión	85
4.7.2. Recepción del número del medidor	86
4.7.3. Dato para guardar en la EEPROM	88
5. Configuración del Xbee	88
5.1. Configuración del módulo concentrador	89
5.2. Configuración de los módulos terminales	93

CAPITULO IV: PRUEBAS DE FUNCIONAMIENTO

1. Introducción	94
2. Pruebas de hardware en el protoboard	94

2.1 Funcionamiento de la fuente de 5 voltios	94
2.2. Funcionamiento del microcontrolador y del LCD	95
2.3. Funcionamiento del Xbee.....	96
2.4. Funcionamiento del Relé de Estado Solido	97
2.5. Funcionamiento del Contador de pulsos.....	98
2. Pruebas de hardware en la tarjeta	99
3. Pruebas de Software	102
3.1. Pruebas de lectura	102
3.2. Pruebas de Corte y Reconexión.....	113
CONCLUSIONES	120
RECOMENDACIONES	121
BIBLIOGRAFÍA	122
GLOSARIO	126
ANEXO	127

ÍNDICE DE FIGURAS

Figura 1. 1: Sistema básico de un sistema AMR	4
Figura 1. 2: Componentes de un sistema AMI	10
Figura 1. 3: Comparación de Sistemas	12
Figura 2. 1: Relé de estado sólido de 60 Amperios	16
Tabla 2. 1: Características Microcontroladores	18
Figura 2. 2: Pinaje del pic 18f4550	21
Figura 2. 3: Distribución de Pines Xbee S2B	21
Figura 2. 4: Tarjeta reguladora de voltaje y distribución de pines	22
Figura 2. 5: Fuente de corriente Alterna	23
Figura 2. 6: Fuente de Corriente Continua	23
Figura 2. 7: Señales del puente de Graetz	25
Figura 2. 8: Onda de salida del filtro	25
Figura 2. 9: Diseño de la fuente de 5 voltios de corriente continúa	26
Figura 2. 10: Pic 18f4550	27
Figura 2. 11: Conexión básica del 18F4550	28
Figura 2. 12: Conexión de un Xbee	29
Figura 2. 13: Conexión LCD	30
Figura 2. 14: Conexión del Relé de estado sólido	32
Figura 2. 15: Contador de Pulsos	33
Figura 2. 16: Conexión de pulsantes de sincronización	34
Figura 2. 17: Diseño del circuito final	35
Figura 2. 18: Modelo Placa del Medidor	36
Figura 2. 19: Ruteo de la Placa	37
Figura 2. 20: Vista física de la placa	37
Figura 2. 21: Vista física de la placa	38
Figura 2. 22: Vista de la placa con todos sus componentes	38
Figura 2. 23: Esquema de montaje del prototipo con el medidor	39

Figura 3. 1: Pantalla de inicio de programa Labview 2011	41
Figura 3. 2: Pagina inicial de ODBC Data Source Administrator	44
Figura 3. 3: Lista con tipos de datos fuente	45
Figura 3. 4: Conector ODBC.....	45
Figura 3. 5: Crear un enlace de datos	46
Figura 3. 6: Propiedades del enlace de datos	47
Figura 3. 7: Escoger nombre de la fuente de datos	48
Figura 3. 8: Test de conexión exitosa	48
Figura 3. 9: Guardar archivo .udl	49
Figura 3. 10: Panel Frontal de Ingresar Nuevo Cliente	50
Figura 3. 11: Diagrama de bloques de Ingresar Nuevo Cliente	51
Figura 3. 12: Panel Frontal de Toma de lecturas y control del servicio	52
Figura 3. 13: Diagrama de bloques de Toma de lecturas y control del servicio	54
Figura 3. 14: Panel Frontal de Ingresar código de medidor	55
Figura 3. 15: Diagrama de Bloques de Ingresar código de medidor	56
Figura 3. 16: Panel Frontal de Pago	56
Figura 3. 17: Diagrama de Bloques de Pago	57
Figura 3. 18: Panel Frontal de Corte y Reconexión.....	58
Figura 3. 19: Diagrama de Bloques de Corte y Reconexión.....	59
Figura 3. 20: Diagrama de Bloques de Visualización de la base de datos	60
Figura 3. 21: Diagrama de Bloques de Registro de datos por tiempo.....	61
Figura 3. 22: Panel Frontal Perfil de carga	62
Figura 3. 23: Diagrama de Bloques perfil de carga	63
Figura 3. 24: Declaración de variables	64
Figura 3. 25: Conexión e inicialización del módulo LCD	64
Figura 3. 26: Programa inicial.....	65
Figura 3. 27: Configuración de puertos	66
Figura 3. 28: Configuración EUSART	66
Figura 3. 29: Configuración interrupciones	67
Figura 3. 30: Inicialización del LCD	68
Figura 3. 31: Inicialización de variables.....	69
Figura 3. 32: Primera lectura de la EEPROM	70
Figura 3. 33: Lectura de dígitos de la medida	70

Figura 3. 34: Lectura de dígitos del medidor.....	71
Figura 3. 35: Lectura EEPROM estado del servicio.....	72
Figura 3. 36: Bucle infinito	72
Figura 3. 37: Bucle principal.....	73
Figura 3. 38: Contadores de la medida.....	74
Figura 3. 39: Contadores de la medida.....	75
Figura 3. 40: Transmisión de la medida	76
Figura 3. 41: Transmisión de código del medidor	76
Figura 3. 42: Puntero para sincronizar la medida en el LCD.....	78
Figura 3. 43: Visualización completa en LCD	79
Figura 3. 44: Interrupción en el puerto RB0.....	80
Figura 3. 45: Interrupción en el puerto RB1	82
Figura 3. 46(a): Interrupción en el puerto RB2	83
Figura 3.46 (b): Interrupción en el puerto RB2	84
Figura 3. 47: Condición “Q” en recepción	85
Figura 3. 48: Acción de corte y reconexión.....	86
Figura 3. 49: Recepción código del medidor.....	87
Figura 3. 50: Guardar datos en la memoria EEPROM	88
Figura 3. 51: PC Settings X – CTU.....	90
Figura 3.52: Configuración modem ZIGBEE COORNIDADOR AT	91
Figura 3. 53: Habilitar encriptación.....	92
Figura 3. 54: Ingresar la llave de encriptación	92
Figura 3. 55: Configuración Modem ZIGBEE ROUTER AT	93
Figura 4. 1: Funcionamiento de la Fuente de tensión de 5 voltios.....	95
Figura 4. 2: Funcionamiento del microcontrolador y LCD	95
Figura 4. 3: Funcionamiento de la alimentación del Xbee	96
Figura 4. 4: Funcionamiento de la comunicación del xbee 1.....	96
Figura 4. 5: Funcionamiento de la comunicación del xbee 2.....	97
Figura 4. 6: Funcionamiento del Relé de Estado Solido.....	97
Figura 4. 7: Señal generadora de pulsos	98
Figura 4. 8: Primera medición de pulsos	98
Figura 4. 9: Segunda medición de pulsos	99
Figura 4. 10 (a): Prueba 1 Funcionamiento del contador de pulsos	100

Figura 4.9 (b): Prueba 2 Funcionamiento del contador de pulsos	100
Figura 4. 11: Funcionamiento del Xbee en la placa.....	101
Figura 4. 12: Relé de estado sólido conectado a la placa.....	102
Figura 4. 13: Conexión Final de todo el circuito	102
Figura 4. 15: Configuración de banco de trabajo para pruebas de la lectura	103
Figura 4. 16: Medidor 1 en minutos iniciales	104
Figura 4. 17: Placa 1 en minutos iniciales	104
Figura 4. 18: Medidor 2 en minutos iniciales	105
Figura 4. 19: Placa 1 en minutos iniciales	105
Figura 4. 20 (a): Registro de medidas de la tarjeta 1	106
Figura 4.19 (b): Registro de medidas de la tarjeta 1	106
Figura 4.19 (c): Registro de medidas de la tarjeta 1	107
Figura 4.19 (d): Registro de medidas de la tarjeta 1	107
Figura 4.19 (e): Registro de medidas de la tarjeta 1	108
Figura 4.19 (f): Registro de medidas de la tarjeta 1	108
Figura 4. 21 (a): Registro de medidas de la tarjeta 2	109
Figura 4.20 (b): Registro de medidas de la tarjeta 2	109
Figura 4.20 (c): Registro de medidas de la tarjeta 2	109
Figura 4.20 (d): Registro de medidas de la tarjeta 2	110
Figura 4.20 (e): Registro de medidas de la tarjeta 2	110
Figura 4.20 (f): Registro de medidas de la tarjeta 2.....	110
Figura 4. 22: Medidor 1, minutos finales	111
Figura 4. 23: Placa 1, minutos finales	111
Figura 4. 24: Medidor 2, minutos finales	112
Figura 4. 25: Placa 1, minutos finales	112
Figura 4. 27: Relé en estado de corte para la tarjeta 1	113
Figura 4. 29: Relé en estado de conectado para la tarjeta 1	115
Figura 4. 31: Relé en estado de corte para la tarjeta 1	116
Figura 4. 32: Relé en estado de corte para la tarjeta 2	117
Figura 4. 34: Relé en estado de conectado para la tarjeta 1	118
Figura 4. 35: Relé en estado de conectado para la tarjeta 2	119
Figura 4. 36: Tamaño de la información en la Base de Datos	119

030613

RESUMEN

DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE TELEMETRÍA Y CONTROL EN EL SISTEMA DE MEDICIÓN EN LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR C.A.

Para diseñar y construir un circuito electrónico capaz de realizar las acciones de lectura, corte y reconexión del servicio de energía eléctrica y un software que controle y monitoree remotamente estos procesos. Se tomó como referencia los sistemas de medición inteligente que se implementan para gestionar la información en línea de los clientes y controlar desde una central.

Se reutilizó los medidores monofásicos híbridos acoplado un circuito electrónico, para procesar la lectura, que sea transmitida hacia un computador mediante un protocolo Zigbee y controle el componente encargado del corte y reconexión del servicio, mediante un programa en Labview.

Es una solución eficiente al problema de acceso físico a los medidores instalados en diferentes lugares de la ciudad, eliminando las dificultades, costos y molestias adicionales relacionadas con el ingreso a domicilios en donde se necesita autorización para poder realizar las lecturas, cortes o reconexiones.

Palabras Claves: Zigbee, Labview, AMI (Advanced Metering Infrastructure), AMR (Automatic Meter Reading), Smart grid, Xbee, X-CTU, telemedición, medidor ciclométrico.



DIRECTOR DE ESCUELA

Ing. Francisco Vásquez.



DIRECTOR DE TESIS

Ing. Eduardo Sempertegui



AUTOR

Freddy Germán Montero Mariño



AUTOR

Juan Pablo Dávalos Ríos

Suárez

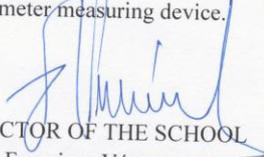
*Comunicado
24
06/06/13*

ABSTRACT

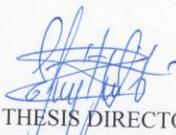
DESIGN AND CONSTRUCTION OF A TELEMETRY AND CONTROL PROTOTYPE IN THE MEASUREMENT SYSTEM OF EMPRESA ELECTRICA CENTRO SUR C.A. ELECTRIC COMPANY

The smart metering system implemented for the management of online information of the clients was taken as a reference point in order to design and build an electronic circuit capable of performing actions such as meter reading, shutdown, and reconnection of the electric service, and to develop software that controls and monitors these processes from the distance. We reutilized the hybrid single-phase measuring devices by adapting an electronic circuit to process the meter reading that is then transmitted to a computer through Zigbee protocol and controls the component in charge of the shutdown and reconnection of the service through Labview program. It is an efficient solution to the problem of physical access to the measuring devices installed in different parts of the city. It eliminates the difficulties, costs, and additional problems related to entering the homes where authorization is needed to carry out meter readings, shutdowns or reconnections.

Key Words: Zigbee, Labview, AMI (Advanced Metering Infrastructure), AMR (Automatic Meter Reading), Smart grid, Xbee, X-CTU, telemetry, cyclometer measuring device.



DIRECTOR OF THE SCHOOL
Ing. Francisco Vásquez



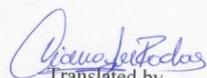
THESIS DIRECTOR
Ing. Eduardo Sempertegui



AUTHOR
Freddy German Montero Mariño



AUTHOR
Juan Pablo Dávila Ríos



Translated by,
Diana Lee Rodas

Dávalos Ríos Juan Pablo
Montero Mariño Freddy German
Trabajo de Grado
Ing. Eduardo Rodrigo Sempértegui Cañizares
Junio, 2013

**DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE
TELEMETRÍA Y CONTROL EN EL SISTEMA DE MEDICIÓN EN
LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR C.A.**

INTRODUCCIÓN

En la actualidad la facturación de servicios de suministros de electricidad se ha convertido en uno de los principales problemas para las compañías proveedoras de energía, debido al crecimiento constante del número de usuarios. Estas compañías se han visto obligadas a realizar importantes inversiones tanto en recursos humanos como en tiempo, por otro lado los errores que se comenten como consecuencia de la operación manual de los registros de las lecturas, han generado constantes quejas por parte de los clientes.

Actualmente en la ciudad de Cuenca, para cobrar por el servicio prestado, la compañía de suministro de electricidad instala en cada uno de las residencias o puntos de facturación un medidor de consumo eléctrico. Con una periodicidad no menor a un mes el personal autorizado realiza de forma manual y localmente la lectura en el medidor del servicio y anota el consumo correspondiente.

Es importante recalcar que debido al crecimiento de la cantidad de clientes, esta tarea demanda cada vez una mayor cantidad de inversión en mano de obra y la frecuencia de la medición mensual se ve disminuida. La implementación de una infraestructura de medición avanzada de última tecnología, optimizaría varios de los procesos, la

problemática en este caso tendría lugar en los costos de compra e implementación. El adquirir un sistema con estas prestaciones en la actualidad resulta demasiado costoso, por lo que por el momento no se considera una solución debido a la gran inversión que esto implica.

En el presente estudio se construyó un sistema de telemetría y control que brindara una solución eficiente a estos inconvenientes. Se diseñó un sistema remoto capaz de realizar acciones de lectura, corte y re-conexión de los sistemas de medición de energía eléctrica de los clientes, además de la implementación de una base de datos encargada de almacenar información de cada usuario.

Este prototipo está constituido por una tarjeta electrónica que se acopla a los medidores de marcador ciclométrico que implementa la empresa eléctrica actualmente. Estas tarjetas adquieren señales pulsantes provenientes de los marcadores ciclo-métricos y las transforma a una medida de consumo.

Ya con la medida, la placa envía la información hacia un computador a través de un protocolo inalámbrico el mismo que se encuentra debidamente encriptado. Se diseñó un programa el cual interpreta los datos de medida recibidos en el computador. Este programa cuenta con herramientas que nos permiten enviar y recibir información de la tarjeta, con lo cual podemos realizar las acciones de lectura corte y reconexión principalmente.

CAPITULO I

RED ELÉCTRICA INTELIGENTE Y REUTILIZACIÓN DE MEDIDORES

1. Introducción

En la actualidad con los avances tecnológicos, sobre todo en los países desarrollados se ha permitido implementar nuevas técnicas de medición y facturación más eficientes, confiables y económicas en lo referente a consumo eléctrico. El desarrollo de la electrónica que ha traído como consecuencia la aparición de equipos de medición más precisos y económicos llamado comúnmente "Instrumentación inteligente", ha permitido avances en los sistemas de comunicaciones y las redes públicas; ya que brindan diversas formas de conectarse remotamente y bidireccional entre los dispositivos inteligentes instalados en cada residencia y los centros de control o facturación.

2. Sistemas AMR

Que por sus siglas en ingles significan "Automatic Meter Reading" o lectura de medición remota, básicamente es un sistema diseñado para realizar lecturas de mediciones automáticas con mayor eficiencia, monitoreando los datos de energía y administrando el consumo a distancia. [2]

2.1. Características

Esta tecnología nos permite la recopilación automática del consumo, diagnósticos y datos de estado del medidor de energía, además permite la transferencia de estos datos hacia una base de datos que permitirá la facturación del servicio. Esta tecnología ahorra la mano de obra de los servicios de lectura corte y reconexión, principalmente el gasto de los viajes periódicos a cada residencia para leer el dato de un medidor. Otra de las funcionalidades más ventajosas es que la facturación se la puede realizar en tiempo real, en lugar de tener que esperar los cálculos basados en el pasado o consumo previo. Esta información oportuna, junto con el análisis puede ayudar a los proveedores de servicios públicos y los clientes a controlar mejor el uso y la producción de energía eléctrica, gas, o agua.

2.2. Estructura de un Sistema AMR

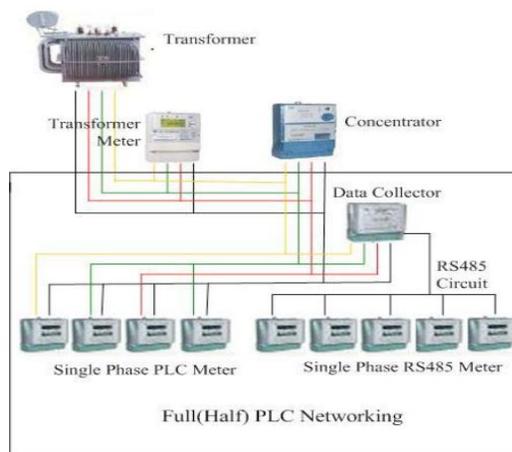


Figura 1. 1: Sistema básico de un sistema ARM

(SISTEMAS AMR) [Ref. 9 de Octubre de 2012]

Fuente: http://www.romatech.cl/Romatech/AMR_files/AMR%20Expo%20202.pdf

A continuación describiremos los componentes principales de este sistema:

2.2.1. Medidor de energía digital AMR

Reemplaza al medidor mecánico. Se trata de un medidor digital capaz de registrar los parámetros eléctricos, con diferentes tipos de protocolos para el envío de la información; ya sea este vía inalámbrica o por las redes eléctricas de baja o media tensión, con el objetivo de obtener remotamente las medidas.

2.2.2. Concentrador de datos

El concentrador es el dispositivo utilizado para la captura automática de los datos de contadores y dispositivos tarifarios, almacenando estos datos en su memoria interna y transfiriéndolos al centro de almacenamiento de datos. La información obtenida no se procesa en el registrador, sino que es transferida ya sea por medio de redes Ethernet, PSTN/GSM/GPRS, RS232/RS485, etc.

2.2.3. Servidor de datos

Es un componente que se instala en la compañía eléctrica. Se trata de un software que permite al administrador ingresar a cada uno de los concentradores y medidores instalados, haciendo posible que se puedan obtener los datos en tiempo real. El programa cuenta con un administrador de base de datos SQL el cual nos permitirá realizar una facturación más precisa del consumo.

2.3. Aplicaciones

Las aplicaciones de este tipo de sistemas están creciendo sistemáticamente a escala mundial. A continuación enumeraremos las principales:

- Medición y facturación de servicios de Electricidad gas y agua en zonas residenciales.
- Medición y facturación de servicios de Electricidad gas y agua en zonas remotas de gran dispersión geográfica.
- Medición y facturación de servicios de energía eléctrica, gas y agua en parques industriales.
- Facturación única de los servicios de agua, energía eléctrica, y gas.
- Mantenimiento preventivo.
- Servicios de atención al cliente.

2.4. Ventajas de los AMR

- Mayor seguridad de los datos

Con la lectura automática de datos se proporciona una mayor seguridad a la hora de la lectura de datos, ya que se prescinde de los servicios de una persona que realice esto evitando errores de lectura y falta de lecturas de contadores, también se evita que las personas manipulen los medidores para su beneficio.

- Costes de explotación más bajos

Se puede tener los datos a tiempo real a cualquier hora, proporcionando al operador una ayuda eficiente ante cualquier reclamo. Por el lado económico, los costes de lectura de

contadores se ven disminuidos por la disminución de personal, y con ellos los costes totales de explotación.

- Mejora en el presupuesto y la gestión del flujo de caja

La facturación se realiza a partir de datos en tiempo real, desapareciendo la necesidad de facturar según estimaciones, se evitan errores a la hora de cobrar y se agilitan todos los procesos por los cuales normalmente las empresas proveedoras de los servicios pierden dinero.

- Mejor servicio al cliente

Si las facturas de consumo son exactas, no hacen falta estimaciones ni correcciones. Usted podrá reaccionar más rápidamente en situaciones anormales y monitorizar más de cerca la demanda y el consumo además se agilizaría el corte y reconexión del servicio en caso de ser necesario.

3. Sistemas AMI

El sistema de Infraestructura de medición avanzada (IMA) o Advanced Metering Infrastructure (AMI) en inglés, plantea una infraestructura con dos vías de comunicación capaz de intercambiar información entre el medidor inteligente del cliente y la empresa proveedora del servicio, con capacidad de controlar a distancia el corte y reconexión del servicio, registrar formas de onda, vigilar el voltaje y la corriente.¹

¹VIDRIO LÓPEZ, GILBERTO. Infraestructura de Medición Avanzada AMI en las redes Inteligentes. En línea. Internet. Disponible: < <http://www.slideshare.net/FiiDEM/infraestructura-de-medicin-avanzada-ami-en-las-redes-inteligentes>>. Consulta: 11 de Octubre de 2012.

3.1. Características

Las características principales de los sistemas AMI son los siguientes:

- Sistema Bidireccional, es decir, permite el intercambio de información desde la central al medidor y del medidor a la central.
- Lectura de datos en tiempo real.
- Interoperabilidad de equipos.
- Protocolos de comunicación abiertos
- Gestión y administración del corte y reconexión del servicio automatizado.
- Medidor medio de comunicación del cliente.

3.2. Componentes de una Sistema AMI

3.2.1. Medidores

Los medidores AMI, por lo general constan de un componente medidor y la Unidad de Interfaz del Medidor (MIU), los cuales se encargan de manejar la información y comunicarse con la red local. No requieren ser leídos manualmente, tienen la capacidad de realizar cortes y reconexión en forma remota, pueden limitar el suministro de electricidad, son capaces de detectar robo de energía, los consumidores disponen de mayor información sobre su consumo, etc.

3.2.2. Red de comunicación

La comunicación entre medidores y central es bidireccional, desde el medidor se envían los datos recogidos del consumo del servicio, mientras que desde la central se envían datos de control hacia el medidor. La comunicación no siempre es la misma, depende de

muchos factores por lo que es difícil definir un solo sistema de comunicación. Esta clase de medidores tienen la capacidad de transmitir la información recolectada a través de redes fijas, como la Banda Ancha sobre la línea eléctrica (BPL), PLC, Redes fijas de Radio frecuencia (RF) y redes públicas.

3.2.3. Sistema de recolección de Datos

Es el sistema en el cual la red recolecta, almacena, retransmite y muestra los datos adquiridos en un tiempo determinado. Un Gestor de datos medidos (MDM), es en donde los datos son recolectados y se transforman en información válida para estar disponibles en los servicios administrativos de los datos. Trabaja con un Repositorio de Datos (MDR) que es constituida por una base de datos para el almacenamiento de la información.

3.2.4. Interoperabilidad

Es la capacidad que tienen los equipos de diferentes fabricantes de comunicarse entre sí, es el requisito para que los sistemas heterogéneos puedan intercambiar información. Es decir se refiere a la compatibilidad que requieren los equipos de diferentes marcas para funcionar correctamente.

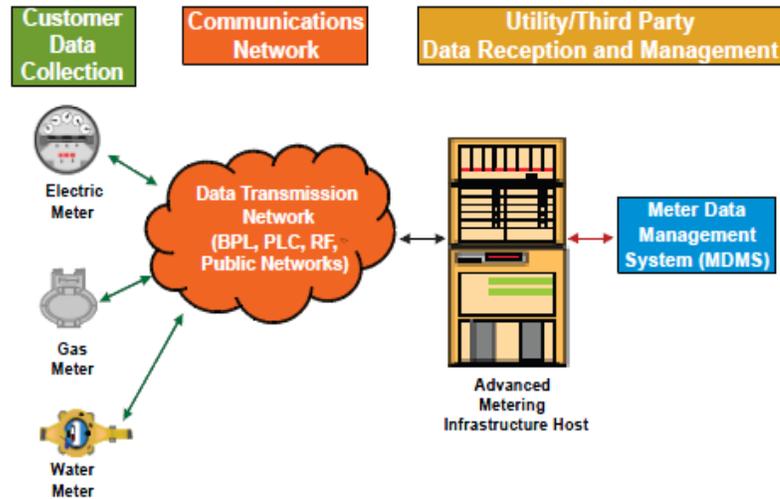


Figura 1. 2: Componentes de un sistema AMI

(ELECTRIC POWER RESEARCH INSTITUTE. Advanced metering infrastructure. 2007) [Ref. 11 de Octubre de 2012]

Fuente: <http://www.ferc.gov/eventcalendar/Files/20070423091846-EPRI%20-%20Advanced%20Metering.pdf>

3.3. Ventajas de un sistema AMI

- Consumidor

Para el consumidor aumenta las opciones acerca del servicio y el precio, más información para manejar su consumo, controlar sus costos, mayor confiabilidad, cuentas más precisas y lo más importante es que reducen costos operativos por lo tanto reducen también los costos del servicio eléctrico

- Beneficios de Utilidad

El sistema AMI anula la estimación de las lecturas, permitiendo tener una facturación más precisa y oportuna, ofreciendo así, un mejor servicio al cliente. Con este tipo de sistemas se puedes saber inmediatamente cuando y en donde existió un corte del

servicio, de esta manera se puede trabajar más oportuna y eficientemente controlando en todo momento el servicio de la empresa que ofrece.

Un beneficio significativo es administrar eficazmente las cuentas de los clientes. Empezando con la capacidad de corte y reconexión remota del servicio, eliminando la necesidad de enviar personal al sitio, en lugar de basarse en estimaciones, los ingenieros pueden adecuadamente dimensionar los equipos y los dispositivos de protección, y comprender de una mejor manera el comportamiento del sistema de distribución.

4. Smart Grids

Es una red inteligente capaz de realizar todo tipo de acción y monitoreo sobre ella, con el fin de conseguir un suministro eléctrico eficiente, seguro y sostenible. Los Smart Grids son sistemas que utilizan equipos y servicios de última tecnología en los campos de las comunicaciones, control, monitorización y auto-diagnostico con la finalidad de conseguir los siguientes objetivos:

- Robustecer y automatizar la red
- Reducción de pérdidas de energía
- Mejorar la calidad del servicio
- Disminuir el costo de operación por red

4.1. Estructura de un Smart Grid

La estructura de un Smart Grid se compone de las redes AMI y AMR, en la siguiente figura se muestra el alcance, los beneficios y la composición de este tipo de sistemas además de la relación de esta con las redes ya mencionadas anteriormente en este capítulo.



Figura 1. 3: Comparación de Sistemas

(CORONEL GUTIÉRREZ, MARCO. (ESTUDIO PARA LA IMPLEMENTACIÓN DEL SISTEMA AMI EN LA EMPRESA ELÉCTRICA REGIONAL CENTRO SUR C.A. INGENIERÍA ELÉCTRICA. Universidad Politécnica Salesiana, 2011) [Ref. 11 de Octubre de 2012]

Como se puede observar en la figura, el componente más básico de este sistema es el AMR, el cual gracias a sus prestaciones mejoradas en relación de los medidores antiguos nos permite obtener de modo remoto las lecturas de cada medidor de consumo eléctrico. El segundo componente fundamental es el sistemas AMI, cuyas prestaciones son mayores a los AMR, ya que a más de una lectura remota, podemos tener un control bidireccional del sistema y lecturas de muchos más parámetros a más de la medida de consumo. Por último el SMART GRID optimiza cada una de las prestaciones de los sistemas anteriores formando redes eficientes y de última tecnología.

4.2. Ventajas de los Smart Grids

A continuación indicaremos las principales ventajas del uso de esta tecnología para las empresas proveedoras de este servicio:

- Reducción de pérdidas de energía

Se podrá gestionar de manera autónoma, identificable y controlable el gasto de la energía en cada uno de los hogares, permitiendo que tanto el consumidor como la empresa proveedora de energía eléctrica conozcan el balance de consumo de energía mensual.

- Eficiencia

Se podrá realizar de una manera eficaz y precisa los análisis de patrones de consumo, siendo de gran utilidad ya que el personal de la empresa eléctrica necesitara mucho menos tiempo a la hora de facturar el consumo y procesar los reclamos por parte de los abonados.

- Pago por uso

Se eliminan estimaciones por parte de los que realizan las medidas, por lo que el abonado tendrá que pagar exclusivamente lo que consumen. Esto reducirá los errores que frecuentemente se producen a la hora de pagar la planilla, además de los reclamos por parte de los clientes.

- Gestión en remoto del suministro de energía

No será necesario contar con personal que conecte y desconecte cada red manualmente, si no que se contara con un sistema remoto que lo haga con mayores facilidades. Con esto se elimina que personal de la empresa eléctrica tenga que realizar visitas periódicas a las casas de los abonados.

- Promover el uso de energías renovables

Se promoverá el uso de energías renovables, ya que al ser un sistema bidireccional, se podrá saber el consumo y la generación de energía en el lado de los abonados. Esto permitirá que los clientes que cuenten con sistemas de energías renovables puedan tener remuneraciones si es que generan energía.

5. Reutilización de los medidores de la empresa eléctrica

Para este prototipo de tesis se reutilizaron los medidores electrónicos que actualmente la Empresa Eléctrica implementa en la casa de clientes, Este modelo de medidor monofásico fue construido exclusivamente para la zona residencial. Diseñado para la medición de 1 sola fase, este medidor aguanta un máximo de 60 Amperios de corriente, funciona a 60Hz de frecuencia y posee un microcontrolador o pic interno el cual realiza la acción de medición. Este pic genera 3200 impulsos de pequeñas tensiones por cada kWh consumido, los mismos que son transformados a un valor de medida que se visualiza en un registrador ciclométrico.

CAPITULO II

DISEÑO Y CONSTRUCCIÓN DE HARDWARE

Introducción

En el presente capítulo se detalla paso a paso la elaboración de la parte de hardware del trabajo de tesis, en el que constan cada uno los procesos que unen a los componentes de la placa que realizaran todas las acciones que se pretenden con este trabajo de tesis como es el caso de la lectura, corte y reconexión del servicio.

1. Conceptos teóricos

1.1. Relé

1.1.1. Concepto de Relé

El relé es un dispositivo electromecánico, mediante el cual se pueden controlar potencias altas a través de potencias mucho más bajas. Este dispositivo, la mayoría de los tipos, funcionan por medio de un electroimán, los cuales accionan varios contactos que permiten abrir y cerrar circuitos eléctricos de mayor potencia.

1.1.2. Tipos de Relés

- Relés electromecánicos
- Relé de estado sólido
- Relé de corriente alterna

- Relé de láminas

1.1.3. Relé de estado sólido

El relé de estado sólido, es un dispositivo que como su nombre lo indica utiliza un sistema de interruptor compuestos de elementos de estado sólido, como transistores o tiristores, en lugar de contactos mecánicos normales, con el fin de conmutar cargas de mayor potencia y a mayor velocidad. Además de las ventajas anteriores, cuenta con un sistema de aislamiento para el circuito de control, lo que evita sobre corrientes en el circuito de control.²



Figura 2. 1: Relé de estado sólido de 60 Amperios

1.2. Microcontrolador

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de

² Relé de estado sólido con triac. En línea. Internet. Disponible: <http://www.forosdeelectronica.com/proyectos/rele-estado-solido.htm>. Consulta: 5 de Noviembre de 2012

Entrada/Salida, es decir se trata de un computador completo en un solo circuito integrado.⁹

“Básicamente el microcontrolador es un circuito programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica”¹⁰. El pic en otras palabras es un pequeño chip que une varios pequeños componentes electrónicos.

1.2.1. Pic 18f4550

Es un microcontrolador de la familia de los 18f de 40 pines. En la siguiente tabla se detalla las características de este pic y las ventajas y desventajas frente a otros pics de la misma familia, con lo cual se justifica la utilización del mismo para la elaboración de este trabajo.

⁹ MICROCONTROLADOR. 2007. En línea. Internet. Disponible: < <http://es.wikipedia.org/wiki/Microcontrolador>>. Consulta: 10 de Noviembre de 2012.

¹⁰ MICROCONTROLADOR. En línea. Internet. Disponible: < <http://microcontroladores-e.galeon.com/>>. Consulta: 10 de Noviembre de 2012.

CARACTERÍSTICAS	PIC18F2455	PIC18F2450	PIC18F4455	PIC18F450
Frecuencia de Operación	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz	Hasta 48MHz
Memoria de Programa (bytes)	24.576	32.768	24.576	32.768
Memoria RAM de Datos (bytes)	2.048	2.048	2.048	2.048
Memoria EEPROM Datos (bytes)	256	256	256	256
Interrupciones	19	19	20	20
Líneas de E/S	24	24	35	35
Temporizadores	4	4	4	4
Módulos de Comparación/Captura PWM (CCP)	2	2	1	1
Módulos de Comparación/Captura PWM mejorado (ECCP)	0	0	1	1
Canales de Comunicación Serie	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART	MSSP,EUSART
Canal USB	1	1	1	1
Puerto Paralelo de Transmisión de Datos (SPP)	0	0	1	1
Canales de Conversión A/D de 10 bits	10 Canales	10 Canales	13 Canales	13 Canales
Comparadores analógicos	2	2	2	2
Juego de Instrucciones	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)	75 (83 ext.)
Encapsulados	PDIP 28 pines SOIC 28 pines	PDIP 28 pines SOIC 28 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines	PDIP 40 pines QFN 40 pines TQFP 40 pines

Tabla 2. 1: Características Microcontroladores

(DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD DE VALENCIA.
MICROCONTROLADOR PIC18F4550.) [Ref. 5 de Noviembre de 2012]

Fuente: <http://es.scribd.com/doc/57978444/PIC18F4550>

1.2.2. Distribución de pines del pic 18f4550

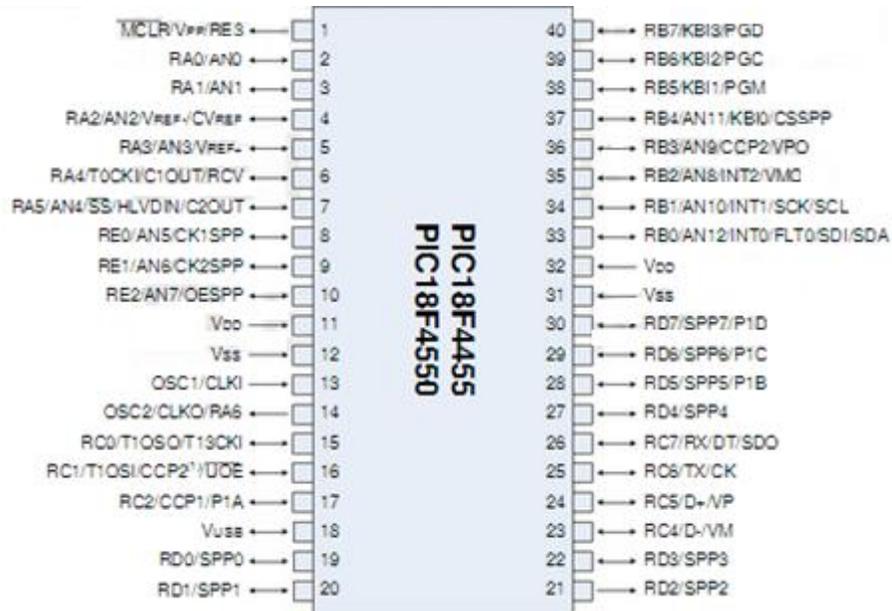


Figura 2. 2: Pinaje del pic 18f4550

(DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD DE VALENCIA.
 MICROCONTROLADOR Pic18F4550.) [Ref. 5 de Noviembre de 2012]

Fuente: <http://es.scribd.com/doc/57978444/PIC18F4550>

1.3. Módulo de comunicación Xbee

Los módulos XBee son una solución potente y eficiente con referencia a lo que son las comunicaciones inalámbricas para transmitir y recibir datos. Estos sistemas, se puede encontrar en diversas aplicaciones industriales y comerciales, como sensores remotos, control y manipulación de robots, control de equipos y automatización.

1.3.1. Concepto de Xbee

Los módulos XBee son módulos de radio frecuencia que nos permiten enviar y recibir datos de manera inalámbrica. Estos módulos remplazan el cable RS232 cumpliendo con la misma función. Trabajan en la banda de 2.4 GHz con protocolo de comunicación zigbee, que no es más que un conjunto estandarizado de soluciones a sistemas inalámbricos. ZigBee está basado en el estándar IEEE 802.15.4 para redes inalámbricas de área personal y tiene como objetivo las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

1.3.2. Tipos de Xbee

- Xbee serie 1
- Xbee serie 2
- Xbee serie 2B
- Xbee pro 900

1.3.3. Características Xbee Serie 2B

Para el desarrollo de este trabajo de tesis hemos optado por utilizar el xbee Serie 2B, ya que presenta las mejores propiedades para el sistema que estamos por realizar. A continuación mostraremos las principales características de este componente detallando ciertas ventajas.

- Funciona a la frecuencia libre de los 2.4GHz
- Posee un alcance máximo con línea de vista de 1,5 Kilómetros
- Sin línea de vista llega hasta los 90 metros

- Cumple con el estándar IEEE 802.15.4
- Tiene una sensibilidad de recepción de -102dBm
- Potencia de transmisión de 63mW
- Voltaje de operación: 2.7V – 3.6V

1.3.4. Distribución de los pines del módulo XBee y tarjeta Xbee

La tarjeta xbee en si es un dispositivo programable que posee un microcontrolador. En cuanto a lo que se refiere a comunicación inalámbrica, este es bastante potente, pero, el resto de sus aplicaciones son limitadas y es por eso que tan solo se utilizó el puerto serial inalámbrico que éste contiene. La distribución de pines del xbee se muestra en la siguiente figura.



Figura 2. 3: Distribución de Pines Xbee S2B

(CONFIGURACIÓN Y CONCEPTOS BÁSICOS XBEE S2B. 2012.) [Ref. 6 de Noviembre de 2012]

Fuente: <http://plataformaszigbee.blogspot.com/2012/05/practica-1-configuracion-y-conceptos.html>

Como vemos en la figura 6 el xbee cuenta con 20 pines, de los cuales utilizaremos 4, la alimentación VCC ubicada en el pin 1, la salida de datos OUT que está en el pin 2, la entrada de datos IN, tomada en el pin 3 y tierra que está en el pin 10. Estas tarjetas se

conectan a un módulo regulador de tensión llamado xbee explore, las mismas que cuentan con pines de salida para ser acopladas a cualquier otra tarjeta. Este módulo nos sirve para regular la tensión a 3,3 voltios que es lo que necesita exactamente el xbee para funcionar correctamente, caso contrario existen errores en la comunicación entre los xbee, se pierden datos, su funcionamiento no es óptimo.



Figura 2. 4: Tarjeta reguladora de voltaje y distribución de pines

(CONFIGURACIÓN Y CONCEPTOS BÁSICOS XBEE S2B. 2012.) [Ref. 6 de Noviembre de 2012]

Fuente: <http://plataformaszigbee.blogspot.com/2012/05/practica-1-configuracion-y-conceptos.html>

1.4. Fuente de Corriente Continua

Se denomina fuente de corriente continua, o también conocido como generador ideal de corriente, aquel elemento de circuito que provee una corriente sin variaciones en el voltaje y corriente, siendo cualquiera la tensión que se le suministre. En las siguientes ilustraciones se puede diferenciar la tensión alterna de suministro (Figura 2.5) con la salida de una fuente de corriente continua (Figura 2.6)

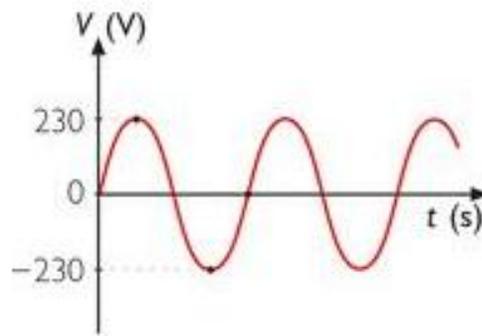


Figura 2. 5: Fuente de corriente Alterna

(CORRIENTE CONTINUA. 2010.) [Ref. 6 de Noviembre de 2012]

Fuente: <http://introduccionelectronica.blogspot.com/2010/06/corriente-continua.html>



Figura 2. 6: Fuente de Corriente Continua

(CORRIENTE CONTINUA. 2010.) [Ref. 6 de Noviembre de 2012]

Fuente: <http://introduccionelectronica.blogspot.com/2010/06/corriente-continua.html>

2. Diseño de hardware

Para el desarrollo de este proyecto se utilizó el programa llamado Altium desing, el cual cuenta con las herramientas necesarias que permite diseñar todo tipo de circuitos

electrónicos y también nos sirve para la elaboración de la placa física en donde se colocaran todos los componentes.

2.1. Diseño de la fuente de tensión de 5 voltios

2.1.1. Materiales

- Transformador de 110V a 12V de 60HZ y 3A
- Puente de Graetz de 1 Amperio
- 1 capacitor de 100uF
- 1 capacitor de 10uF
- 1 capacitor de 22pF
- 1 regulador lm7805

2.1.2. Diseño

Para el diseño de nuestra fuente como primer paso fue necesario reducir la tensión de ingreso utilizando un transformador, el cual convierte los 110-120 voltios de alterna proveniente de la red en 12 voltios, con esto logramos tener una tensión que puede soportar cualquier componente electrónico.

A continuación conectamos la salida del transformador a un Puente de Graetz, el cual es un componente que rectificar la onda proveniente del transformador. En el siguiente grafico se puede apreciar la transformación de la onda antes y después del Puente de Graetz.

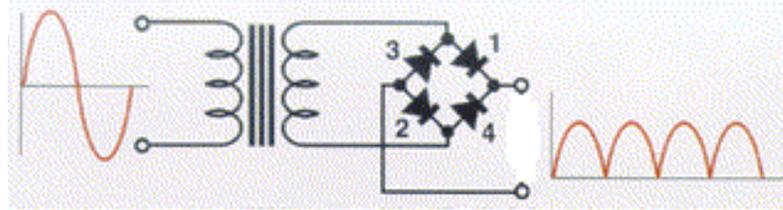


Figura 2. 7: Señales del puente de Graetz

Como se puede apreciar en la figura 2.7, la onda no es completamente constante, por lo que es necesario utilizar un filtro, el mismo que es un circuito formado por un condensador conectado en paralelo a la salida del puente de graetz. Este condensador en paralelo cumple la función de reducir el rizado de la onda que sale del puente. En el siguiente gráfico se puede apreciar la forma de onda que sale del filtro, en este caso está representado por la línea azul.

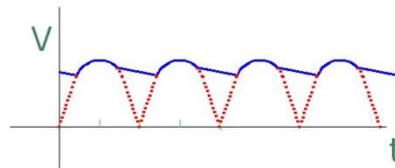


Figura 2. 8: Onda de salida del filtro

Una vez que obtuvimos los 12 voltios de corriente continua, el siguiente paso fue reducir el voltaje a 5 voltios, para ello utilizamos un regulador de tensión de 5 voltios LM7805, el cual estabiliza y transforma la tensión de ingreso de 12 a 5 voltios. Como último paso colocamos un condensador en paralelo a la salida del regulador, con el objetivo de asegurar la estabilidad de la fuente. En la siguiente figura se muestra el diseño de toda nuestra fuente con cada uno de sus componentes.

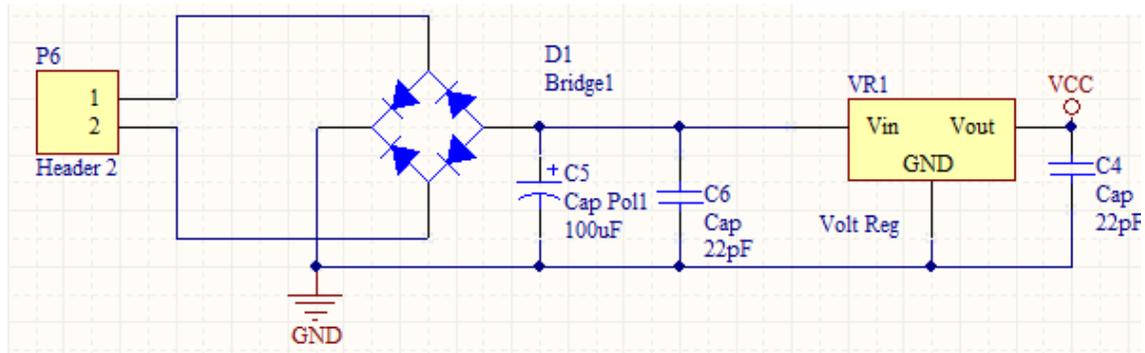


Figura 2. 9: Diseño de la fuente de 5 voltios de corriente continúa

2.2. Conexión del microcontrolador

2.2.1. Materiales

- Pic 18f4550
- 2 condensadores de 33pF
- 1 oscilador de 16MHz
- 1 condensador de 1uF
- 1 pulsante
- 1 resistencia de 4,7k

2.2.2. Diseño

Como primer paso de diseño colocamos el microcontrolador 18F4550, el cual posee una memoria de programación de hasta 32 bytes, una memoria EPROM de 256 bytes, que será en donde se guardará los valores obtenidos de las medidas, un sistema de comunicación USART el cual nos permitirá transmitir y recibir la información con la computadora, 20 líneas de entrada y salida de datos para ejecutar órdenes y hasta 35 interrupciones que nos permitirán hacer uso eficiente de la memoria de programación del microcontrolador.

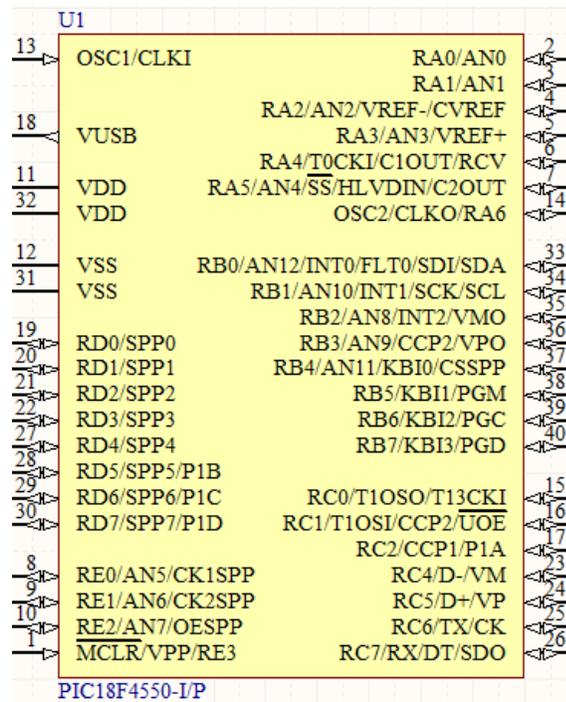


Figura 2. 10: Pic 18f4550

A continuación se muestra en la figura la conexión de componentes básicos del microcontrolador, tales como el oscilador de 16MHz, una alimentación de 5 voltios y un pulsante de reset. El oscilador es un componente que nos indica la velocidad a la que va a trabajar el pic, este está conectado en los extremos a 2 condensadores de 33 pico faradios en los extremos con el objetivo de evitar oscilaciones indeseables. La alimentación de 5 voltios proveniente de la fuente de tensión, nos permitirá energizar al microcontrolador para que este funcione, por último el pulsante de reset que estará conectado al pin 1 del microcontrolador servirá para reiniciarlo en caso de errores.

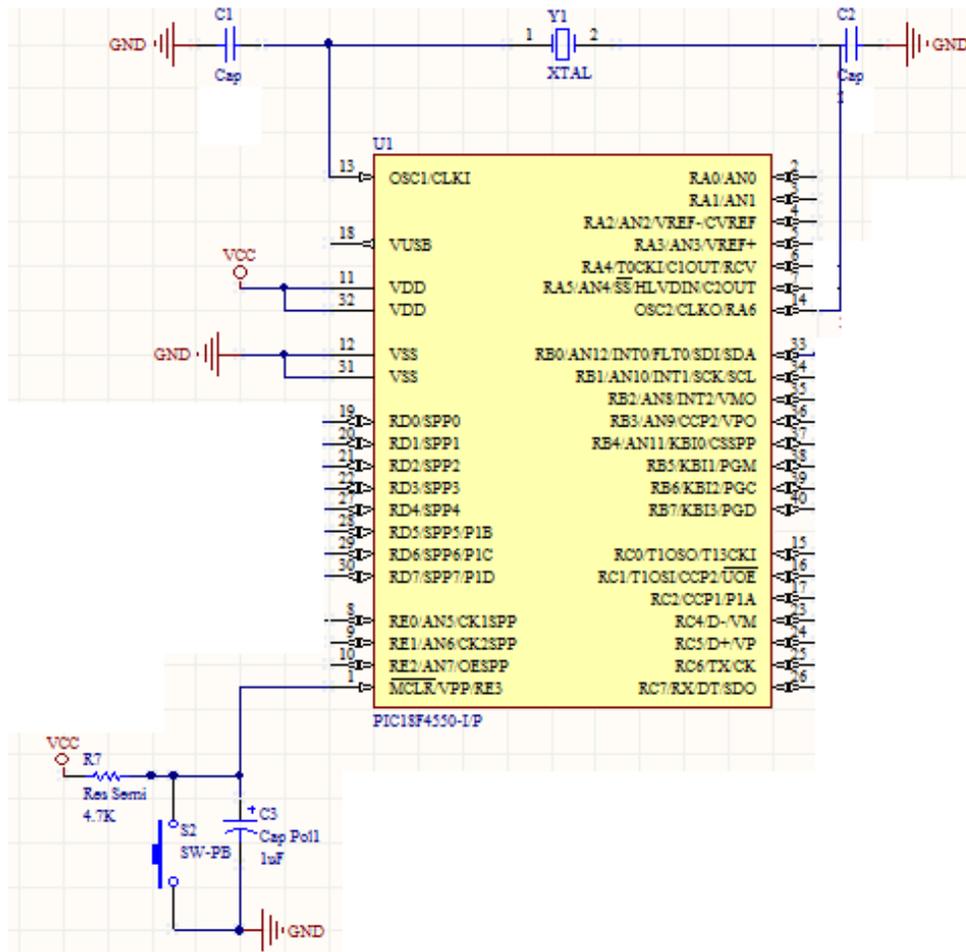


Figura 2. 11: Conexión básica del 18F4550

2.3. Conexión del módulo de comunicación xbee

2.3.1. Materiales

- Peineta hembra para 4 pines
- Módulo xbee
- Tarjeta xbee
- Tarjeta reguladora para xbee

2.3.2. Diseño

Para la conexión del xbee, se requiere utilizar el puerto serial del pic, es por eso que se inicia por conectar el microcontrolador al módulo de comunicación xbee mediante el pin de transmisión y recepción serial, con la finalidad de poder enviar y recibir los datos por el modulo inalámbrico.

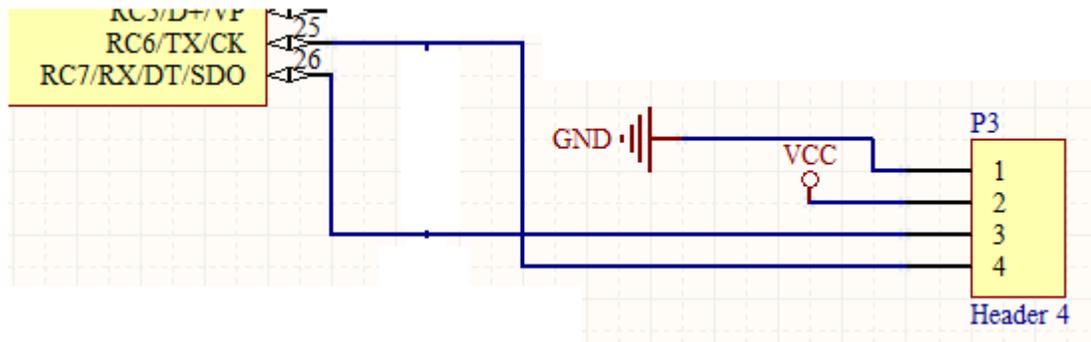


Figura 2. 12: Conexión de un Xbee

En la figura 2.12 se puede observar la conexión del pin 25 de transmisión y pin 26 de recepción del microcontrolador hacia un conector P3 donde se monta el módulo xbee. Podemos observar adicionalmente que el pin 1 y 2 están conectados a tierra y 5 voltios respectivamente, debido a que el módulo xbee requiere de una alimentación adicional para funcionar. Como ya mencionamos previamente el módulo xbee cuenta con un regulador de tensión de 3,3 voltios, que requiere 5 voltios de entrada para funcionar.

2.4. Conexión del LCD

2.4.1. Materiales

- 1 LCD de 2x16
- 1 peineta hembra para 16 pines

- 1 potenciómetro de precisión de 10K
- 1 resistencia de 2.2K
- 1 resistencia de 100
- 1 transistor 2N3904

2.4.2. Diseño

Siguiendo con el diseño, se procedió a conectar un LCD que mostrará el valor de la medida digitalizada y el número del medidor, el mismo que cuenta con 16 pines de conexión y una pantalla de 2 filas de 16 caracteres cada una para mostrar los dos valores que se requieren.

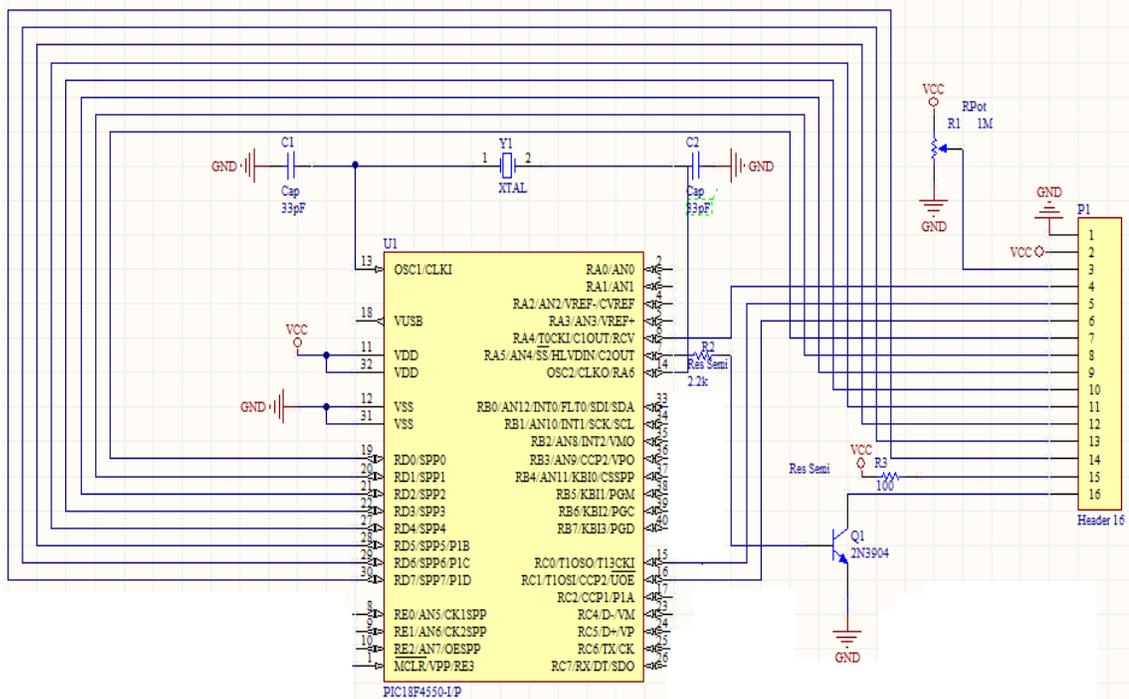


Figura 2. 13: Conexión LCD

Como podemos observar en la figura 2.13, el LCD está representado por el conector P1. Lo primero que se conecta es tierra y 5 voltios a los pines 1 y 2 del conector respectivamente, ya que necesita de una alimentación extra para funcionar. El potenciómetro R1 de 10K se conecta al pin 3, con el objetivo de poder tener control de la intensidad de luz de los símbolos escritos, así mismo conectamos los pines 6, 15, 16 y todo el puerto D del microcontrolador a los pines del LCD, por cuanto por estos pines se envía los caracteres en ASCII a visualizar. Debido a que el LCD cuenta con una luz de fondo o backligh se requierw alimentar con 5 voltios en el pin 15. Por último conectamos el pin 16 al emisor del transistor 2n3904 que en la figura está representado por Q1, el mismo que estará manejado desde el pin 7 del microcontrolador para controlar el encendido y el apagado del backligh.

2.5. Conexión de Relé de Estado Solido

2.5.1. Materiales

- 1 resistencia de 1K
- 1 resistencia de 330
- 1 transistor 2n3904
- Una bornera
- 1 relé de estado sólido de 60A normalmente abierto

2.5.2. Diseño

Para esta conexión se hizo uso de un transistor 2N3904, el mismo que se activara desde el puerto RE0 del microcontrolador para manejar el corte y reconexión del relé. Este

transistor cuenta con las suficientes características básicas para funcionar en este circuito es por eso que se optó por la utilización del mismo.

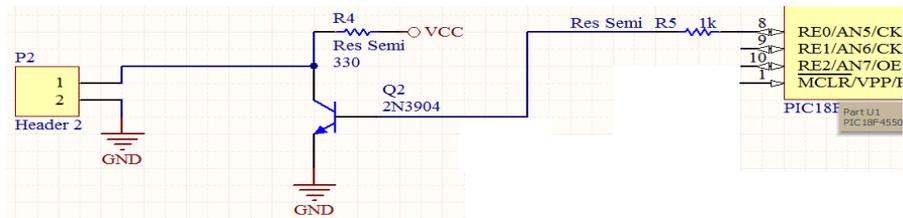


Figura 2. 14: Conexión del Relé de estado sólido

La figura 2.14 muestra un transistor que comanda la señal que va dirigida a la bornera P2, que con sus dos salidas conectará al relé de corte y reconexión de electricidad. Cuando activamos el 2N3904, representado por Q2, el voltaje que alimentaba al relé para que éste se mantenga cerrado se descarga por tierra, produciendo que se desenergice y se abra.

2.6. Conexión del contador de pulsos y pulsantes de sincronización.

2.6.1. Materiales

- 2 pulsantes normalmente abiertos
- 2 resistencias de 220
- 1 transistor 2n3904
- Una peineta macho
- Conector para la señal de pulsos

2.6.2. Diseño

2.6.2.1. Contador de pulsos

Para el conteo de pulsos, fue necesario acoplar la señal de pulsos proveniente de la tarjeta del medidor a una de las entradas del microcontrolador; para ello se utiliza una bornera, la cual estará conectada a la señal que mueve el motor del medidor ciclo métrico y al mismo tiempo al pic.

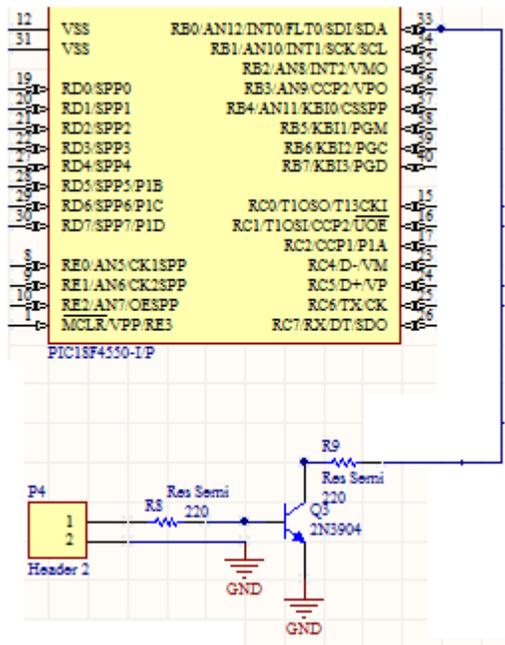


Figura 2. 15: Contador de Pulsos

Si observamos la figura 2.15, el conector P4 representa la bornera, que conecta a una señal de pulsos proveniente de la tarjeta del medidor (esta señal mueve el motor del marcador ciclo métrico del medidor y marca 1/20 de kilovatio/hora) con un transistor 2n3904, que funciona como interruptor para generar pulsos hacia el puerto RB0 del pic.

Los pulsos provenientes del medidor activan la base del transistor Q3. La señal de la base del transistor produce que exista continuidad de corriente entre el colector y el

emisor del transistor, en otras palabras que este se cierre. El transistor está conectado en su colector al pin RB0 del pic y en su emisor a tierra. Como veremos en el siguiente capítulo las entradas del microcontrolador están configuradas para activarse cuando sus pines marquen 0 voltios, o para ser más específicos se conecten a tierra, es por eso que al activarse el transistor Q3, produce inmediatamente que el pin RR0 se active y marque un pulso. En la figura 2.15 se muestra también una resistencia R9 conectada al colector del transistor, ésta se utiliza para proteger la entrada del microcontrolador en caso de corrientes muy altas.

2.6.2.2. Pulsantes de sincronización

Se conectan 2 pulsantes, los cuales permiten sincronizar el valor del medidor híbrido ciclo métrico con el de nuestra tarjeta. El primero de los pulsantes está conectado al puerto RB1, este sirve para aumentar o disminuir manualmente la medida de nuestra tarjeta. El segundo pulsante se conectara al puerto RB2 y servirá como puntero para apuntar el dígito de la medida que se desea variar.

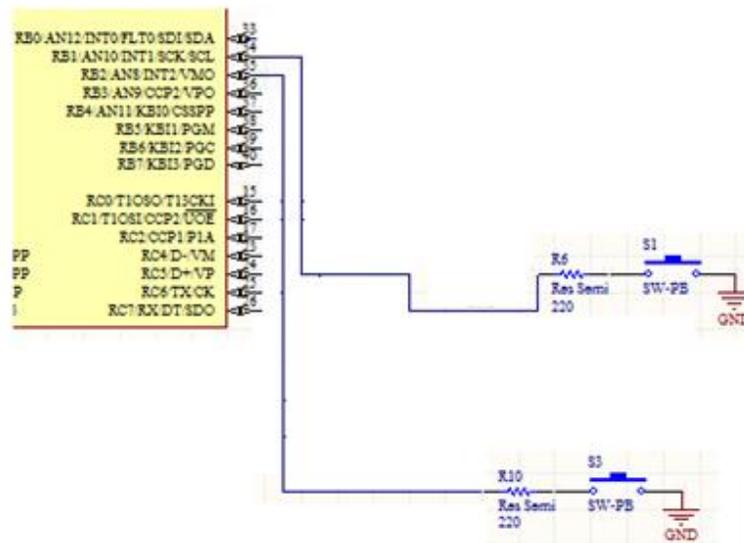


Figura 2. 16: Conexión de pulsantes de sincronización

En la figura 2.16 se aprecia un pulsante S1, que está conectado al puerto RB1, este pulsante sirve para aumentar o disminuir manualmente el valor de cada dígito de la medida de la tarjeta. De igual manera S3, está conectado al puerto RB2, que al activarse permite movernos mediante un puntero sobre cada dígito de la medida. Como sucedía con el contador de pulsos, los puertos de entrada RB1 y RB2 funcionan con pulsos bajos, es por eso que S1 y S2 tienen un lado de su conexión que va hacia tierra. Las resistencias R6 y R10 evitan que se quemen los puertos por sobre corrientes.

2.7. Diseño del circuito Final

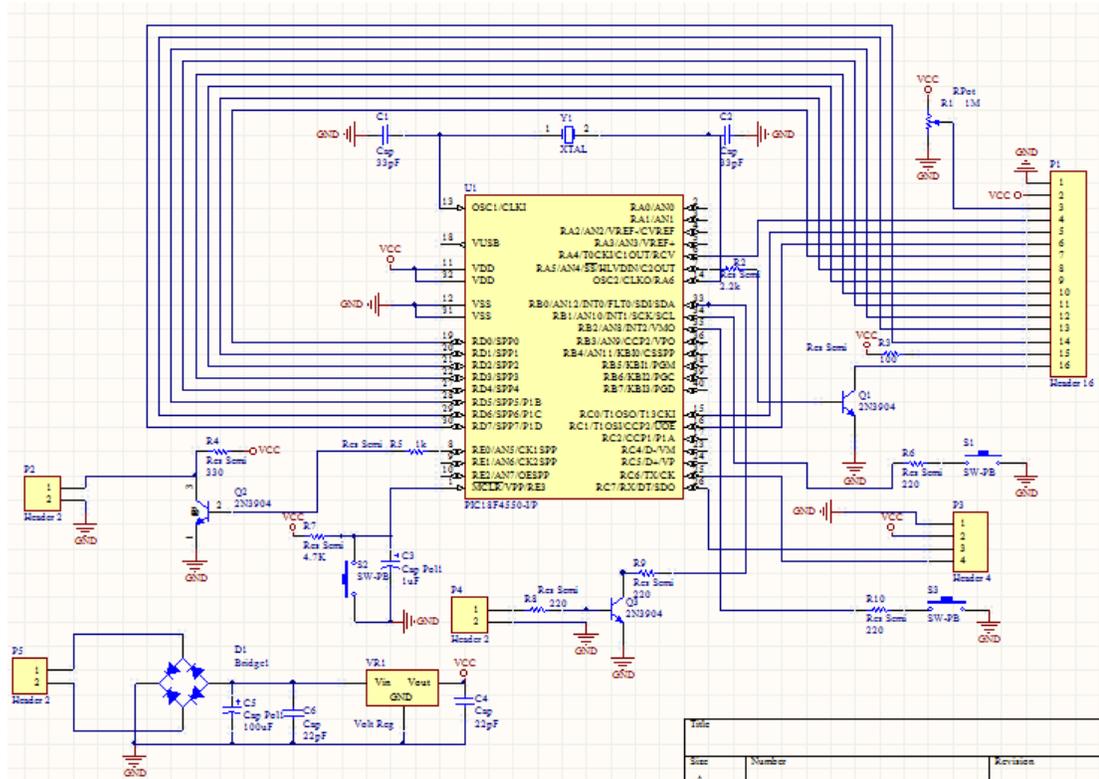


Figura 2. 17: Diseño del circuito final

3. Construcción de la tarjeta

Para la construcción de la tarjeta, primeramente se utiliza la herramienta de Altium desing para crear PCBs, la misma que a partir de un circuito previamente diseñado, construye el mismo sobre una placa virtual. En la siguiente figura se muestra un modelo de la placa con todos sus componentes.

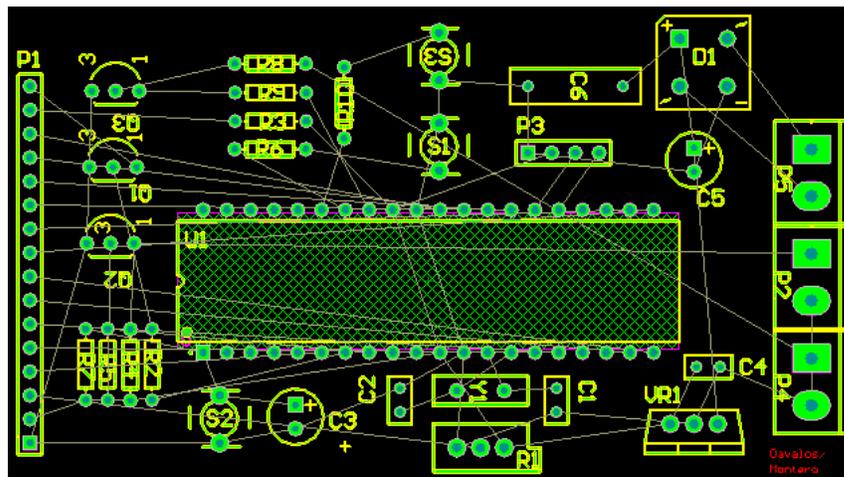


Figura 2. 18: Modelo Placa del Medidor

En la figura 2.18 se muestran cada uno de los componentes que serán impresos sobre la placa y la distribución de los mismos. El siguiente paso fue crear el ruteo para las conexiones, en nuestro caso utilizamos una placa de doble capa, esto quiere decir que existirán dos caminos para cada lado de la placa.

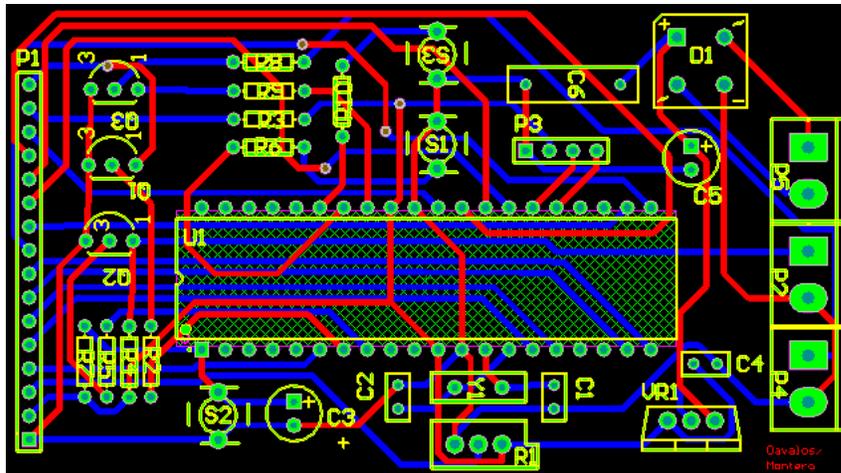


Figura 2. 19: Ruteo de la Placa

Como vemos en la figura 2.19, existen 2 ruteos de diferente color, uno azul y otro rojo, cada uno de estos está sobre una capa distinta por lo que los cruces entre estos no interferirán con el circuito. La ventaja de utilizar placas de doble capa, es que los circuitos pueden ser más compactos y pequeños ahorrando espacio y dinero.

Una vez trazado los caminos o rutas, se visualiza a la tarjeta en 3 dimensiones para tener una idea de cómo iba a quedar físicamente y verificar que cada una de las huellas de cada componente sean las correctas ya que existen diferentes tamaños y formas para cada componente.

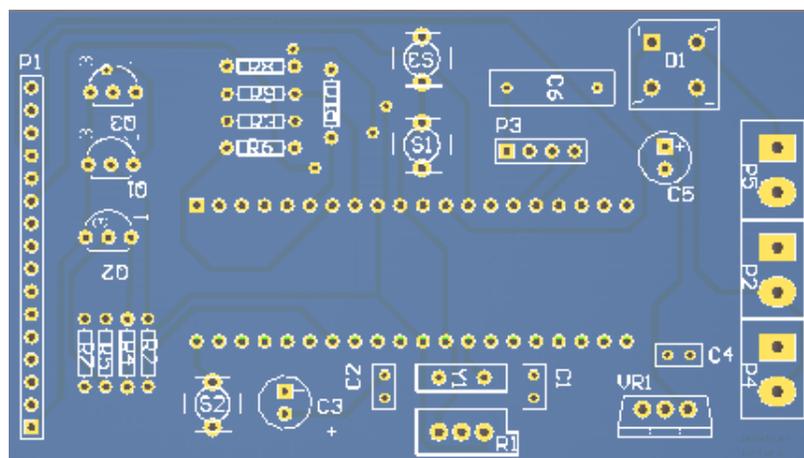


Figura 2. 20: Vista física de la placa

Ya con todo listo se envía la placa a un lugar especializado en donde se imprimen adecuadamente el diseño sobre una placa que cumpla con los requisitos adecuados para que el circuito funcione adecuadamente. De esta forma se obtiene la placa física y se sueldan todos sus componentes.

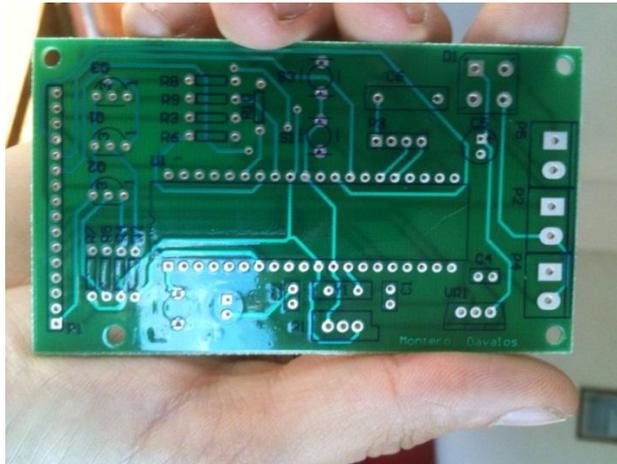


Figura 2. 21: Vista física de la placa



Figura 2. 22: Vista de la placa con todos sus componentes

4. Esquema de montaje del circuito con el medidor

Para alimentar el circuito electrónico, se debe conectar directamente de la red, es decir, paralelo a la entrada del medidor para que el consumo del prototipo no se sume al consumo del cliente. La fase en la salida del medidor debe conectarse al sistema de control de servicio, y los conectores restantes de ambos dispositivos, se deben conectar al hogar.

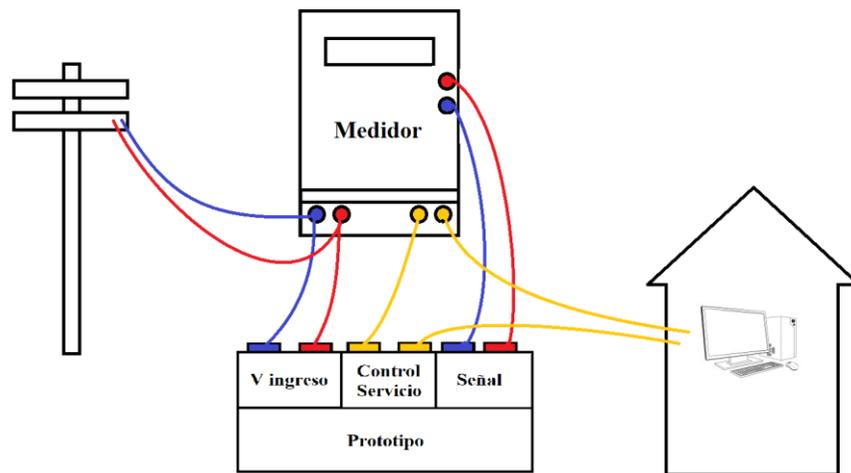


Figura 2. 23: Esquema de montaje del prototipo con el medidor

CAPITULO III

DISEÑO DE SOFTWARE

Introducción

En este capítulo se elabora la parte de programación del proyecto de telemetría y control de los medidores, el mismo que se lo divide en tres secciones. La primera sección de este capítulo describe la elaboración y conexión a una base de datos, ya que, este trabajo de tesis cuenta con un sistema que permitirá almacenar la información adquiridas de cada cliente.

En la segunda parte se detalla la elaboración del programa de interfaz gráfica que interactúa con el usuario y que almacena las medidas dentro de una base de dato. Para ello se usa un programa denominando LabVIEW, el cual cuenta con herramientas que permitirán conectarse con la placa elaborada en el capítulo anterior.

Por otro lado en la tercera parte de este capítulo, se explica el desarrollo de la programación en la placa elaborada en el capítulo 2, la misma que cuenta con un microcontrolador que comanda todas las acciones físicas del proyecto, para su programación se hace uso de un programa llamado mikroC que facilita la programación.

1. Marco teórico

Para entender todos los procesos de programación de este capítulo, a continuación se presenta una breve explicación de cada una de las herramientas utilizadas, tales como lo son: LabVIEW, la base de datos MySQL, MikroC utilizado para programar los microcontroladores y X-CTU que sirve para configurar los Xbee.

1.1. LabVIEW

LabVIEW es una plataforma y entorno de desarrollo para pruebas, control y diseño mediante la programación gráfica. Utiliza un lenguaje llamado lenguaje G, donde la G simboliza Gráfico. Es utilizado principalmente por ingenieros y científicos para tareas como:

- Adquisición de datos y análisis matemático
- Comunicación y control de instrumentos de cualquier fabricante
- Automatización industrial y programación de PACs (Controlador de Automatización Programable)
- Diseño de controladores
- Diseño embebido de micros y chips
- Control y supervisión de procesos
- Visión artificial y control de movimiento
- Robótica
- Domótica y redes de sensores inalámbricos



Figura 3. 1: Pantalla de inicio de programa Labview 2011

1.2. Base de Datos

Es un sistema que nos permite guardar grandes cantidades de información de forma organizada para luego poder acceder a los datos de una manera más rápida y estructurada.

Características

- Independencia lógica y física de los datos
- Redundancia mínima
- Integridad de los datos.
- Consultas complejas optimizadas
- Respaldo y recuperación.

1.2.1. Marcas de Base de Datos

Existen varias marcas de base de datos entre las cuales destacan los siguientes:

- MySQL
- PostgreSQL
- Acces
- Microsoft SQL Server
- Oracle
- Interbase

Para el desarrollo de este prototipo hemos optado por utilizar MySQL, la misma que se describe a continuación:

1.2.1.1. MySQL

Es un sistema de software libre que ayuda en la gestión de base de datos relacional, multiusuario y multihilo, es decir, que realiza varias tareas a la vez.

Características

- Disponibilidad en gran cantidad de plataformas y sistemas
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación.
- Conectividad segura.
- Búsqueda e indexación de campos de texto.

1.3. Enlace de conexión a una Base de Datos con Labview

Para enlazar una base de datos con LabVIEW, se utiliza la herramienta llamada Database Connectivity Toolset, la misma que permite el acceso a la información almacenada en las tablas desde Labview, para así poder utilizarla de manera útil en el proyecto.

Las principales características de esta herramienta son:

Conecta rápidamente a las bases de datos locales y remotos, ejecuta muchas operaciones comunes de base de datos sin tener que realizar lenguaje estructurado de consultas (SQL). Se conecta con las bases de datos más populares, como lo son Microsoft Acces, SQL Server y Oracle. Ofrece también capacidades completas de SQL.

1.4. MikroC

La programación de microcontroladores está basada en lo que se conoce como código de ensamblador, el mismo que conforma cada una de las instrucciones del programa. Al ser este código extenso y de difícil comprensión, ciertos fabricantes, como microchip, optaron por la creación de compiladores de alto nivel, facilitando así la edición y creación de programas.

Es posible adquirir compiladores para pics tales como PIC, CCS, PIC Basic, entre otros. En este trabajo de tesis se optó por la utilización del compilador MikroC, el mismo que está basado en el lenguaje C y soporta pics de la familia de los 18F que es utilizado en este trabajo.

MikroC es un software con una amplia variedad de herramientas y ayudas que facilitan la creación de proyectos. Este lenguaje basado en C, optimiza las operaciones matemáticas, por medio del uso de funciones predefinidas, así como el uso de diferentes tipos de variables como, de tipo carácter, entero y decimal. El compilador crea automáticamente un archivo con código en ensamblador con extensión *.hex, el mismo que permite grabar el programa dentro del pic.

2. Crear Base de Datos

Para crear una Base de Datos como primer paso se debe abrir el Administrador de Datos Fuente o Data Source Administrator ODBC, con lo cual se desplegará una ventana como la de la Figura 3.2. El administrador de Datos Fuente es la herramienta que nos permite ingresar a una base de datos desde cualquier aplicación.

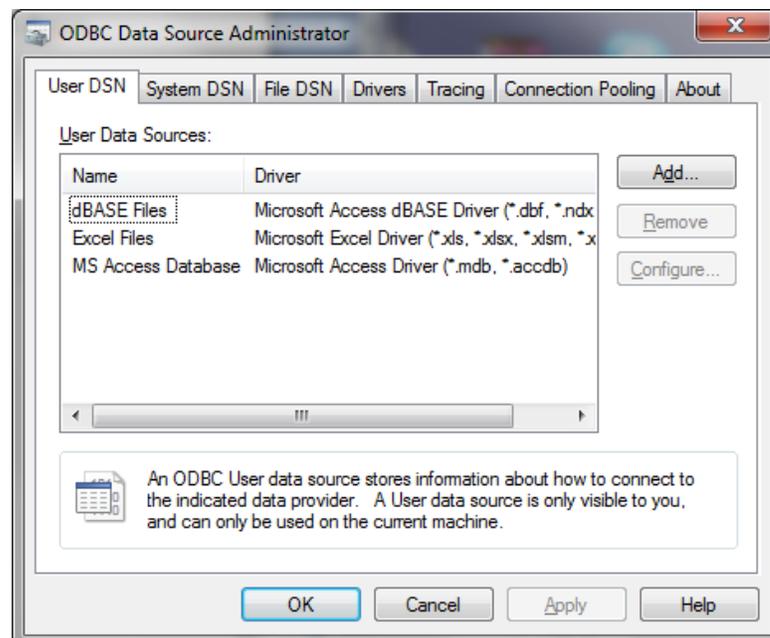


Figura 3. 2: Pagina inicial de ODBC Data Source Administrator

Luego damos clic en el botón agregar (Add) para agregar una nueva fuente de datos, se abrirá otra ventana (Figura 3.3) con una lista de diferentes tipos de datos fuente, en donde escogeremos MySQL ODBC 3.51 Driver, por último damos clic en finalizar (Finish).

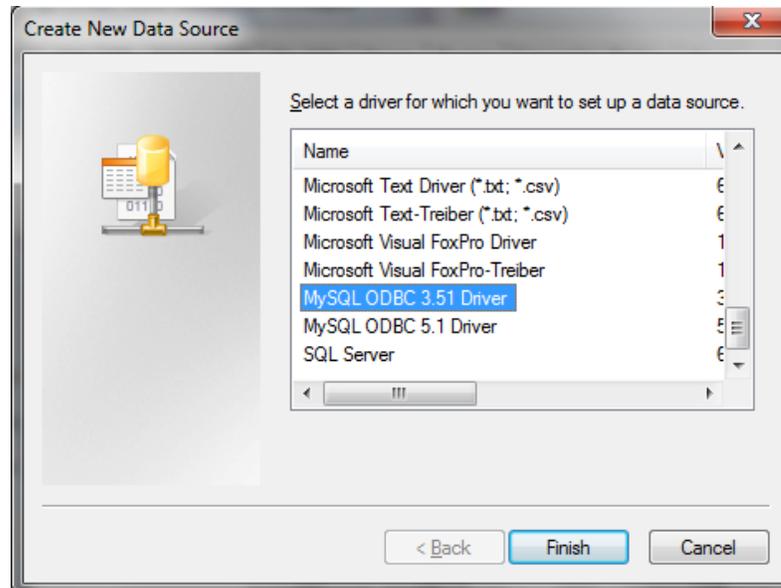


Figura 3. 3: Lista con tipos de datos fuente

A continuación se abrirá una ventana llamada Connector/ODBC 3.51.12 – Add Data Source Name, tal como se muestra en la Figura 3.4, en donde se asigna identificadores para el enlace, tales como, el nombre de la fuente de datos que se utiliza en el desarrollo del prototipo, el Data Source Name, en donde se le da el nombre Tesis, el Server será el localhost, el User root y el Database será tesis, finalizamos con un clic en el botón OK.

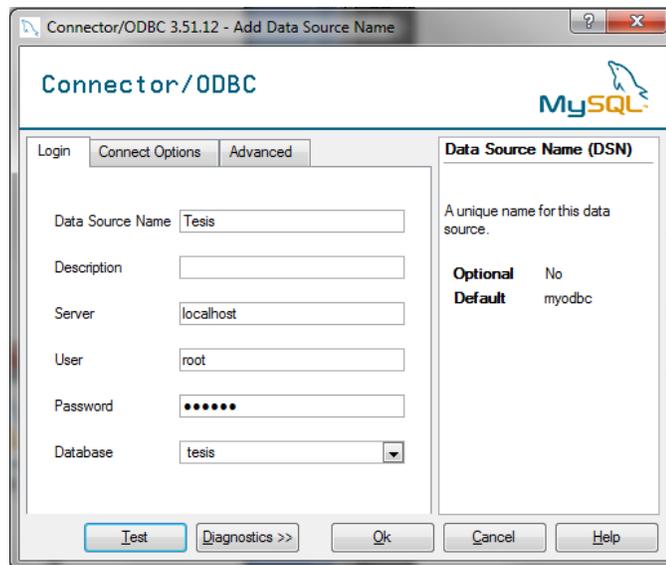


Figura 3. 4: Conector ODBC

En el Panel Frontal de LabVIEW se selecciona la pestaña Tools en donde se escoge la opción Create Data Link (Figura 3.5), que permitirá crear un archivo de extensión .udl (Universal Data Link) que guarda la información de conexión, que se utiliza como enlace con la Base de Datos.

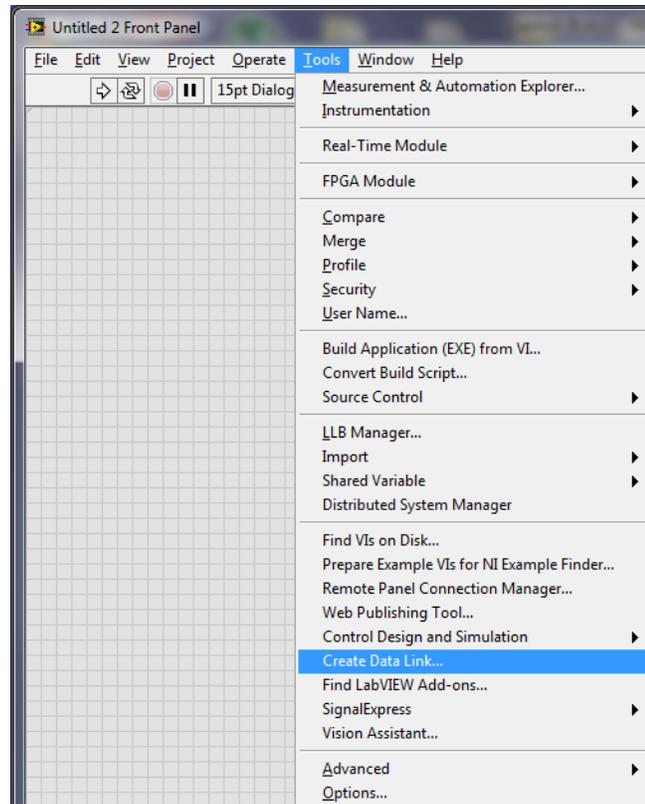


Figura 3. 5: Crear un enlace de datos

Se desplegará una ventana llamada Data Link Properties, como podemos ver en la Figura 3.6, en donde se debe escoger el proveedor de datos, en este caso será Microsoft OLE DB Provider for ODBC Drivers.

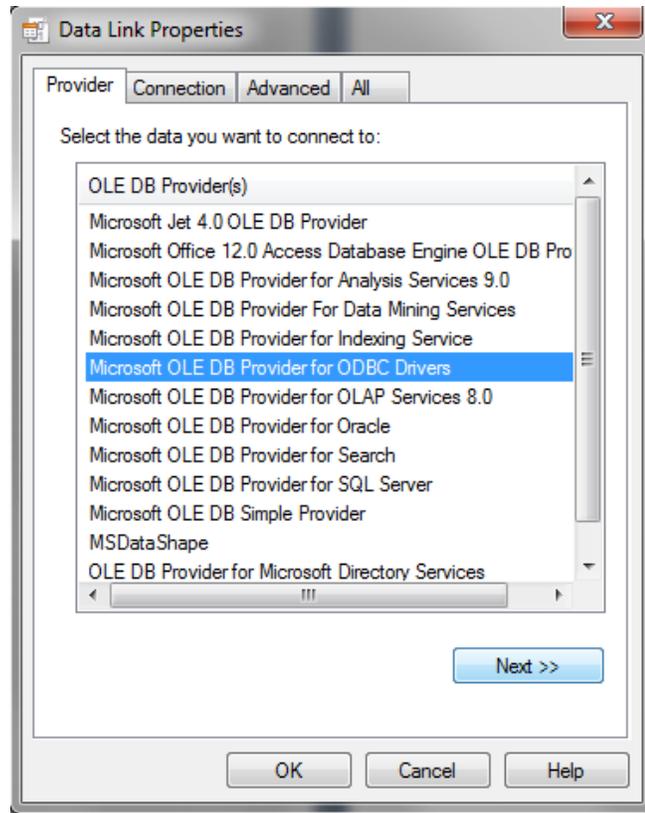


Figura 3. 6: Propiedades del enlace de datos

En la pestaña Connection se escoge el nombre fuente que creamos en un inicio, llamado Tesis, tal como lo realizamos en la Figura 3.7. Luego se da un clic en Test Connection para comprobar que la conexión esté funcionando correctamente como lo muestra la Figura 3.8. Para finalizar el proceso de crear el archivo .udl escogemos el lugar en donde se va a guardar y el nombre que asignaremos a este enlace, para este proyecto lo llamaremos enlace (Figura 3.9).

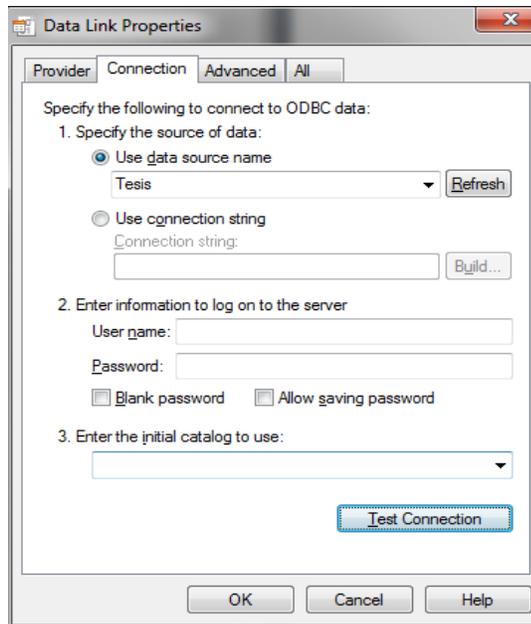


Figura 3. 7: Escoger nombre de la fuente de datos

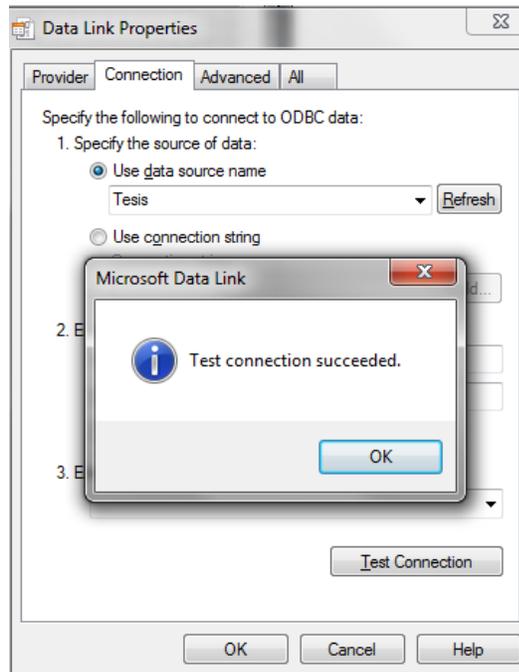


Figura 3. 8: Test de conexión exitosa

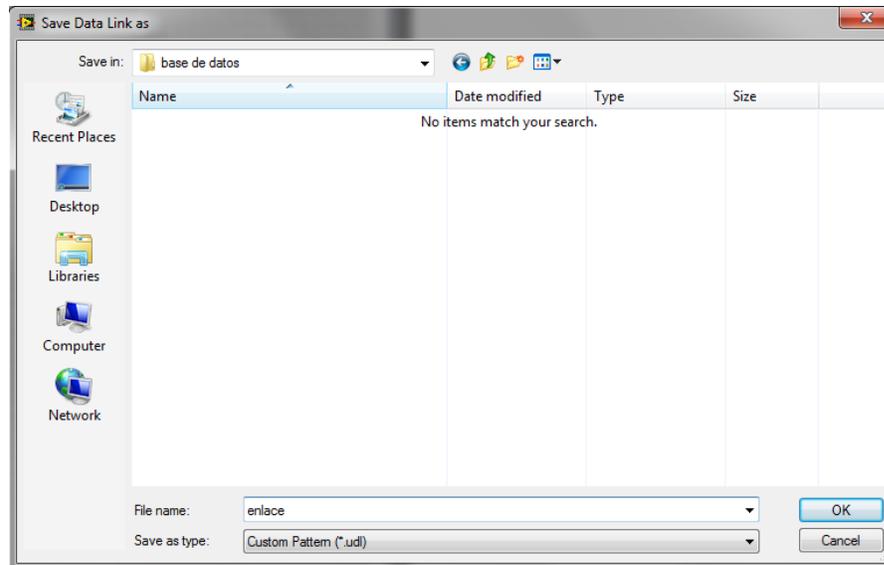


Figura 3. 9: Guardar archivo .udl

3. Programas en LabVIEW

Después de haber creado el archivo con la información que se necesita para la conexión entre la base de datos y el software de monitoreo y control, se procede a desarrollar los programas en LabVIEW que permitirán controlar todas las características que tendrá el sistema.

3.1. Ingresar Nuevo Cliente

Este programa servirá para ingresar un nuevo abonado a la tabla “clientes” de la Base de Datos “tesis”. Los campos necesarios que se encuentran en el Panel Frontal de la Figura 3.10 son: el código del cliente, nombre, apellido, código de medidor y la medida inicial que marca el medidor.

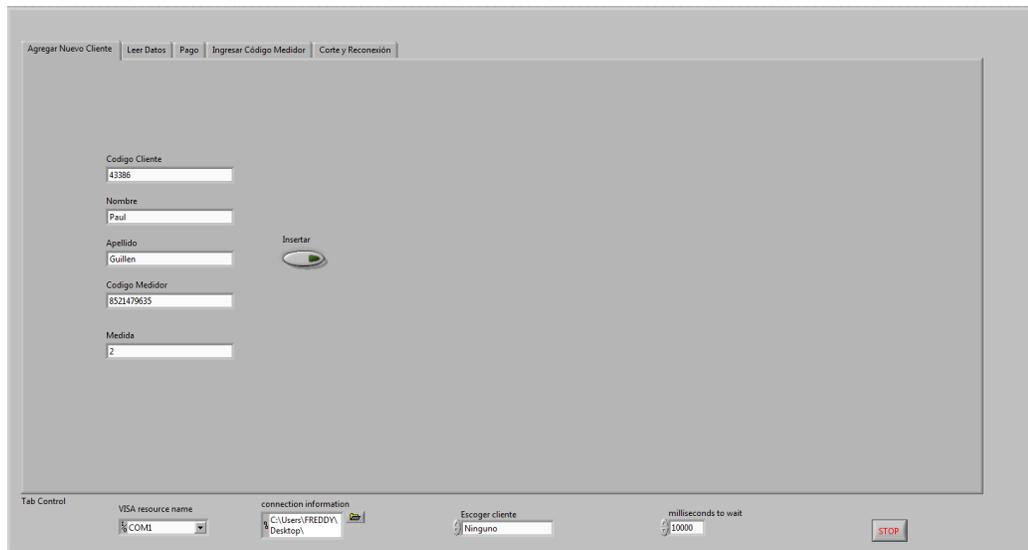


Figura 3. 10: Panel Frontal de Ingresar Nuevo Cliente

Para poder almacenar los datos del Panel Frontal en la Base de Datos se debe iniciar la comunicación entre la Base de Datos y LabVIEW para esto se utiliza la paleta DB Tools Open Connection VI que tendrá como información de conexión el archivo .udl creado en el punto 4 de este capítulo.

Para ingresar una nueva fila en la tabla de la Base de Datos utilizaremos la paleta DB Tools Insert Data VI, en la cual, es necesario cablear el nombre de la tabla, además anexar el nombre de cada columna para cablearlo en la opción columns, y en un cluster agregamos los datos necesarios que se van a ingresar en la tabla de la Base de Datos. Para finalizar se debe cerrar la conexión con la Base de Datos utilizando DB Tools Close Connection VI, como podemos ver en la Figura 3.11.

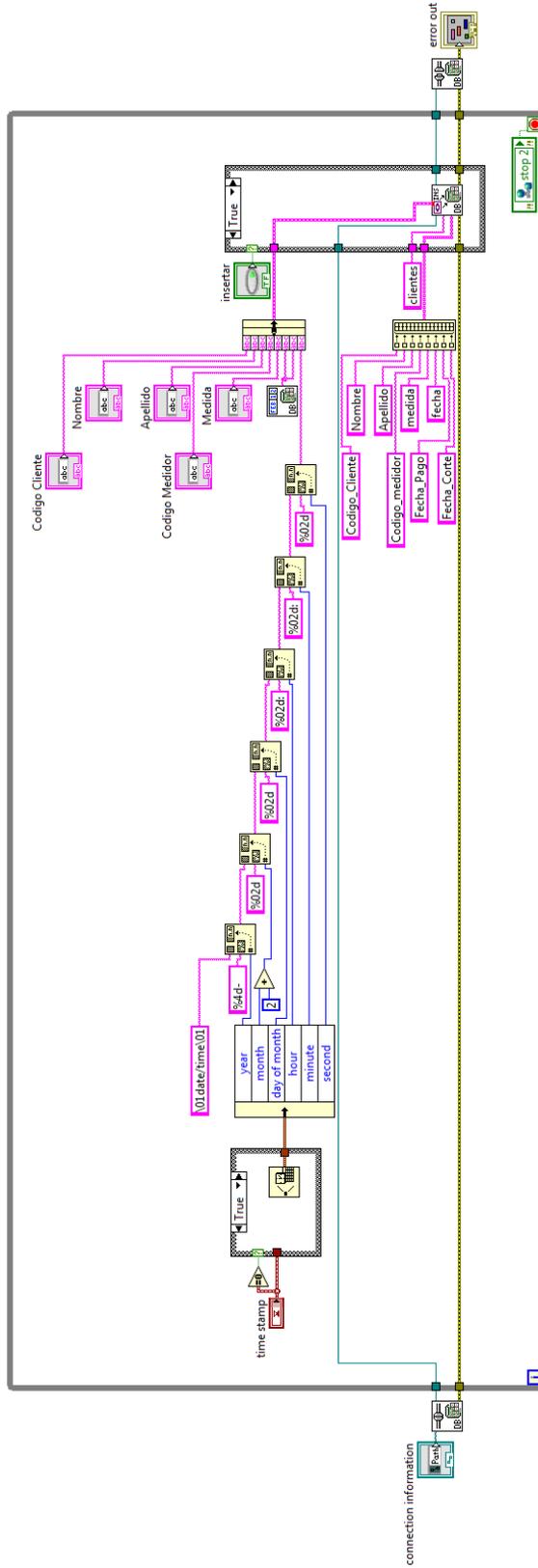


Figura 3. 11: Diagrama de bloques de Ingresar Nuevo Cliente

3.2. Toma de lecturas

El siguiente programa con un Panel Frontal como el de la Figura 3.12, es capaz de obtener la medida al instante de los medidores de forma individual o de todos al mismo tiempo, además tiene la opción de ingresar la hora que deseamos inicie la toma de lecturas secuencial y el lapso de tiempo entre cada medida.

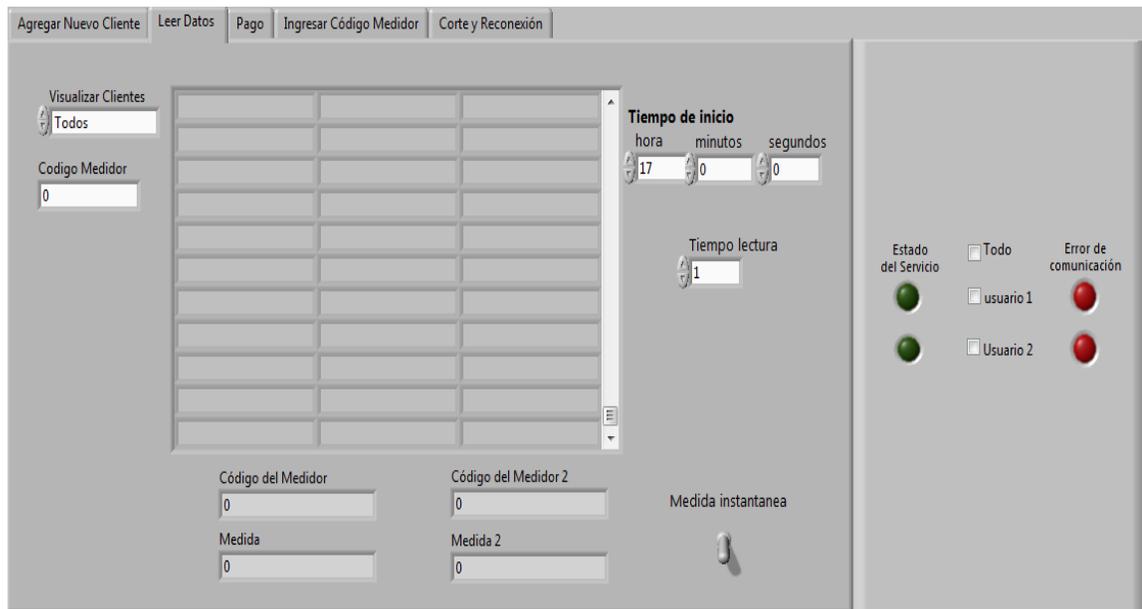


Figura 3. 12: Panel Frontal de Toma de lecturas y control del servicio

Se ha realizado una conexión con las herramientas VISA de LabVIEW (Figura 3.13), las cuales son las encargadas de la comunicación entre el microcontrolador y nuestro programa, de esta manera se puede enviar información del corte y reconexión, además de recibir datos con las lecturas de los medidores. Se envían identificadores para el corte o reconexión del servicio, además funcionan como banderas para escoger el medidor que debe realizar la acción.

Se tiene que decodificar las lecturas que se recibe de los medidores, por lo que, se realiza un cálculo matemático muy sencillo para luego proceder a almacenar la

información en la Base de Datos, el programa realizará el filtro necesario para agregar esa información en el cliente indicado.

Se puede distinguir los siguientes errores en la transmisión de datos:

Error por desincronización, sucede cuando recibimos la trama de datos con los caracteres en desorden, esto origina un error en el perfil de carga. Con un carácter inicial que servirá como bandera, se puede controlar el orden de los datos, puesto que, este siempre irá al inicio de los datos que se reciben.

Error en la comunicación ya sea porque el dispositivo de transmisión está siendo manipulado, o porque el abonado se quedó sin servicio por un error en el servicio eléctrico, en todo caso habrá una notificación para saber cuál es el cliente con este tipo de problemas.

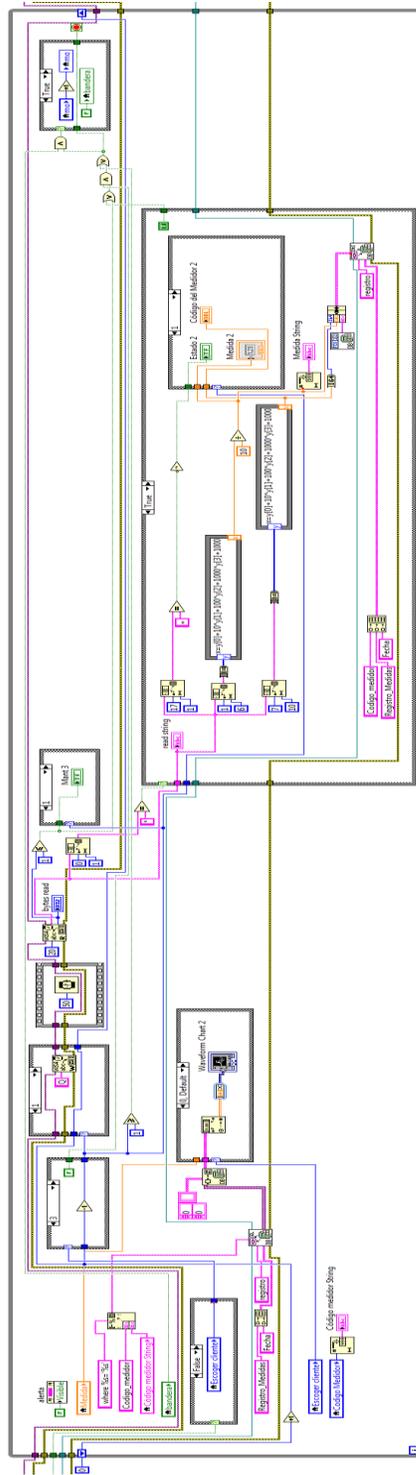


Figura 3. 13: Diagrama de bloques de Toma de lecturas y control del servicio

3.3. Ingresar código de medidor

Para facilitar el reconocimiento de cada medidor es necesario un identificador, por lo que, se ha desarrollado un programa en donde se puede ingresar el código del medidor impreso en su portada, en la memoria EEPROM del microcontrolador de esta manera se puede distinguir cada medidor.

En la Figura 3.14 se tiene una interface de usuario muy sencilla, la cual permite ingresar el código del medidor en forma de un Arreglo, para luego simplemente presionar el pulsante de ingresar código y automáticamente el código será guardado en la memoria EEPROM del microcontrolador.

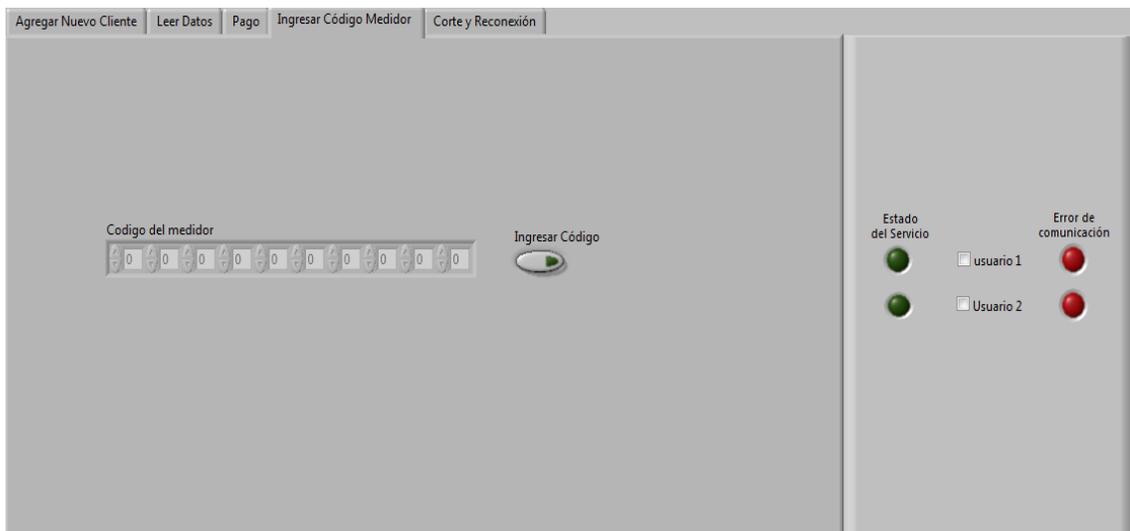


Figura 3. 14: Panel Frontal de Ingresar código de medidor

Para lograr que esto suceda, en el Diagrama de Bloques de la Figura 3.15 se debe definir los parámetros de la comunicación serial utilizando las herramientas VISA de LabVIEW, luego se agrega un identificador que en este caso será la letra C, para que el microcontrolador sepa cuando tiene que decodificar los datos para guardarlos como el código del medidor en su memoria EEPROM.

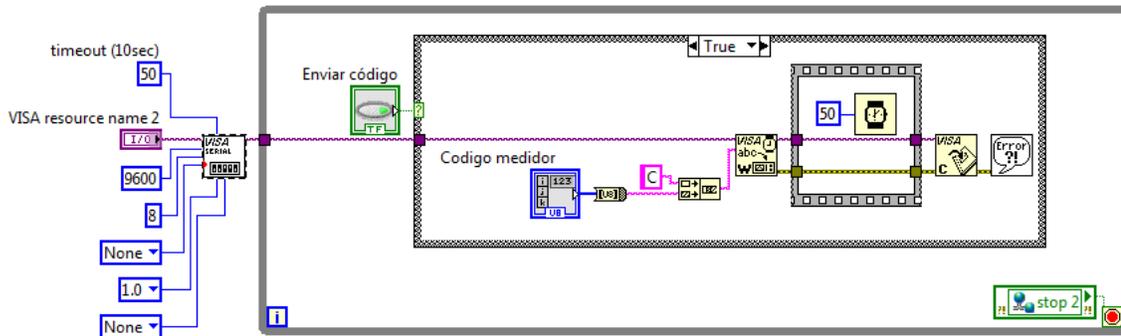


Figura 3. 15: Diagrama de Bloques de Ingresar código de medidor

3.4. Pago por servicio

Dado que la EMPRESA ELÉCTRICA ofrece un servicio pagado, es necesario que, cuando un cliente abone por dicho servicio, se registre la fecha de pago con su medida en ese momento. El modo de operación de este programa es muy sencillo (Figura 3.16), se debe ingresar el código del cliente y del medidor, a continuación se da clic en el botón de Pago y la Base de Datos se actualiza automáticamente.



Figura 3. 16: Panel Frontal de Pago

Se debe realizar un filtro en la tabla clientes de la Base de Datos para ingresar el nuevo valor obtenido de la lectura realizada en ese momento del medidor y la fecha actual en la que se realiza el pago. Además se incluye la fecha de corte en caso de un incumplimiento por parte del abonado. El Diagrama de Bloque del programa Pago por servicio se lo puede visualizar de manera más detallada en la Figura 3.17.

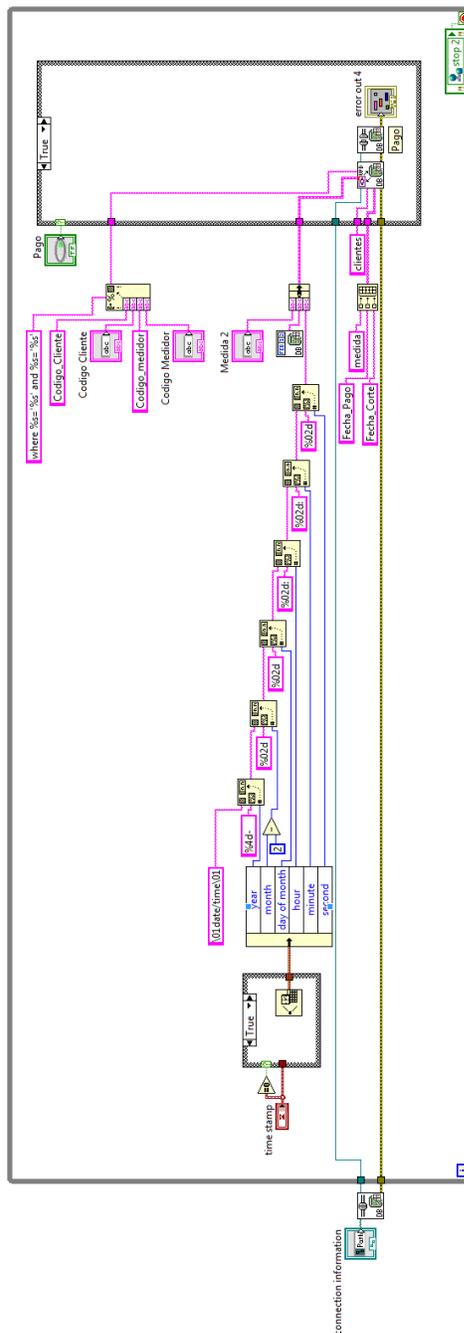


Figura 3. 17: Diagrama de Bloques de Pago

3.5. Corte y reconexión del servicio

La EMPRESA ELÉCTRICA debe realizar el corte del servicio eléctrico a los abonados por el incumplimiento de pago. Cuando un abonado que tiene su servicio de energía eléctrica cortado y cumple con sus deudas, el abonado tendrá nuevamente su servicio eléctrico. Para el corte y reconexión del servicio hemos realizado el programa de la Figura 3.18 en donde podremos seleccionar el medidor al que deseamos realizar la acción requerida o se lo hace a todos de manera simultánea.



Figura 3. 18: Panel Frontal de Corte y Reconexión

Para realizar la acción de corte y reconexión hemos desarrollado el diagrama de bloques de la Figura 3.19, en donde, primero se envía una identificación para saber cuál es el cliente al que se cambiara el estado del servicio, a continuación otro identificador será el que determinará la acción a realizarse, luego se podrá constatar el estado del servicio con los Leds verdes que se encuentran junto a los casilleros de los usuarios de la red.

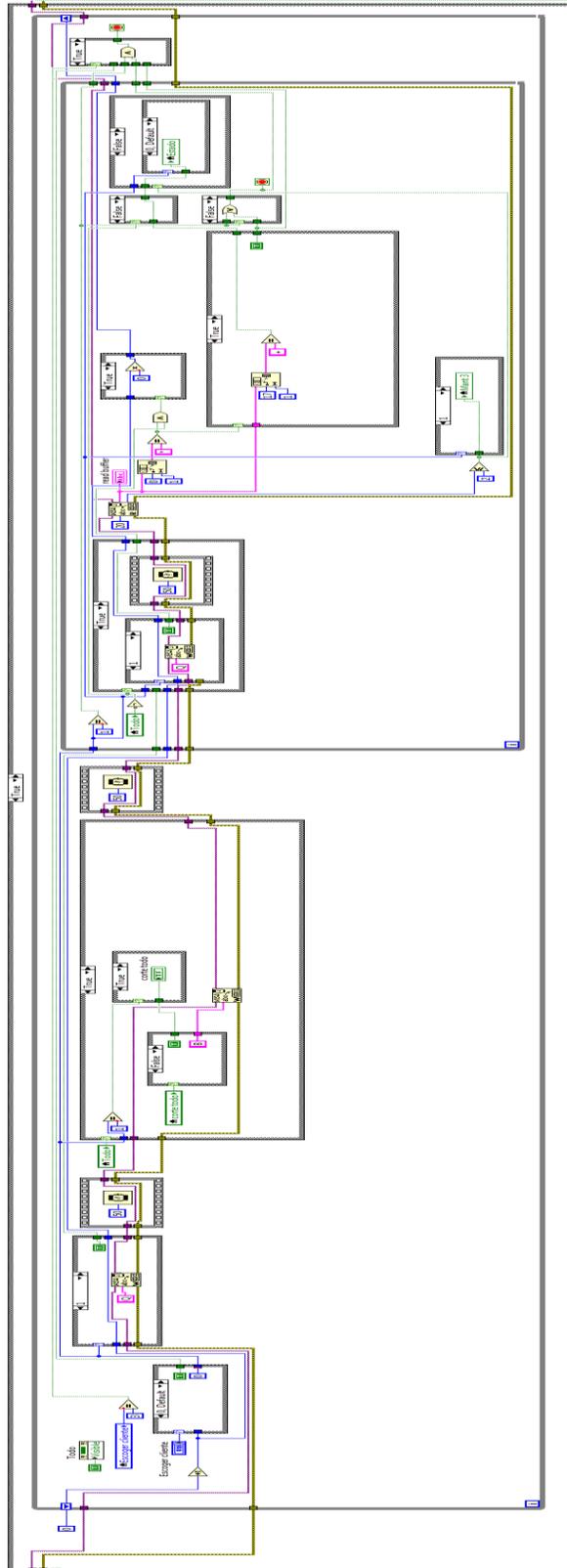


Figura 3. 19: Diagrama de Bloques de Corte y Reconexión

3.6. Visualización de la base de datos

La necesidad de comprobar que las medidas sean correctas y que se están grabando en la base de datos en el tiempo deseado, se ha creado el segmento de programa de la Figura 3.20, para que visualice la base de datos de todos los abonados o se pueda filtrar por código de medidor. La tabla se actualiza cada segundo con lo que podemos comprobar en tiempo real si los datos están siendo almacenados correctamente.

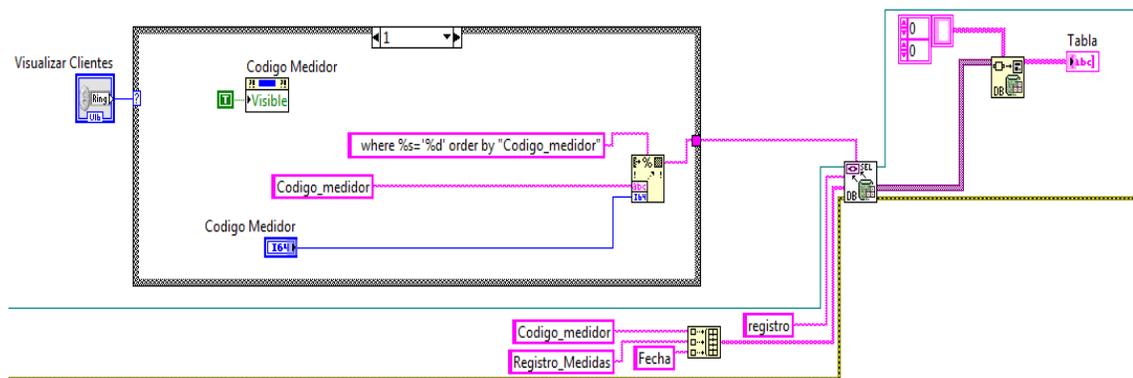


Figura 3. 20: Diagrama de Bloques de Visualización de la base de datos

3.7. Registro de datos por tiempo

La Figura 3.21 muestra el segmento de programa que determina la hora y el lapso de tiempo en el que se registrara una nueva medida, es decir, se ingresa la hora que se desea empiece a registrar las medidas de los medidores, luego se debe escoger el lapso de tiempo, está dado en minutos, se deben realizar cálculos matemáticos sencillos para lograr que el programa no tenga errores conforme avanza el tiempo. Para lograr esto se toma la hora exacta del computador y se realiza una serie de comparaciones, así cuando todas las condiciones se cumplan, se realiza el registro de las lecturas obtenidas por los microcontroladores.

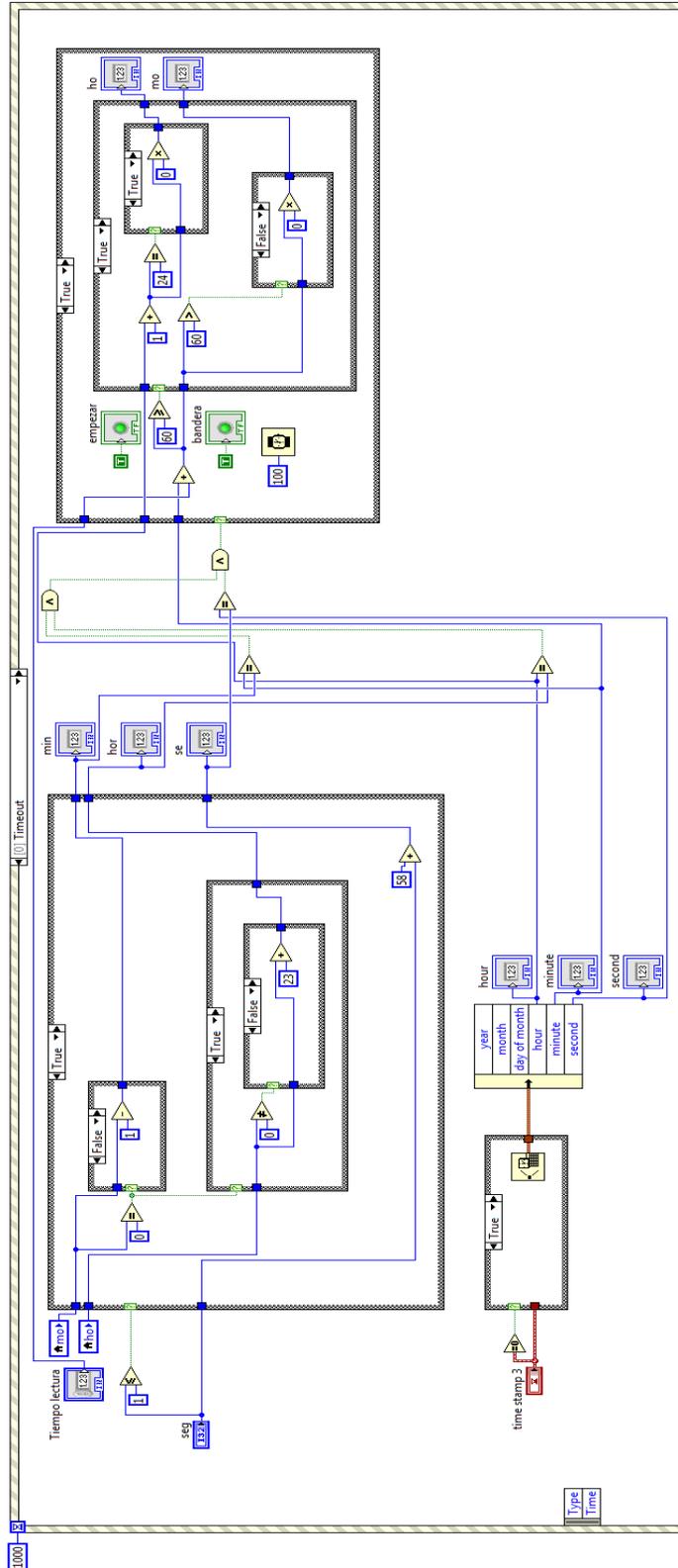


Figura 3. 21: Diagrama de Bloques de Registro de datos por tiempo

3.8. Perfil de carga

Al tener una gran cantidad de medidas almacenadas en la Base de Datos se puede encontrar varias utilidades de los datos almacenados, una muy importante es el perfil de carga, la misma que nos permite conocer el consumo del cliente a lo largo del día, semana, mes, etc., con esto se puede determinar las horas de consumo más críticas de consumo en el día.

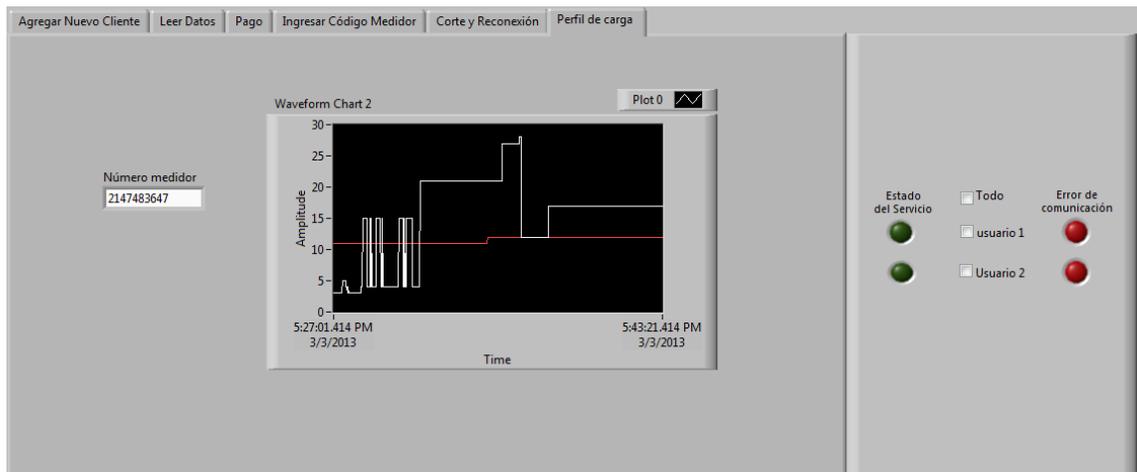


Figura 3. 22: Panel Frontal Perfil de carga

Para poder dibujar el perfil de carga de un cliente, se debe realizar una conexión con la base de datos y filtrar las medidas según el código del medidor, luego debemos transformar la información a números enteros para poder visualizar el perfil de carga del cliente.

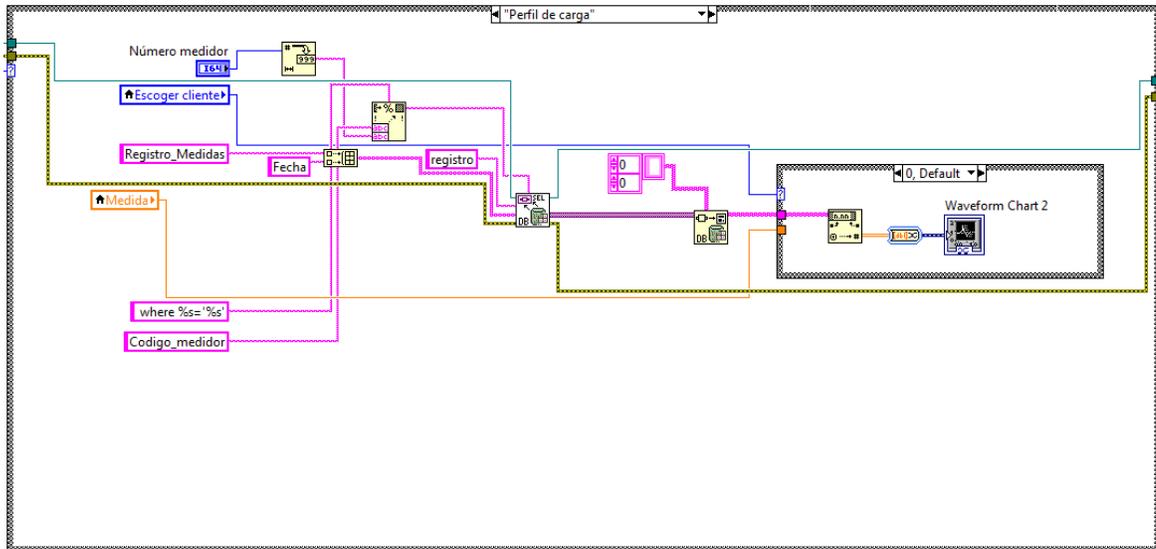


Figura 3. 23: Diagrama de Bloques perfil de carga

4. Programación en el microcontrolador utilizando MikroC

Una vez terminada la programación del entorno gráfico que interactúa con el usuario, se procede a realizar el programa del microcontrolador que se incluye en la tarjeta de este proyecto. El objetivo de este programa es el de controlar todas las acciones físicas de la placa, como es la adquisición de la medida, dar la señal de corte y reconexión, además de configurar en la placa para la comunicación con el computador.

4.1. Declaración de variables

El primer paso que se realiza en el programa es la declaración de variables, las mismas que son un grupo de letras o palabras que nos sirven para guardar y devolver valores dentro del software del pic. Como se muestra en la figura 3.24, hemos optado por utilizar dos tipos de variables, los de tipo entero (int), que sirven para almacenar cualquier tipo de variable numérica, y los de tipo carácter (char), que nos servirán para almacenar valores de tipo ASCII. Como se ve en la figura 3.24 hemos utilizado una

gran variedad de variables; paulatinamente en este capítulo se ira explicando el significado y la función de cada una de estas.

```
//declaracion de variables
unsigned int cont1, cont2, cont3, cont4, cont5, cont6, cont7, bandera, bandera1, bandera2, bandera3, bandera4, bandera5, bandera6, bandera7;
unsigned char contador, contador1, contador2, contador3, contador4, contador5;
unsigned char dato1, dato2, dato3, dato4, dato5, dato6, milisimas, dig1, dig2, dig3, dig4, dig5, dig6, dig7, dig8, dig9, dig10;
unsigned char dato, estado;
```

Figura 3. 24: Declaración de variables

4.2. Conexión del módulo LCD

Como se mencionó en el capítulo anterior, la placa cuenta con un módulo LCD, el cual muestra los valores de la medida y del número del medidor, por cuya razón es necesario establecer una comunicación entre este y el microcontrolador mediante los siguientes comandos que se muestran en la figura a continuación.

```
// Conexion e inicializacion del modulo LCD
sbit LCD_RS at RA4_bit;
sbit LCD_EN at RC1_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at RD7_bit;

sbit LCD_RS_Direction at TRISA4_bit;
sbit LCD_EN_Direction at TRISC1_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;
```

Figura 3. 25: Conexión e inicialización del módulo LCD

Como vemos en la figura 3.25, se hace uso del comando `sbit`, el mismo que permite colocar en un valor de 1 o valor activo a cualquier bit de un puerto del microcontrolador. Lo que hace este segmento de programa es asignar a cada puerto que se está utilizando, las diferentes funciones del LCD que son necesarias para el correcto funcionamiento del mismo. Así se asigna la función de habilitar el LCD al puerto RC1, la del cambio de línea dentro del LCD al puerto RA4 y la de envío de información en ASCII del puerto RD4 al RD7.

4.3. Inicialización del programa principal e inicialización de variables

El siguiente paso es desarrollar el programa principal, mediante el comando `void main` (`void`), el mismo es el bucle principal que se ejecuta todo el tiempo al funcionar el microcontrolador, este repetirá consecutivamente todas las líneas de código mientras el circuito este energizado.

```
void main(void)
{
```

Figura 3. 26: Programa inicial

4.3.1. Inicialización de los puertos

En la siguiente figura se muestra como está configurado cada uno de los puertos a utilizarse.

```

//CONFIGURACION DE PUERTOS
INTCON2.RBPU=0;           // Habilito pull ups del puerto B
TRISB = 0xFF;            // Puerto B configurado como entradas
TRISE = 0;               //Puerto E configurado como salidas
TRISC = 0;               //Puerto C configurado como salidas
TRISD = 0;               //Puerto D configurado como salidas
PORTD = 0;               //Puerto D en cero
PORTE = 0;               //Puerto E en cero

```

Figura 3. 27: Configuración de puertos

Como se puede observar en la figura 3.27, se empieza con habilitar los pull ups del puerto B, colocando en cero el bit 7 del registro INCON2, estos pull ups, son conexiones de cada puerto hacia los voltajes de referencia del microcontrolador, esto permite que el puerto posea un valor lógico de 1; esto solo es utilizado cuando el puerto está configurado como entrada, es decir se configura de esta manera cuando por este puerto se recibirá algún tipo de orden desde el exterior al pic.

Posteriormente tenemos la configuración del registro TRIS de cada puerto, este nos permite colocar a cada puerto mediante un 0 como salida de datos y con un 1 como entrada. Por último inicializamos a los puertos D y E que serán los puertos más utilizados, en 0 como medio de precaución para evitar confusiones posteriores del estado de los mismos.

4.3.2. Inicialización del Puerto Serial USART

Para este proyecto se inicializa el puerto de una manera asíncrona a 9600 baudios de velocidad. A continuación se detalla la configuración del mismo.

```

// EUSART ASICRONO
TXSTA = 0;
TXSTA.TXEN=1;
RCSTA = 144;
SPBRG = 25;
SPBRGH = 0;
BAUDCON.BRG16 = 0;

```

Figura 3. 28: Configuración EUSART

En las dos primeras líneas de configuración se habilita la transmisión del puerto serial, posteriormente se configura la recepción de datos en el registro RCSTA, en el cual se habilita la recepción y se le configura para que esta sea continua. En el registro SPBRG se configura para que este sea de baja velocidad a 9600 baudios, colocando el SPBRG en 25 y el SPBRGH en 0.

4.3.3. Inicialización de las interrupciones

Las interrupciones en el microcontrolador son aquellas que por medio de un evento externo o un tiempo configurado realizar una acción específica dentro del pic. Como vemos en la figura 3.29, las interrupciones básicamente se dan por ocurrencias en los bits del puerto B y por recepción de datos en el puerto serial, ya que en estos se generaran las principales acciones de todo el programa como se verá posteriormente.

```
// INTERRUPTACIONES
INTCON.GIE=1;           //habilita interrupciones
INTCON.INT0IE=1;       //habilita interrupciones en RB0
INTCON3.INT1IE=1;     //habilita interrupciones en RB1
INTCON3.INT2IE=1;     //habilita interrupciones en RB2
INTCON.PEIE=1;        //habilita interrupciones de perifericos
INTCON2.INTEDG0=1;    //interrupcion en RB0 en flanco de subida
INTCON2.INTEDG1=1;    //interrupcion en RB1 en flanco de subida
INTCON2.INTEDG2=1;    //interrupcion en RB2 en flanco de subida
PIE1.RCIE=1;          //Interrupcion por recepcion de dato
IPR1.RCIP=0;
```

Figura 3. 29: Configuración interrupciones

4.3.4. Inicialización del LCD

En esta parte del programa se coloca las condiciones iniciales con las que empieza a funcionar el LCD. Como observamos en la figura 3.30 se inicializa el LCD, después se limpia la pantalla, se apaga el cursor, se prende la luz de fondo y por último se escriben información en el LCD que no se va a alterar durante el resto del programa.

```
//INICIALIZACION DEL LCD
Lcd_Init();           //Inicializamos el LCD
Lcd_Cmd(_LCD_CLEAR); //Limpiar el LCD
Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
PORTA.RA5=0xFF;      //BACKLIGH ENCENDIDO
Lcd_Out(1,1,"Medida:"); // Escribimos un texto fijo en el LCD
Lcd_Out(1,13,".");    //Escribimos el punto decimal del LCD
Lcd_Out(2,1,"Me#");
```

Figura 3. 30: Inicialización del LCD

4.3.5. Inicialización del Variables

En el código del programa es necesario inicializar con un valor a todas las variables que se declararon y que fueron ya mencionadas en este capítulo, ya que se requiere que estas inicien con un valor para que trabajen correctamente en el programa. A continuación se muestra el segmento de programa de cada una de las variables inicializadas con su respectivo comentario el mismo que indica la función que cumplen las mismas.

```

//INICIALIZACION DE VARIABLES
dato=' '; //Dato de Recepcion
estado=0; // estado del rele de corte o reconexion
dig1=0; //1er digito del numero del medidor
dig2=0; //2do digito del numero del medidor
dig3=0; //3ro digito del numero del medidor
dig4=0; //4to digito del numero del medidor
dig5=0; //5to digito del numero del medidor
dig6=0; //6to digito del numero del medidor
dig7=0; //7mo digito del numero del medidor
dig8=0; //8vo digito del numero del medidor
dig9=0; //9no digito del numero del medidor
dig10=0; //10mo digito del numero del medidor
cont1=0; //contador de digitos recibidos del numero del medidor
cont2=0; //contador de milesimas de kW/h
cont3=0; //contador de pulsos para habilitar el puntero
cont4=0; //indicador de ubicacion del puntero
cont5=0; //contador para habilitar la sincronizacion de valores
cont7=0; //contador de pulsos provenientes del medidor
milesimas = 0; // 2do digito decimal de kw/h
contador = 0; //1er dogito decimal de kw/h
contador1 = 0; //1er numero q indica kw/h
contador2 = 0; //2do numero q indica kw/h
contador3 = 0; //3er numero q indica kw/h
contador4 = 0; //4to numero q indica kw/h
contador5 = 0; //5to numero q indica kw/h
bandera=0; // bandera de inicializacion de la eeprom
bandera2=0; //Habilita la recepcion del numero del medidor
bandera3=0; //habilita la lectura de la eeprom para el numero del medidor
bandera4=1; //habilita la trasmision de datos
bandera5=0; //habilita la llegada de datos
bandera6=0; //Habilita la lectura de la EEPROM para el estado del medidor
bandera7=1; //habilita la visualizacion en el LCD

```

Figura 3. 31: Inicialización de variables

4.4. Lectura inicial de la EEPROM

Antes de empezar con la ejecución del programa principal, es necesario leer los datos guardados en la EEPROM, la misma que es una memoria interna del microcontrolador que permite almacenar constantemente valores. Los primeros datos que se leen, son las banderas que indican si existe o no valores almacenados en otros bancos de la

EEPROM. Mediante el comando “EEPROM_Read (x)” se leen un banco de la memoria, donde x en este caso es el número de banco de donde se quiere extraer el valor que está almacenado.

```
//Primera lectura de la EEPROM
bandera=EEPROM_Read(9); //lee bandera para activar o no lectura
bandera3=EEPROM_Read(20);
bandera6=EEPROM_Read(22);
```

Figura 3. 32: Primera lectura de la EEPROM

4.4.1. Lectura de los dígitos de la medida

Lo primero que se hace es preguntar por el valor de la variable “bandera”, la misma que fue leída anteriormente como se indicó en la figura 3.32 e indica si existe o no un valor de medida guardado, esta variable de ser igual a 1, permite la lectura de otro sector de la EEPROM donde está guardados los dígitos de la medida.

```
if(bandera==1) //si bandera = 1 leo la eeprom caso contrario no para el valor de lectora
{
  Delay_ms(1);
  contador = EEPROM_Read(5);
  Delay_ms(1);
  contador1 = EEPROM_Read(6);
  Delay_ms(1);
  contador2 = EEPROM_Read(7);
  Delay_ms(1);
  contador3 = EEPROM_Read(8);
  Delay_ms(1);
  cont2 = EEPROM_Read(23);
  Delay_ms(1);
  contador4 = EEPROM_Read(24);
  Delay_ms(1);
  contador5 = EEPROM_Read(25);
  Delay_ms(1);
}
```

Figura 3. 33: Lectura de dígitos de la medida

4.4.2. Lectura de los dígitos del medidor

De igual manera que el punto anterior se pregunta por una variable ya leída anteriormente, en este caso “bandera3” la misma que indica si es que hay o no un valor de los dígitos del número del medidor, si el valor de “bandera3” es igual a 1, habilita la lectura del número del medidor, caso contrario no lo hace.

```

if(bandera3==1) //si bandera = 1 leo la eeprom caso contrario no para el numero del medidor
{
    Delay_ms(1);
    dig1 = EEPROM_Read(10);
    Delay_ms(1);
    dig2 = EEPROM_Read(11);
    Delay_ms(1);
    dig3 = EEPROM_Read(12);
    Delay_ms(1);
    dig4 = EEPROM_Read(13);
    Delay_ms(1);
    dig5 = EEPROM_Read(14);
    Delay_ms(1);
    dig6 = EEPROM_Read(15);
    Delay_ms(1);
    dig7 = EEPROM_Read(16);
    Delay_ms(1);
    dig8 = EEPROM_Read(17);
    Delay_ms(1);
    dig9 = EEPROM_Read(18);
    Delay_ms(1);
    dig10 = EEPROM_Read(19);
    Delay_ms(1);
}

```

Figura 3. 34: Lectura de dígitos del medidor

4.4.3. Lectura del estado del relé de corte o reconexión

Por medio del valor de la variable “bandera6” al igual que los dos puntos anteriores habilita la lectura de un banco de la EEPROM, en este caso el estado de corte o conexión del relé. Si bandera6 es igual a 1, lee el estado, caso contrario el estado del medidor se coloca con el valor de 0 para que el relé inicialice como conectado.

```

if(bandera6==1) //si bandera6 = 1 leo la eeprom caso contrario no para el el estado de corte o reconexion
{
  PORTE.RE0 = EEPROM_Read(21);
  Delay_ms(1);
}
else
{
  PORTE.RE0=0;
}

```

Figura 3. 35: Lectura EEPROM estado del servicio

4.5. Construcción del bucle principal

Una vez inicializadas todas las variables y hechas las lecturas de la EEPROM, el siguiente paso consiste en la construcción del bucle principal, el mismo que se ejecutara en todo momento mientras el microcontrolador se encuentre alimentado. Dentro de este bucle se realiza la verificación del estado del relé de corte y reconexión, las operaciones que permitirán obtener cada dígito de la medida, el envío de datos y la visualización de los mismos en el LCD.

```

//PROGRAMA PRINCIPAL
while(1)          // Bucle Infinito
{

```

Figura 3. 36: Bucle infinito

4.5.1. Verificación del estado del relé

A continuación se explica el segmento de programa que permite verificar en qué estado se encuentra el relé encargado del corte y reconexión de servicio. En el segmento de programa de la figura 3.37, se pregunta por el estado del puerto RE0, el mismo que como ya se indicó anteriormente es el que envía la señal al relé para cortar o reconecta el servicio. Si este puerto se encuentra con el valor de 0xFF, a la variable “estado” se le

asigna el signo “+”, de estar este puerto en 0, “estado” vendría a tener el valor del signo “-”.

```

while(1)           // Bucle Infinito
{
    if (PORTE.RE0==0xFF)
    {
        estado='+';
    }
    else
    {
        estado='-';
    }
}

```

Figura 3. 37: Bucle principal

En este programa la variable “estado” es indicadora, la misma que nos permite conocer cómo se encuentra el estado del relé que corta y reconecta el servicio, así si ésta tiene asignada el signo “-” indica que el servicio está funcionando correctamente, caso contrario de con el signo “+” el servicio está cortado.

4.5.2. Operaciones para obtener los dígitos de la medida

La siguiente fase del programa consiste en realizar las operaciones correspondientes para obtener los valores de cada dígito de la medida, luego podremos matemáticamente unir cada dígito para obtener la medida real que muestra el marcador ciclométrico. a continuación se muestra el segmento de programa que respalda lo dicho.

```

//Comparasion de datos para ser escritos
if (contador == 10){
.....
    contador1++;
    contador=0;
}
if (contador1 == 10){
.....
    contador2++;
    contador1=0;
}
if (contador2 == 10){
.....
    contador3++;
    contador2=0;
}
if (contador3 == 10){
.....
    contador4++;
    contador3=0;
}
if (contador4 == 10){
.....
    contador5++;
    contador4=0;
}
if (contador5 == 10){
.....
    contador5=0;
}
}

```

Figura 3. 38: Contadores de la medida

Anteriormente ya se mostró que los valores de contador, contador1, contador2, contador3, contador4 y contador5 representan los valores de cada dígito del medidor. Empezaremos por explicar que el valor de la variable “contador”, representa el primer dígito decimal, en el segmento de programa de la figura de arriba, se empieza por preguntar si el dígito decimal ha llegado al valor de 10, de ser verdadera esta afirmación se procede a hacer 0 el valor de “contador” y a aumentar el valor de contador1, el mismo que representa el primer dígito entero de la medida. Posteriormente de la misma manera que con contador, se pregunta si es que “contador1” ha llegado al valor de 10, que de

igual forma de ser cierto se volvería el valor de este a 0 y se aumentara en una unidad al valor de contador2 que es el segundo dígito entero de la medida. Se repite este proceso hasta llegar a contador5, el mismo que es el último dígito de la medida, el cual de ser igual a 10 se hace 0 y se reinicia el valor de la medida.

4.5.3. Envío de datos hacia el concentrador

Antes de enviar los datos se requiere transformar a cada dígito entero de la lectura que se obtiene del medidor en un símbolo en ASCII, para que así el concentrador lo pueda interpretar de una manera sencilla, para esto se realiza lo que se muestra en el siguiente código.

```
milesimas=(cont2) % 10; //segundo digito decimal transformado a ASQUI
dato1=(contador) % 10; //Primer digito decimal transformado a ASQUI
dato2=(contador1) % 10; //Primer digito entero transformado a ASQUI
dato3=(contador2) % 10; //segundo digito entero transformado a ASQUI
dato4=(contador3) % 10; //tercero digito entero transformado a ASQUI
dato5=(contador4) % 10; //cuarto digito entero transformado a ASQUI
dato6=(contador5) % 10; //quinto digito entero transformado a ASQUI
```

Figura 3. 39: Contadores de la medida

Una vez que se tenga todos los datos listos para ser enviados, se procede a hacer la trasmisión de los mismos mediante los comandos TXREG y deley para la demora de envío entre cada dato. A continuación se muestra el código que permite realizar lo antes dicho.

```

//ENVIO DE DATOS
if (bandera4>0)
{
//TRASMISION DE LA MEDIDA
TXREG='*';
delay_ms(1);
TXREG=dato1;
delay_ms(1);
TXREG=dato2;
delay_ms(1);
TXREG=dato3;
delay_ms(1);
TXREG=dato4;
delay_ms(1);
TXREG=dato5;
delay_ms(1);
TXREG=dato6;
delay_ms(1);
}

```

Figura 3. 40: Transmisión de la medida

```

//TRASMISION DEL CODIGO DEL MEDIDOR
TXREG=dig10;
delay_ms(1);
TXREG=dig9;
delay_ms(1);
TXREG=dig8;
delay_ms(1);
TXREG=dig7;
delay_ms(1);
TXREG=dig6;
delay_ms(1);
TXREG=dig5;
delay_ms(1);
TXREG=dig4;
delay_ms(1);
TXREG=dig3;
delay_ms(1);
TXREG=dig2;
delay_ms(1);
TXREG=dig1;
delay_ms(1);
TXREG=estado;
delay_ms(1);
bandera4=0;
}

```

Figura 3. 41: Transmisión de código del medidor

Antes de transmitir se pregunta por el estado de “bandera4” la misma que solo habilita o deshabilita la transmisión, esto con el fin de no enviar en todo momento los datos si no tan solo cuando se necesite. La transmisión empieza con un asterisco, el mismo que se envía con la finalidad de sincronizar el paquete de datos, después se envía cada dígito de la medida y cada dígito del medidor, con una demora de 1 milisegundo entre cada valor con el objetivo de garantizar el envío de cada dato.

4.5.4 Visualización de los datos en el LCD

Para la visualización de datos, se divide el segmento de programa en 2 partes, en la primera parte se pintan los dígitos individualmente considerando la aparición de un puntero para la sincronización de los dígitos de la medida de la tarjeta con el del medidor, mientras que en la otra parte se realiza una visualización completa de todos los caracteres que irán en el LCD. A continuación se explica con mayor detalle lo expuesto.

- Visualización individual con el puntero

```

//VISUALIZACION DE DATOS EN EL LCD
//Visualizacion del puntero para la sincronizacion
if ((cont4==1) && (bandera7==0))
{
    cont6++;
    if (cont6<2)
    {
        Lcd_Cmd(_LCD_UNDERLINE_ON);
    }
    Lcd_Chr(1,15,48+milesimas);
}
if ((cont4==2) && (bandera7==0))
{
    cont6++;
    if (cont6<2)
    {
        Lcd_Cmd(_LCD_UNDERLINE_ON);
    }
    Lcd_Chr(1,14,48+dato1);
}
if ((cont4==4) && (bandera7==0))
{
    Lcd_Chr(1,11,48+dato3);
}
if ((cont4==5) && (bandera7==0))
{
    Lcd_Chr(1,10,48+dato4);
}
if ((cont4==6) && (bandera7==0))
{
    Lcd_Chr(1,9,48+dato5);
}
if ((cont4==7) && (bandera7==0))
{
    Lcd_Chr(1,8,48+dato6);
}
if ((cont4>7 || (cont4==0)) && (bandera7==0))
{
    cont4=0;
    bandera7=1;
    Lcd_Cmd(_LCD_CURSOR_OFF);
}
}

```

Figura 3. 42: Puntero para sincronizar la medida en el LCD

Este segmento de programa se empieza por preguntas sobre el estado de la variable “cont4”, la misma que indica la posición del puntero dentro del LCD para cada uno de los dígitos de la medida además se requiere preguntar por el estado de “bandera7” que habilita la visualización de todos los dígitos del LCD la cual necesitamos que esté en 0 para que este segmento funcione. Así si la variable “cont4” tiene un valor entre 1 y 6 y la

bandera 7 es igual a 0 visualiza al dígito según la posición en la que se encuentre el puntero.

- **Visualización de todos los datos sobre el LCD**

```

//Visualizacion de todos los datos por pulsos del medidor
if (bandera7==1)
{
  Lcd_Cmd(_LCD_CURSOR_OFF);
  Lcd_Chr(1,8,48+dato6);
  Lcd_Chr_CP(48+dato5);
  Lcd_Chr_CP(48+dato4);
  Lcd_Chr_CP(48+dato3);
  Lcd_Chr_CP(48+dato2);
  Lcd_Chr(1,14,48+dato1);
  Lcd_Chr_CP(48+milesimas);
  Lcd_Chr(2,5,48+dig1);
  Lcd_Chr_CP(48+dig2);
  Lcd_Chr_CP(48+dig3);
  Lcd_Chr_CP(48+dig4);
  Lcd_Chr_CP(48+dig5);
  Lcd_Chr_CP(48+dig6);
  Lcd_Chr_CP(48+dig7);
  Lcd_Chr_CP(48+dig8);
  Lcd_Chr_CP(48+dig9);
  Lcd_Chr_CP(48+dig10);
  bandera7=0;
}
}

```

Figura 3. 43: Visualización completa en LCD

En esta parte de código se hace una visualización de todos los datos cada vez que la variable “bandera7” se ponga en 1. Esta bandera solo se activa siempre y cuando exista un pulso de medida, haciendo este proceso logramos que el programa sea más rápido y eficiente.

4.6. Interrupciones por cambio de estado en el puerto B

4.6.1. Interrupciones en el puerto RB0

Cuando existe una interrupción en el puerto RB0, significa que ha existido un cambio de estado en RB0, y esto en el programa como ya se indicó en el capítulo 2 significa que se ha producido un pulso proveniente del medidor, el cual marca un aumento en la medida de la tarjeta y del medidor en sí mismo.

```
//INTERRUPCION POR CAMBIO DE FLANCO EN EL PUERTO
if (INT0IF_bit)
{
    cont7++;
    if (cont7==2)
    {
        cont2++; //CONTADOR DE PULSOS, CUANDO LLEGUE A 10 ES 1/100KWH
        delay_ms(1);
        EEPROM_Write(23,cont2);
        delay_ms(1);
        cont7=0;
    }
    if (cont2==10)
    {
        bandera=1; //HABILITA LA LECTURA DE LA EEPROM
        bandera7=1;
        //ESCRIBIMOS SOBRE LA EEPROM
        EEPROM_Write(5,contador);
        Delay_us(100);
        EEPROM_Write(6,contador1);
        Delay_us(100);
        EEPROM_Write(7,contador2);
        Delay_us(100);
        EEPROM_Write(8,contador3);
        Delay_us(100);
        EEPROM_Write(9,bandera);
        Delay_us(100);
        EEPROM_Write(24,contador4);
        Delay_us(100);
        EEPROM_Write(25,contador5);
        Delay_us(100);
        contador++;
        cont2=0;
    }
    delay_ms(1);
    INT0IF_bit=0;
}
```

Figura 3. 44: Interrupción en el puerto RB0

Como se puede observar en el segmento de programa de la figura 3.44, se empieza por preguntar si ha existido o no una interrupción por el puerto RB0, en caso de ser verdadero aumenta un contador “cont7” el mismo que cuenta el número de pulsos que llegan del puerto, al llegar este a 2, se aumenta en un valor el dígito “cont2”, el cual representa 1/100 de Kw/h. Posteriormente se puede ver que si el valor de “cont2” llega a 10, este produce que se escriban el nuevo valor de medida en la EEPROM y además se habilite la visualización de todos los datos en el LCD.

4.6.2. Interrupciones en el puerto RB1

Una interrupción en el puerto RB1, representa un movimiento del puntero en la pantalla del LCD marcando a cada uno de los dígitos de la medida, además habilita la posibilidad de cambiar manualmente el valor de cada dígito en donde se encuentre el puntero ese momento.

```

if (INT2IF_bit) //esta interrupcion
// nos permite seleccionar el digito de la medida a sincronizar
{
    cont3++; //
    cont7=0;
    if (cont3==2)
    {
        cont4++;
        cont3=1;
        bandera=1; //HABILITA LA LECTURA DE LA EEPROM
        if (cont4>6)
        {
            EEPROM_Write(5,contador );
            Delay_us(100);
            EEPROM_Write(6,contador1);
            Delay_us(100);
            EEPROM_Write(7,contador2);
            Delay_us(100);
            EEPROM_Write(8,contador3);
            Delay_us(100);
            EEPROM_Write(9,bandera);
            Delay_us(100);
            EEPROM_Write(24,contador4);
            Delay_us(100);
            EEPROM_Write(25,contador5);
            Delay_us(100);
            EEPROM_Write(23,cont2);
            Delay_us(100);
        }
    }
    delay_ms(30);
    INT2IF_bit=0;
}
}

```

Figura 3. 45: Interrupción en el puerto RB1

De igual manera que en el punto anterior se pregunta por la bandera de interrupción por el puerto RB1, la misma que de ser verdadera, produce que aumento de la variable “cont3”, que fue creada para evitar pulsos en falso en el puerto RB1 en el momento de energizar el pic, este valor al llegar a 2, genera un aumento en otra variable llamada “cont4”, la misma que como ya se indicó anteriormente en este capítulo representa la posición del puntero en los dígitos del LCD. Por último en el momento en el que “cont4” sea mayor a 6 y el puntero no marque a ningún dígito de la medida, se manda a guardar la medida modificada manualmente en los bancos de la eeprom.

4.6.3. Interrupciones en el puerto RB2

En el momento que existe un cambio de estado o un evento en el puerto RB2; se genera una interrupción en el programa principal, la misma que significa un aumento en el valor del dígito de la medida en el dígito en donde se encuentre apuntando el puntero en la pantalla LCD.

```

if (INT1IF_bit) //Cuando se interrumpe por RB1
//significa un pulso de sincronizacion para la medida
{
cont5++;
if (cont5==2)
{
if (cont4==1)
{
cont2++;
if (cont2>9)
{
cont2=0;
}
}
if (cont4==2)
{
contador++;
if (contador>9)
{
contador=0;
}
}
if (cont4==3)
{
contador1++;
if (contador1>9)
{
contador1=0;
}
}
if (cont4==4)
{
contador2++;
if (contador2>9)
{
contador2=0;
}
}
if (cont4==5)
{
contador3++;
if (contador3>9)
{
contador3=0;
}
}
}
}

```

Figura 3. 46(a): Interrupción en el puerto RB2

```

    if (cont4==6)
    {
        contador4++;
        if (contador4>9)
        {
            contador4=0;
        }
    }
    if (cont4==7)
    {
        contador5++;
        if (contador5>9)
        {
            contador5=0;
        }
    }
    cont5=1;
    delay_ms(20);
    INT1IF_bit=0;
}
}

```

Figura 3.46 (b): Interrupción en el puerto RB2

De igual manera que el punto anterior esta interrupción utiliza un contador de pulsos “cont5”, el mismo que evita acciones en falso a la hora de encender el microcontrolador, así mismo cuando este llega a 2, empieza por aumentar el valor del dígito según el valor de “cont4” el mismo que como ya se mencionó indica la posición del puntero.

4.7. Interrupción por recepción de datos

La interrupción por recepción de datos en el programa es aquella que permitirá recibir la información desde el concentrador hacia el microcontrolador, de esta manera se podrá enviar órdenes al pic desde el computador. Para empezar, se necesita recibir un dato de identificación para poder diferenciar una tarjeta de otra, en otras palabras se requiere recibir un dato en específico por cada tarjeta para que este pueda realizar cualquier tipo de acción enviada desde el ordenador.

```
if (dato=='Q') //realiza acciones en caso de llegar una Q
{
    bandera4++;
    bandera5=1;
    Delay_us(10);
}
```

Figura 3. 47: Condición “Q” en recepción

Como se muestra en la figura 4.47, para la placa se ha utilizado un identificador con la letra “Q”, es decir en el momento que el pic reciba una letra “Q” este pondrá una variable “bandera5” en 1 lo que hará que se active la recepción de datos para realizar acciones en específico sobre esta placa. Una vez que “bandera5” este en 1, se podrá recibir órdenes en específico provenientes del computador, las mismas que se explicaran a continuación con su respectivo código de programación.

4.7.1. Acciones de corte y reconexión

Para el corte y la reconexión hemos utilizado las letras “A” y “B” respectivamente, es decir que cuando se reciba en el pic la letra A, se colocará al puerto RE0 con el valor de 1, lo que hará que el servicio se corte desactivando el relé. Al recibirse la letra B, sucederá todo lo contrario, el puerto RE0 se pondrá en 0 y el servicio se reconectará.

```

if (bandera5==1)
{
  if (dato=='A') //cortar servicio
  {
    bandera6=1;
    PORTE.RE0=0xFF;
    Delay_us(5);
    EEPROM_Write(22,bandera6);
    Delay_us(50);
    EEPROM_Write(21,PORTE.RE0);
    Delay_us(50);
    bandera5=0;
  }
  else
  {
    if (dato=='B') //la llegada de una B habilita el servicio
    {
      bandera6=1;
      PORTE.RE0=0x00;
      Delay_us(5);
      EEPROM_Write(22,bandera6);
      Delay_us(50);
      EEPROM_Write(21,PORTE.RE0);
      Delay_us(50);
      bandera5=0;
    }
  }
}

```

Figura 3. 48: Acción de corte y reconexión

4.7.2. Recepción del número del medidor

Para recibir el número del medidor, por cuestiones de sincronización, el pic recibe como primer valor la letra C, la cual habilita la recepción de los 10 dígitos del número del medidor mediante una variable “bandera2”. Una vez que el pic pueda recibir los datos, lo primero que se hace es preguntar si es que los datos recibidos son valores numéricos, es decir son números del 1 al 9, evitando errores en la recepción. Validado esto, un contador “cont1” empieza a aumentar de uno en uno su valor por cada recepción, esto permitirá que cada dato recibido se guarde en diferentes bancos de la memoria del pic. En la siguiente figura se muestra el segmento de programa de esta recepción con lo que se entenderá de mejor manera lo indicado.

```

else
{
  if (dato=='C') //marca el inicio de la llegada del numero del medidor
  {
    bandera2=1;
    bandera7=1;//habilita la visualizacion en el LCD
    cont1=0; //este numero indica la poscicion del digito del valor del medidor
  }
  if ((bandera2==1))
  { //se valora que los numeros que lleguen sean solo valores
    //numericos y no letras o simbolos.
    if((dato==0) ||(dato==1) || (dato==2) || (dato==3) ||
      (dato==4) || (dato==5) || (dato==6) || (dato==7) || (dato==8) || (dato==9))
    {
      cont1++;
      if(cont1==1)
      {
        dig1=dato;
      }
      if(cont1==2)
      {
        dig2=dato;
      }
      if(cont1==3)
      {
        dig3=dato;
      }
      if(cont1==4)
      {
        dig4=dato;
      }
      if(cont1==5)
      {
        dig5=dato;
      }
      if(cont1==6)
      {
        dig6=dato;
      }
      if(cont1==7)
      {
        dig7=dato;
      }
      if(cont1==8)
      {
        dig8=dato;
      }
      if(cont1==9)
      {
        dig9=dato;
      }
      if(cont1==10)
      {
        dig10=dato;
      }
    }
  }
}

```

Figura 3. 49: Recepción código del medidor

4.7.3. Dato para guardar en la EEPROM

Existen errores de recepción por la cantidad de datos recibidos del número del medidor, por lo que, es necesario para la escritura de los dígitos del medidor sobre la EEPROM, recibir un dígito extra que habilite este proceso, en este caso utilizamos la letra R para realizar esto.

```

if (dato=='R') //por problemas de sincronizacion, en el momento de que llegue una R
//con el numero del medidor este inmediatamente guardara en la memoria EEprom
{
    bandera3=1;
    cont1=0;
    EEPROM_Write(10,dig1);
    Delay_us(100);
    EEPROM_Write(11,dig2);
    Delay_us(100);
    EEPROM_Write(12,dig3);
    Delay_us(100);
    EEPROM_Write(13,dig4);
    Delay_us(100);
    EEPROM_Write(14,dig5);
    Delay_us(100);
    EEPROM_Write(15,dig6);
    Delay_us(100);
    EEPROM_Write(16,dig7);
    Delay_us(100);
    EEPROM_Write(17,dig8);
    Delay_us(100);
    EEPROM_Write(18,dig9);
    Delay_us(100);
    EEPROM_Write(19,dig10);
    Delay_us(100);
    EEPROM_Write(20,bandera3);
    Delay_us(100);
    bandera5=0;
}

```

Figura 3. 50: Guardar datos en la memoria EEPROM

5. Configuración del Xbee

X-CTU es un programa que permite programar de manera fácil y rápida cualquier tipo de modulo inalámbrico de la familia de los xbee. Posee una interfaz gráfica que evita la utilización de comandos AT, estos comandos se definen como el lenguaje de programación para la configuración de los xbee. Es algo complicado entender esta clase

de comandos por lo que el X-CTU evita la utilización de los mismos, haciendo que la programación en los xbee sea más rápida y con menos fallas. Posteriormente se configura los módulos seriales inalámbricos, más conocidos como Xbee, para ello y por facilitar su configuración se utiliza un programa llamado X-CTU, el cual permite programar el xbee de una manera rápida y sencilla.

Para este trabajo de tesis, se utilizaron tres módulos xbee de similares características, de los cuales dos actúan como terminales finales y se conectan a las tarjetas ya antes construidas, mientras que el restante estará conectado al computador y se configura como concentrador.

5.1. Configuración del módulo concentrador

Como primer paso se conecta el módulo xbee al computador y se abre el programa X-CTU teniendo una pantalla como la que se muestra en la Figura 3.51. En el lugar donde se indica select Com Port, se selecciona el puerto serial por donde se ha conectado el xbee, en este caso es el COM5, el resto de configuraciones en esta pantalla no se mantienen igual.

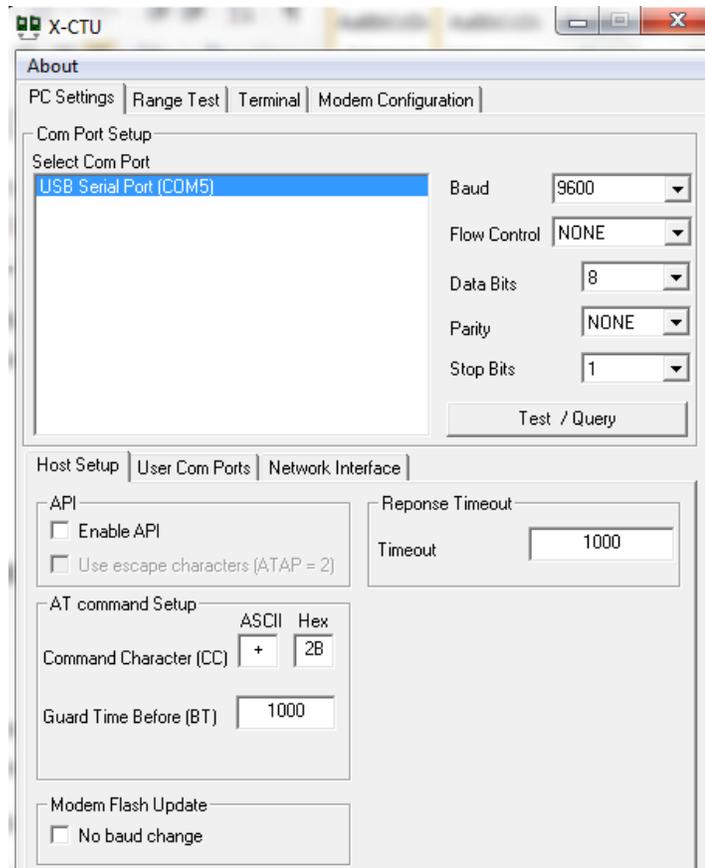


Figura 3. 51: PC Settings X – CTU

A continuación se selecciona la pestaña Modem Configurations, la misma que mostrará las configuraciones del xbee ya conectado. Se presiona en el botón Read, el cual como se muestra en la figura 3.52 lee las configuraciones que tenía guardado el xbee conectado.

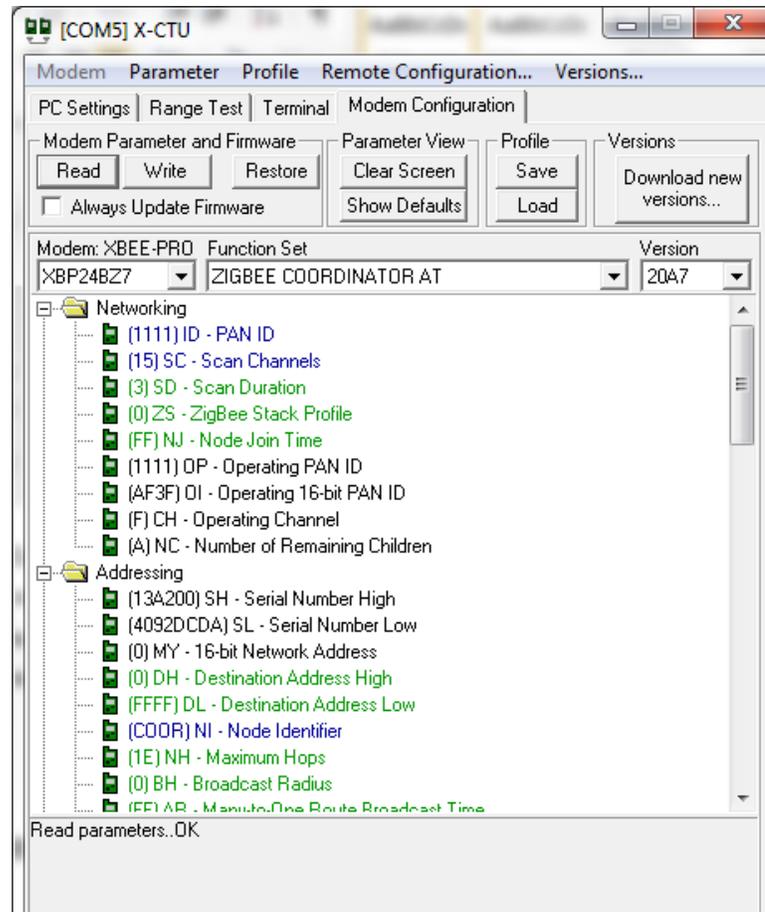


Figura 3.52: Configuración modem ZIGBEE COORNIDADOR AT

En las configuraciones lo primero que se hace es seleccionar la red con que se va a trabajar, para ello se da clic en el lugar que dice PAN ID, y se selecciona la red 1111 la cual tiene que ser igual para todos los Xbee que van a funcionar. Al configurar la red, se obtiene una red mesh, la misma que comunica a todos los xbee entre si y es lo que se conoce como configuración en malla.

A continuación se configuran las direcciones de destino, como vemos en la figura 3.52 existen dos direcciones de destino, una la alta (destination address high) y la baja (destination address low). Como es requerido que el concentrador envíe información de modo broadcast, para lograr esto se colocó a la dirección de destino alta en 0 y a la baja con el valor de FFF. Por último nos dirigimos al lugar donde dice security, aquí se habilita la encriptación de datos.

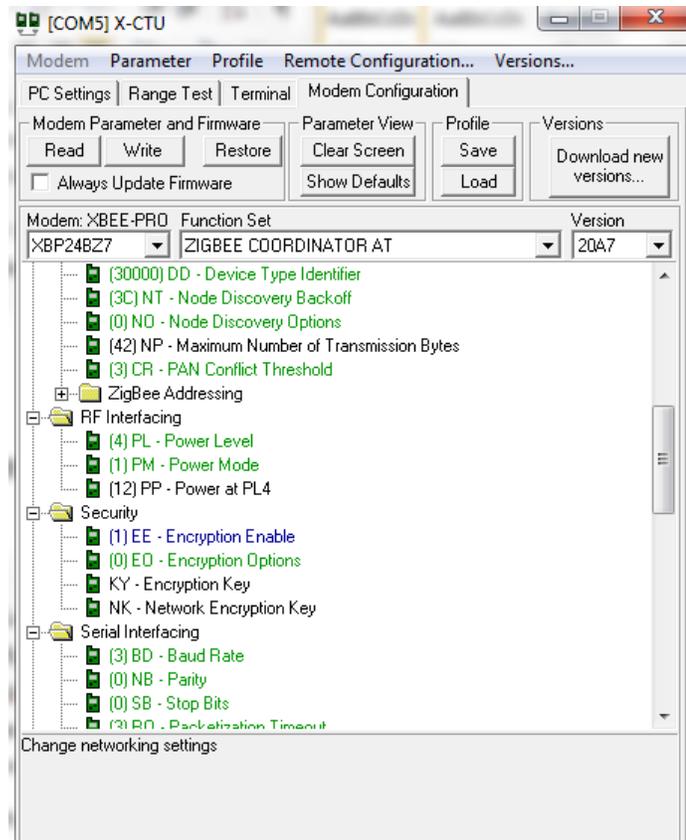


Figura 3. 53: Habilitar encriptación

En la opción de Encryption enable, se digita el número 1, de esta manera se habilita la encriptación de datos, después se da clic en encryption key y aparecerá otra ventana como la que se muestra en la Figura 3.54. En donde se coloca un valor entre 0 y 196 bits en hexadecimal, este número debe ser común para todos los xbee de la red, ya que es usado como una llave que permite recibir y transmitir la información entre los módulos.

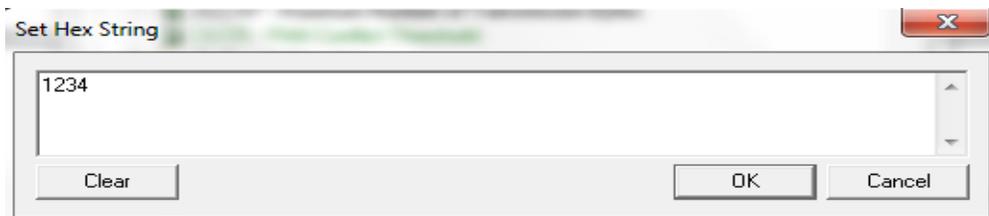


Figura 3. 54: Ingresar la llave de encriptación

5.2. Configuración de los módulos terminales

Estos módulos se configuran de manera similar al módulo concentrador con la única diferencia en las direcciones de destino. Como se observa en la figura 3.55, casi todas las configuraciones son las mismas, la única variante está en la dirección de destino, en donde en vez de colocar el valor de 0 y FFF, en la dirección de destino alta y baja, se coloca el número de serie del xbee que actúa como concentrador, así todos los xbee terminales enviarán la información solo al concentrador.

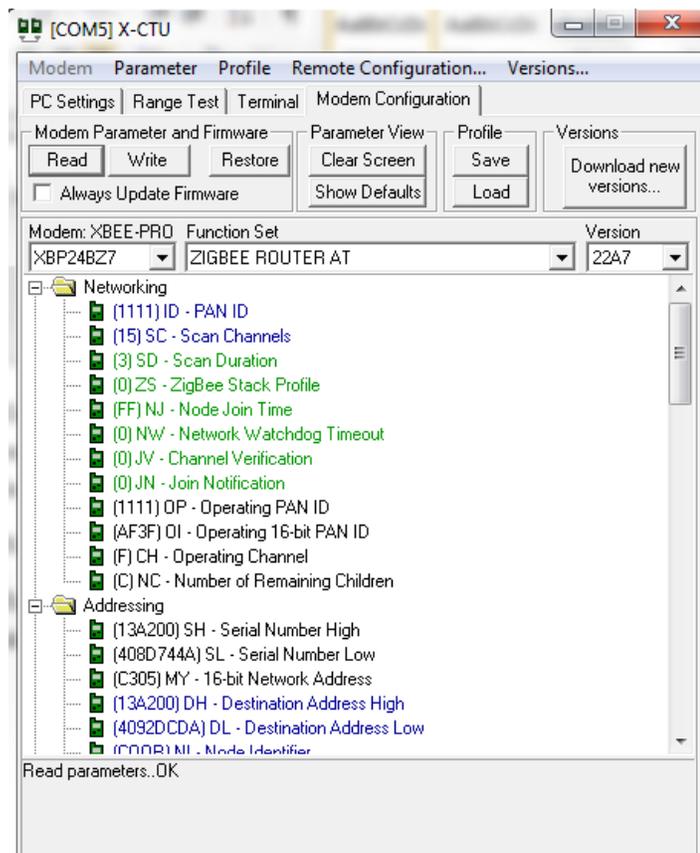


Figura 3. 55: Configuración Modem ZIGBEE ROUTER AT

CAPITULO IV

PRUEBAS DE FUNCIONAMIENTO

1. Introducción

Se ha desarrollado las correspondientes pruebas de funcionamiento, las mismas que se realizaron tanto en un protoboard como en la correspondiente tarjeta electrónica final, además se realizaron las pruebas de software tanto para el microcontrolador como para el programa en el computador.

El protoboard es una herramienta que nos permite armar circuitos sin necesidad de soldar los mismos, por esa razón se creyó conveniente empezar a probar el circuito con este método. Una vez visto el buen funcionamiento con tal herramienta, se hicieron las pruebas definitivas con la tarjeta diseñada.

2. Pruebas de hardware en el protoboard

2.1 Funcionamiento de la fuente de 5 voltios

Como paso inicial se probó la fuente de tensión continua de 5 voltios, la cual tiene que estar totalmente estabilizada para que no existan dificultades de funcionamiento en el resto del circuito. Para demostrar que la fuente está funcionando correctamente se colocó un multímetro en paralelo a la salida de la misma.

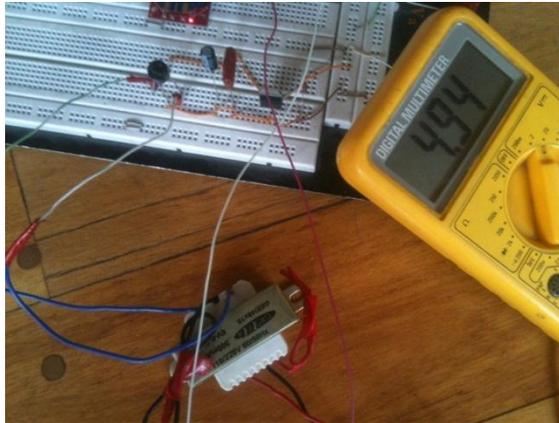


Figura 4. 1: Funcionamiento de la Fuente de tensión de 5 voltios

Como se muestra en la figura 4.1 el valor de tensión que nos proporciona la fuente es de 4.94V, el cual es un valor muy aproximado a los 5 voltios que necesitamos para alimentar el circuito, este es valor es suficiente para que el circuito funcione correctamente y no exista ningún error.

2.2. Funcionamiento del microcontrolador y del LCD

Para demostrar que el microcontrolador está funcionando correctamente se conectó una pantalla LCD, se verifico si se prende y es capaz de mostrar la lectura que marca el registrador ciclométrico y el número de medidor.



Figura 4. 2: Funcionamiento del microcontrolador y LCD

Como vemos en la figura 4.2 el LCD nos muestra la correspondiente lectura de consumo eléctrico y el número de medidor, de esta manera se ha comprobado que el microcontrolador está funcionando de manera adecuada, por lo que se puede garantizar la veracidad de los datos.

2.3. Funcionamiento del Xbee

Para comprobar que el xbee está funcionando, se realizaron dos pruebas, la primera consiste en verificar que el led rojo del módulo se prenda, esto indica que este está correctamente alimentado, la otra prueba y la más importante consistió en verificar que el led de color verde este prendido tanto en el xbee conectado al computador como en el que está conectado al microcontrolador, esto indica que existe una correcta comunicación entre los módulos y que funcionan correctamente.

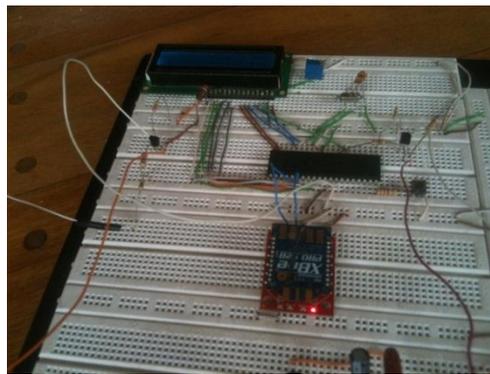


Figura 4. 3: Funcionamiento de la alimentación del Xbee



Figura 4. 4: Funcionamiento de la comunicación del xbee 1



Figura 4. 5: Funcionamiento de la comunicación del xbee 2

La figura 4.3 muestra al xbee alimentado correctamente, lo cual lo comprobamos por el LED rojo que está encendido. Por otro lado en la figura 4.4 y 4.5 se puede constatar que existe una comunicación correcta entre la tarjeta y el computador por que los dos xbee tienen sus leds de color verde prendidos.

2.4. Funcionamiento del Relé de Estado Sólido

El relé de estado sólido es un elemento que está constituido por un led, el cual indica el estado del mismo. Por el momento solamente se comprobará el funcionamiento de este componente mediante el led, ya que más adelante se verá el real funcionamiento del relé realizando las acciones de corte y reconexión.



Figura 4. 6: Funcionamiento del Relé de Estado Sólido

2.5. Funcionamiento del Contador de pulsos

Para verificar que la tarjeta está contando adecuadamente los pulsos, se utilizó un medidor híbrido de prueba, el mismo que está conectado a la red eléctrica y a una serie de focos de consumo. Este medidor cuenta con un marcador ciclométrico el mismo que funciona a base de un motor y que gira mediante una señal pulsante. Es desde esta misma señal donde la procesamos para realizar el conteo en nuestra tarjeta. Cada 2 pulsos generados por la señal representa 1/10 de kilovatio/hora por cuya razón cada 20 pulsos marcarán 1 kilovatio/hora en el LCD de la placa. Para comprobar su correcto funcionamiento se verificaron los cambios de la medida en la tarjeta.



Figura 4. 7: Señal generadora de pulsos



Figura 4. 8: Primera medición de pulsos

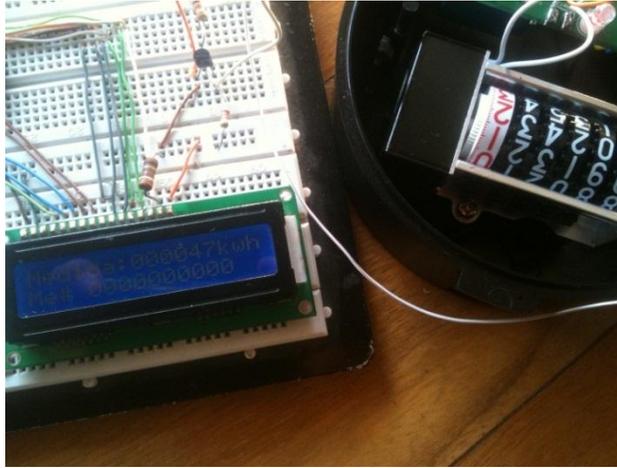


Figura 4. 9: Segunda medición de pulsos

2. Pruebas de hardware en la tarjeta

Para el circuito impreso final se hicieron exactamente las mismas pruebas que para el protoboard, para el cual se obtuvieron los mismos resultados y por ende constatando el correcto funcionamiento del de la tarjeta.



Figura 4. 10 (a): Prueba 1 Funcionamiento del contador de pulsos



Figura 4.9 (b): Prueba 2 Funcionamiento del contador de pulsos

Como podemos observar en la figura 4.9 (a) y 4.9 (b), la fuente de tensión está funcionando debidamente, ya que todo en el circuito se enciende correctamente. Además se puede destacar la diferencia entre la medida de la figura 4.9 (a) con respecto a la 4.9 (b) lo que indica que el contador de pulsos también se encuentra funcionando.

Después se prueba que el xbee este comunicándose correctamente con la computadora.



Figura 4. 11: Funcionamiento del Xbee en la placa

En la figura 4.10 se puede constatar que el xbee se encuentra funcionando ya que como se mencionó anteriormente en este capítulo el led rojo y verde indican una comunicación exitosa. Para finalizar con las pruebas de la tarjeta se conectó el relé de interrupción y se obtuvieron los mismos resultados que con el protoboard por lo que se demostró que todo el circuito funcionando correctamente.

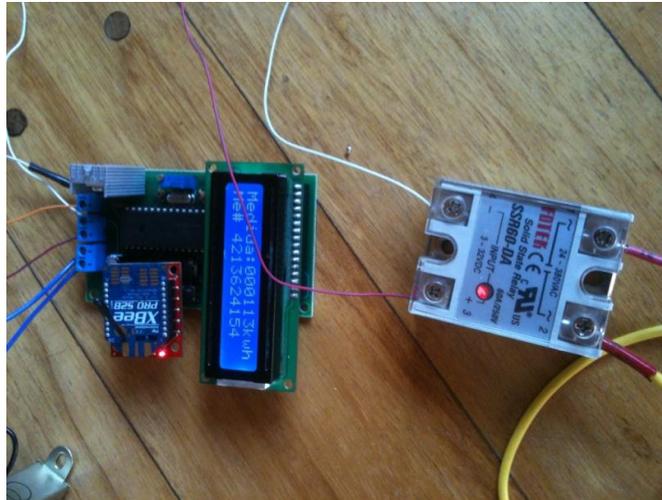


Figura 4. 12: Relé de estado sólido conectado a la placa

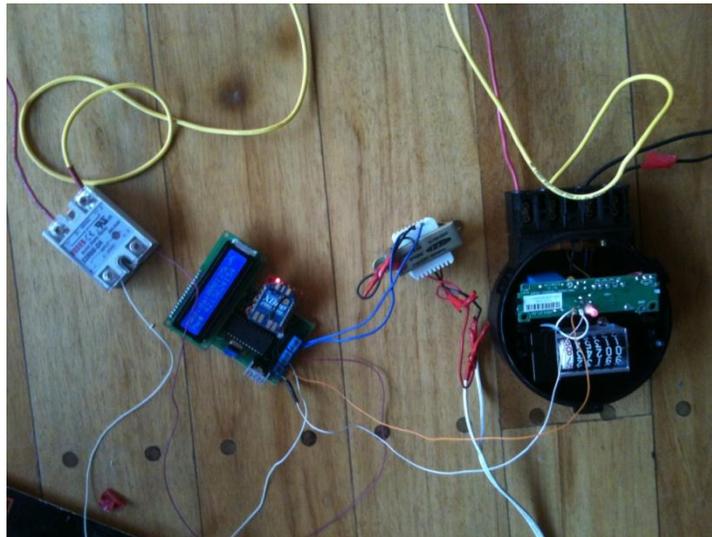


Figura 4. 13: Conexión Final de todo el circuito

3. Pruebas de Software

3.1. Pruebas de lectura

Se hizo uso de los bancos de trabajo de la EMPRESA ELECTRICA CENTRO SUR C.A., los mismos constan de equipos que nos permiten regular el voltaje y la corriente de funcionamiento que va hacia el medidor, a continuación se muestra la figura con todos los parámetros configurados para la prueba.



Figura 4. 14: Banco de trabajo de la EMPRESA ELECTRICA

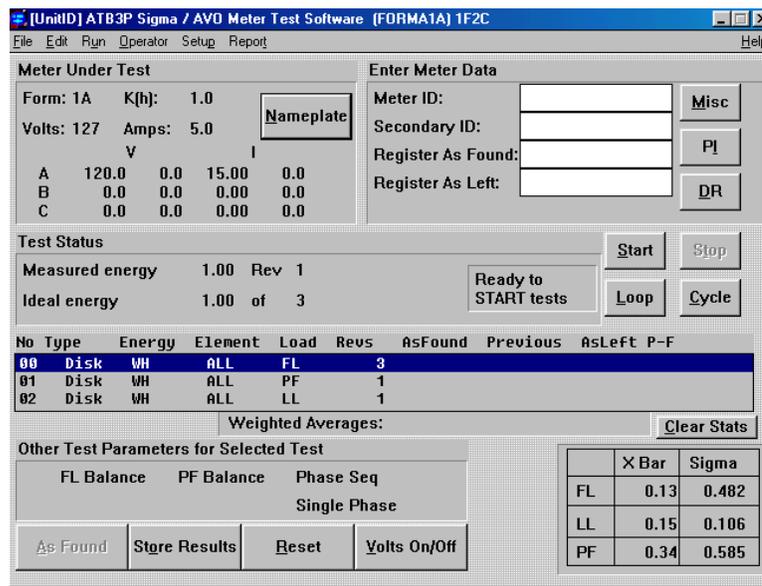


Figura 4. 15: Configuración de banco de trabajo para pruebas de la lectura

La figura 4.14 muestra una configuración del banco de trabajo, para una corriente de hasta 5 amperios a 127 voltios. La prueba consistió en dejar los medidores conectados a la placa y al banco de trabajo durante 16 horas, tomando lecturas cada 15 minutos a las tensiones y corrientes ya mencionadas. A continuación mostraremos las imágenes de las placas funcionando durante los primeros minutos sincronizadas con sus respectivos medidores.



Figura 4. 16: Medidor 1 en minutos iniciales



Figura 4. 17: Placa 1 en minutos iniciales



Figura 4. 18: Medidor 2 en minutos iniciales



Figura 4. 19: Placa 1 en minutos iniciales

Una vez sincronizada las placas se procedió a realizar la correspondiente prueba de 16 horas, la misma que consistió en adquirir las las lecturas que marcaba el medidor y almacenarlas en la base de datos durante el tiempo delimitado, para verificar la sincronía y la veracidad de los datos

- **Lecturas del medidor 1**

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión			
	Codigo Medidor	Medida	Fecha / Hora
Visualizar Clientes <input type="text" value="Por Código"/> Codigo Medidor <input type="text" value="2011121898"/>	2011121898	258,4	04/01/2013 16:10:00
	2011121898	258,5	04/01/2013 16:25:00
	2011121898	258,7	04/01/2013 16:40:00
	2011121898	258,8	04/01/2013 16:55:00
	2011121898	259	04/01/2013 17:10:01
	2011121898	259,1	04/01/2013 17:25:00
	2011121898	259,3	04/01/2013 17:40:00
	2011121898	259,4	04/01/2013 17:55:00
	2011121898	259,6	04/01/2013 18:10:01
	2011121898	259,7	04/01/2013 18:25:00
	2011121898	259,9	04/01/2013 18:40:00

Tiempo de inicio
 hora minutos segundos

Tiempo lectura

Figura 4. 20 (a): Registro de medidas de la tarjeta 1

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión			
	Codigo Medidor	Medida	Fecha / Hora
Visualizar Clientes <input type="text" value="Por Código"/> Codigo Medidor <input type="text" value="2011121898"/>	2011121898	260	04/01/2013 18:55:00
	2011121898	260,2	04/01/2013 19:10:00
	2011121898	260,3	04/01/2013 19:25:01
	2011121898	260,5	04/01/2013 19:40:00
	2011121898	260,6	04/01/2013 19:55:00
	2011121898	260,8	04/01/2013 20:10:00
	2011121898	260,9	04/01/2013 20:25:00
	2011121898	261,1	04/01/2013 20:40:01
	2011121898	261,2	04/01/2013 20:55:00
	2011121898	261,4	04/01/2013 21:10:01
	2011121898	261,5	04/01/2013 21:25:00

Tiempo de inicio
 hora minutos segundos

Tiempo lectura

Figura 4.19 (b): Registro de medidas de la tarjeta 1

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión

Visualizar Clientes
Por Código

Código Medidor
2011121898

Código Medidor	Medida	Fecha / Hora
2011121898	261,7	04/01/2013 21:40:00
2011121898	261,8	04/01/2013 21:55:00
2011121898	262	04/01/2013 22:10:00
2011121898	262,1	04/01/2013 22:25:01
2011121898	262,3	04/01/2013 22:40:00
2011121898	262,4	04/01/2013 22:55:00
2011121898	262,6	04/01/2013 23:10:00
2011121898	262,7	04/01/2013 23:25:00
2011121898	262,9	04/01/2013 23:40:01
2011121898	263	04/01/2013 23:55:00
2011121898	263,2	05/01/2013 0:10:00

Tiempo de inicio
hora minutos segundos
16 10 0

Tiempo lectura
15

Figura 4.19 (c): Registro de medidas de la tarjeta 1

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión

Visualizar Clientes
Por Código

Código Medidor
2011121898

Código Medidor	Medida	Fecha / Hora
2011121898	263,3	05/01/2013 0:25:00
2011121898	263,5	05/01/2013 0:40:00
2011121898	263,6	05/01/2013 0:55:00
2011121898	263,8	05/01/2013 1:10:00
2011121898	263,9	05/01/2013 1:25:00
2011121898	264,1	05/01/2013 1:40:01
2011121898	264,2	05/01/2013 1:55:00
2011121898	264,4	05/01/2013 2:10:00
2011121898	264,5	05/01/2013 2:25:00
2011121898	264,7	05/01/2013 2:40:01
2011121898	264,8	05/01/2013 2:55:00

Tiempo de inicio
hora minutos segundos
16 10 0

Tiempo lectura
15

Figura 4.19 (d): Registro de medidas de la tarjeta 1

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión			
	Codigo Medidor	Medida	Fecha / Hora
Visualizar Clientes Por Código Codigo Medidor 2011121898	2011121898	265	05/01/2013 3:10:00
	2011121898	265,1	05/01/2013 3:25:00
	2011121898	265,3	05/01/2013 3:40:00
	2011121898	265,4	05/01/2013 3:55:01
	2011121898	265,6	05/01/2013 4:10:00
	2011121898	265,7	05/01/2013 4:25:00
	2011121898	265,9	05/01/2013 4:40:00
	2011121898	266	05/01/2013 4:55:00
	2011121898	266,2	05/01/2013 5:10:00
	2011121898	266,3	05/01/2013 5:25:00
	2011121898	266,5	05/01/2013 5:40:01

Tiempo de inicio

hora: 16 minutos: 10 segundos: 0

Tiempo lectura

15

Figura 4.19 (e): Registro de medidas de la tarjeta 1

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión			
	Codigo Medidor	Medida	Fecha / Hora
Visualizar Clientes Por Código Codigo Medidor 2011121898	2011121898	266,7	05/01/2013 5:55:00
	2011121898	266,8	05/01/2013 6:10:00
	2011121898	267	05/01/2013 6:25:00
	2011121898	267,1	05/01/2013 6:40:00
	2011121898	267,3	05/01/2013 6:55:01
	2011121898	267,4	05/01/2013 7:10:00
	2011121898	267,6	05/01/2013 7:25:00
	2011121898	267,7	05/01/2013 7:40:00
	2011121898	267,9	05/01/2013 7:55:00
	2011121898	268,1	05/01/2013 8:10:00
	2011121898	268,2	05/01/2013 8:25:00

Tiempo de inicio

hora: 16 minutos: 10 segundos: 0

Tiempo lectura

15

Figura 4.19 (f): Registro de medidas de la tarjeta 1

- **Lecturas del medidor 2**

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión			
Código Medidor	Medida	Fecha/Hora	
2011128384	712,6	04/01/2013 16:10:02	Tiempo de inicio hora minutos segundos 16 10 0 Tiempo lectura 15
2011128384	712,7	04/01/2013 16:25:02	
2011128384	712,9	04/01/2013 16:40:02	
2011128384	713	04/01/2013 16:55:02	
2011128384	713,2	04/01/2013 17:10:03	
2011128384	713,3	04/01/2013 17:25:02	
2011128384	713,5	04/01/2013 17:40:02	
2011128384	713,6	04/01/2013 17:55:02	
2011128384	713,8	04/01/2013 18:10:03	
2011128384	713,9	04/01/2013 18:25:02	
2011128384	714,1	04/01/2013 18:40:03	

Figura 4. 21 (a): Registro de medidas de la tarjeta 2

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión			
Código Medidor	Medida	Fecha/Hora	
2011128384	714,2	04/01/2013 18:55:02	Tiempo de inicio hora minutos segundos 16 10 0 Tiempo lectura 15
2011128384	714,4	04/01/2013 19:10:02	
2011128384	714,5	04/01/2013 19:25:03	
2011128384	714,7	04/01/2013 19:40:02	
2011128384	714,8	04/01/2013 19:55:03	
2011128384	715	04/01/2013 20:10:02	
2011128384	715,1	04/01/2013 20:25:02	
2011128384	715,3	04/01/2013 20:40:03	
2011128384	715,4	04/01/2013 20:55:02	
2011128384	715,6	04/01/2013 21:10:03	
2011128384	715,7	04/01/2013 21:25:02	

Figura 4.20 (b): Registro de medidas de la tarjeta 2

Agregar Nuevo Cliente Leer Datos Pago Ingresar Código Medidor Corte y Reconexión			
Código Medidor	Medida	Fecha/Hora	
2011128384	715,9	04/01/2013 21:40:02	Tiempo de inicio hora minutos segundos 16 10 0 Tiempo lectura 15
2011128384	716	04/01/2013 21:55:02	
2011128384	716,2	04/01/2013 22:10:02	
2011128384	716,3	04/01/2013 22:25:03	
2011128384	716,5	04/01/2013 22:40:02	
2011128384	716,6	04/01/2013 22:55:02	
2011128384	716,8	04/01/2013 23:10:02	
2011128384	716,9	04/01/2013 23:25:02	
2011128384	717,1	04/01/2013 23:40:03	
2011128384	717,2	04/01/2013 23:55:02	
2011128384	717,4	05/01/2013 0:10:02	

Figura 4.20 (c): Registro de medidas de la tarjeta 2



Figura 4.20 (d): Registro de medidas de la tarjeta 2



Figura 4.20 (e): Registro de medidas de la tarjeta 2

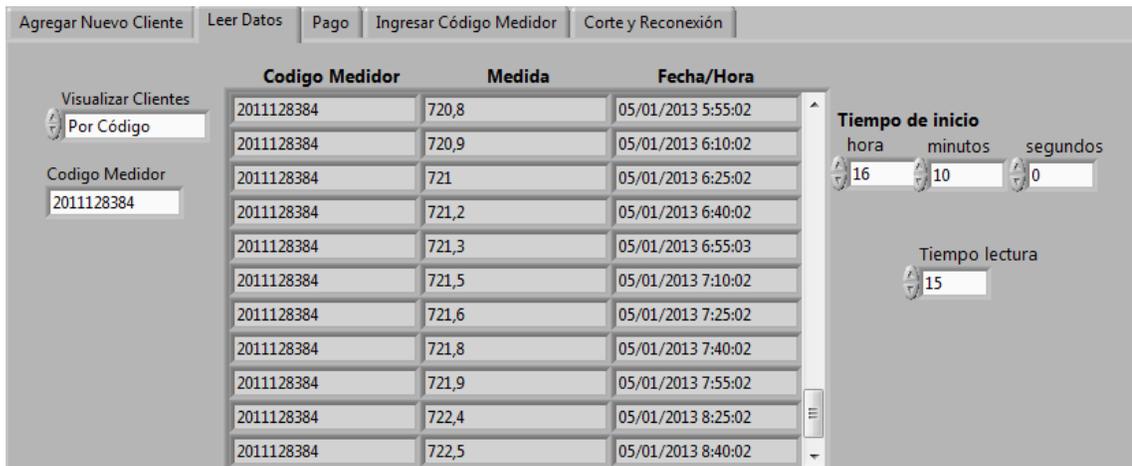


Figura 4.20 (f): Registro de medidas de la tarjeta 2

Como se observa en las figuras 4.19 y 4.20, se adquirieron señales durante las 16 horas sin interrupción, lo que verifica que la adquisición de datos está funcionando correctamente para las 2 placas. Para verificar la sincronía de la medida entre el medidor y placa electrónica, se tomaron fotos de las penúltimas medidas, constatando así que no existieron errores.



Figura 4. 22: Medidor 1, minutos finales



Figura 4. 23: Placa 1, minutos finales



Figura 4. 24: Medidor 2, minutos finales



Figura 4. 25: Placa 1, minutos finales

Como vemos en las figuras de arriba ninguna de las placas marca desincronización al haber pasado 16 horas funcionando, por lo tanto la placa y el almacenamiento de datos funcionan correctamente.

3.2. Pruebas de Corte y Reconexión

Para esta prueba hemos empezado por utilizar una carga de 4 focos de 110W, los mismos que estarán conectados al relé de corte y reconexión, permitiéndonos así conocer si la tarjeta está cortando o no el servicio. Para empezar hemos conectado la tarjeta con carga y hemos establecido la comunicación entre concentrador y tarjeta.

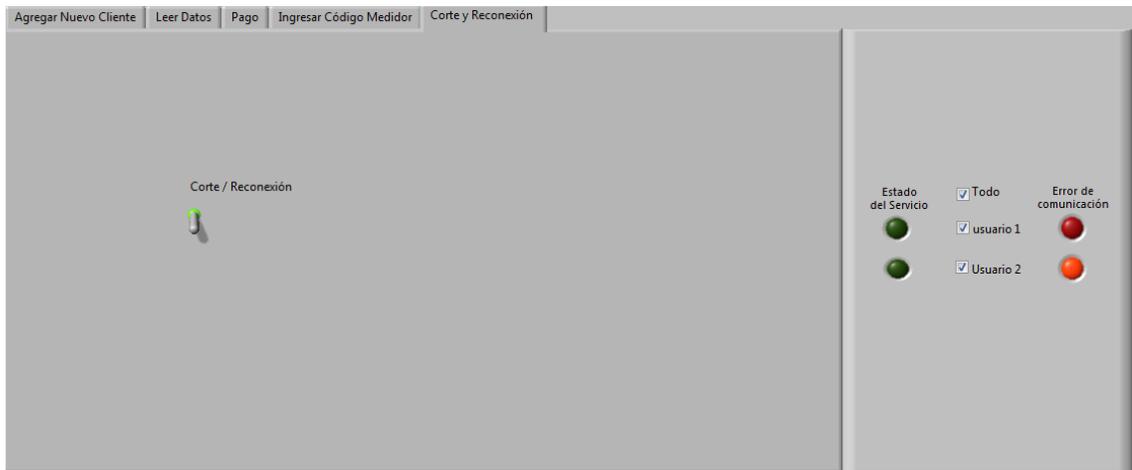


Figura 4. 26: Visualización del estado del servicio con la una tarjeta en estado de corte y la otra desconectada



Figura 4. 27: Relé en estado de corte para la tarjeta 1

Como se aprecia en la figura 4.25, el estado del servicio para el usuario 1 se encuentra cortado, ya que el led verde apagado del mismo indica un corte, por otro lado para el usuario 2 como se ve existe un error de comunicación ya que el led rojo se encuentra prendido, esto es debido a que la tarjeta de este usuario no se encuentra funcionando. Por otro lado si nos fijamos en la figura 4.26 tanto el relé de estado sólido como los focos se encuentran apagados, confirmando así que el software indica el estado correcto del servicio.

Ahora si presionamos el botón de corte y reconexión que se muestra en la figura 4.25, obtendremos los siguientes resultados.



Figura 4. 28: Visualización del estado del servicio con la una tarjeta en estado conectado y la otra desconectada

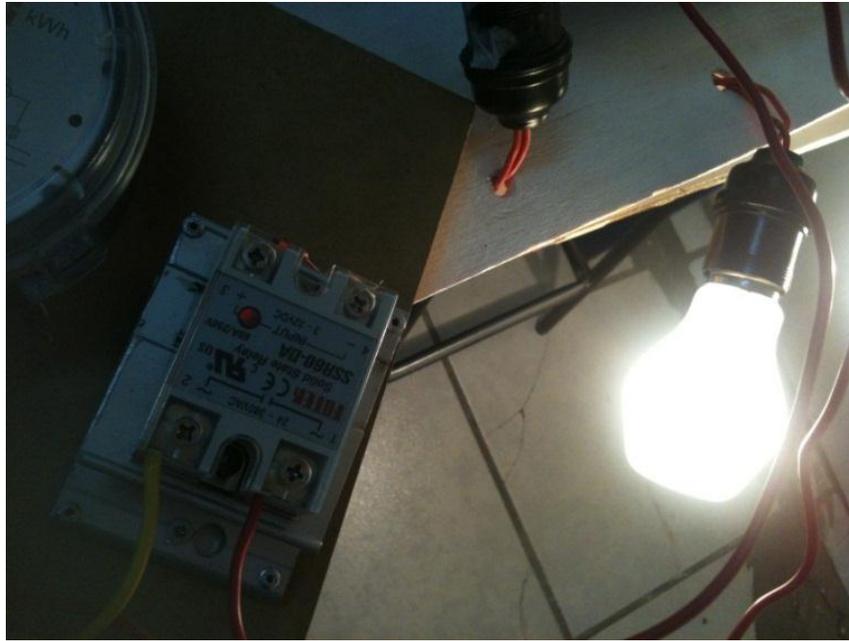


Figura 4. 29: Relé en estado de conectado para la tarjeta 1

Como se puede ver en la figura 4.27, el led verde que anteriormente estaba apagado, indicando un corte en el estado de la tarjeta 1 ahora se encuentra prendido lo que significa que se ha reconectado el servicio, por ahora el led rojo que indica error de conexión para el usuario 2 se mantiene prendido debido a que no se ha conectado la otra tarjeta todavía.

La siguiente prueba que realizamos fue con las 2 tarjetas conectadas y funcionando en estado de corte.



Figura 4. 30: Visualización del estado del servicio con las 2 tarjetas en estado de corte



Figura 4. 31: Relé en estado de corte para la tarjeta 1



Figura 4. 32: Relé en estado de corte para la tarjeta 2

La figura 4.29 nos indica que no existe ningún error en la comunicación, dado que se conectaron las 2 tarjetas correctamente, el led rojo que antes estaba prendido, se apagó, indicando una comunicación exitosa con las 2 placas. En esta misma figura se observa que los 2 leds verdes están apagados lo que nos dice que los 2 relés están en estado de corte para las 2 placas. Las figuras 4.30 y 4.31 confirman lo antes dicho y efectivamente los relés están en estado de corte haciendo exitosa a esta prueba.

El siguiente paso fue prender estos dos al mismo tiempo, es decir, restaurar el servicio eléctrico, para ello seleccionamos el botón de opción que dice “todo” y presionamos el botón de “corte/reconexión”, obteniendo como resultado la reconexión del servicio de los 2 relés.

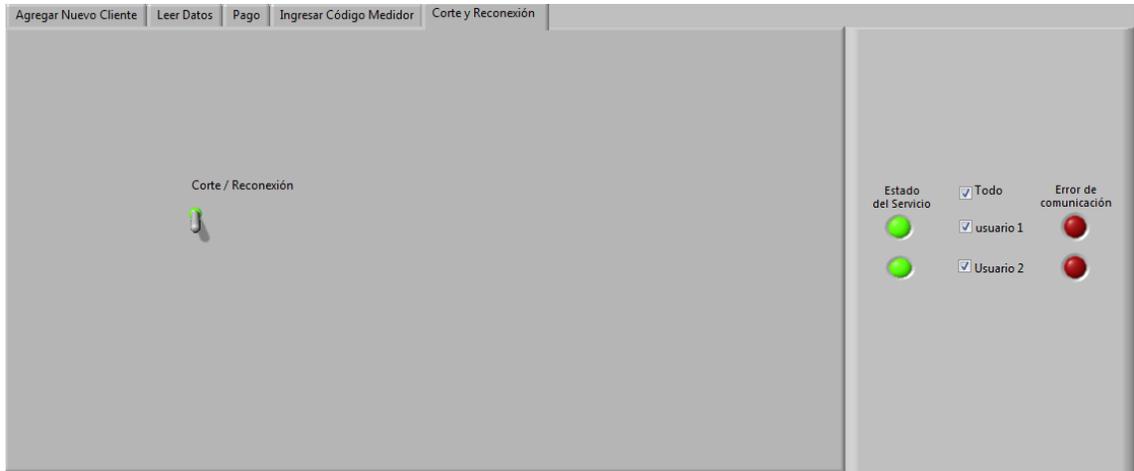


Figura 4. 33: Visualización del estado del servicio con las 2 tarjetas en estado de conectadas

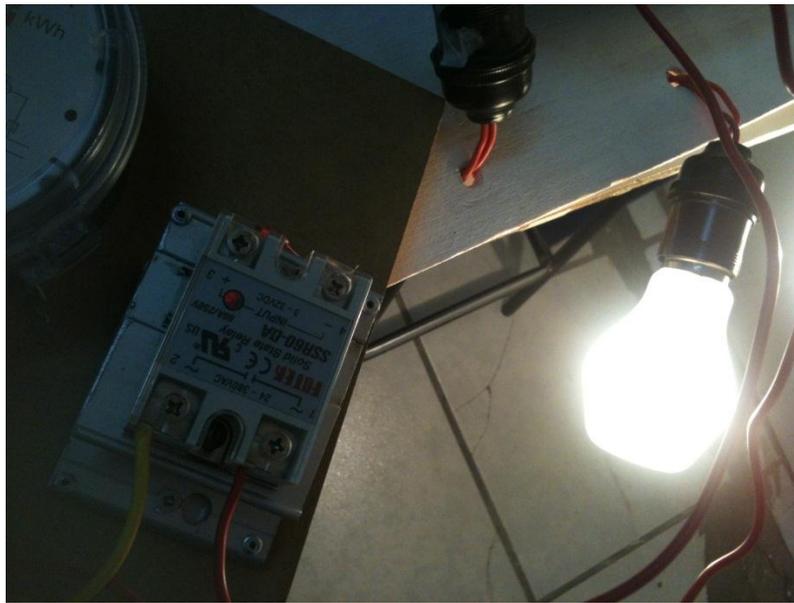


Figura 4. 34: Relé en estado de conectado para la tarjeta 1

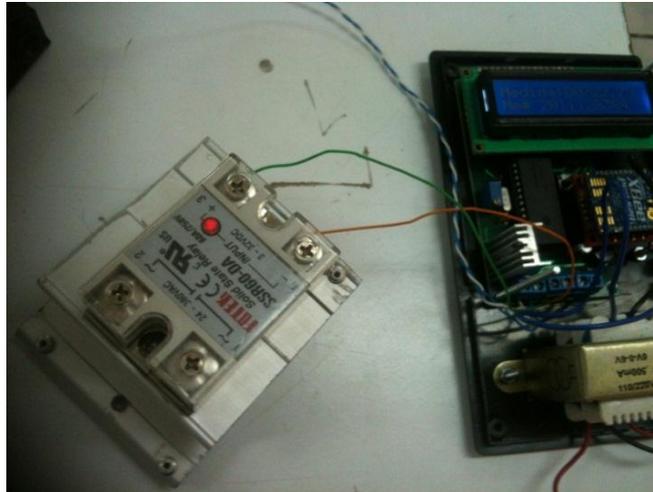


Figura 4. 35: Relé en estado de conectado para la tarjeta 2

Es muy importante acotar que después de muchas pruebas, la Base de Datos ha recogido 8016 medidas y toda esta información solamente utiliza 416KB del disco duro, por lo que se puede constatar que la información que se transmite desde los medidores no es extensa ni ocupa muchos recursos.

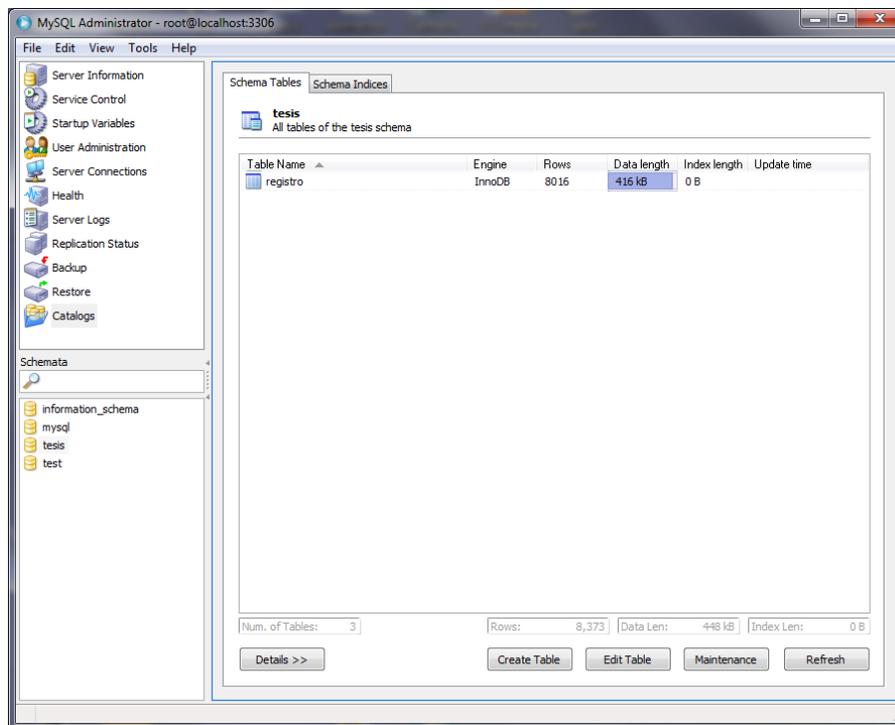


Figura 4. 36: Tamaño de la información en la Base de Datos

CONCLUSIONES

Investigando los sistemas de medición que existen, pudimos determinar que nuestro prototipo se asemeja a los sistemas de medición AMI, sin duda es una propuesta válida para el control y medición en la distribución eléctrica.

En el desarrollo del proyecto de investigación, diseñamos un prototipo de un sistema de telemetría y control, con el cual demostramos que existe la posibilidad de reutilizar los medidores monofásicos híbridos ya implementados en el sistema de medición de la EMPRESA ELECTRICA REGIONAL CENTROSUR.

El objetivo de reutilizar este recurso es de abaratar costos y una reducción de desechos por ende un impacto ambiental menor, puesto que, de implementar un sistema AMI se debe desechar los medidores ahora implementados.

Es una solución eficiente al problema de acceso físico a los medidores instalados en diferentes lugares de la ciudad, eliminando las dificultades, costos y molestias adicionales relacionadas con el ingreso a domicilios en donde se necesita autorización para poder realizar las lecturas, cortes o reconexiones.

El hardware fue diseñado de manera que cumpla todos los objetivos planteados, mientras que el software es muy sencillo de utilizarlo, haciéndolo amigable al usuario y el componente electrónico utilizado para el corte y reconexión del servicio, fue escogido correctamente puesto que funciona de manera adecuada con una carga dimensionada para los clientes residenciales. Es posible la integración del prototipo, puesto que, es muy sencillo de instalar, su tamaño es reducido, la transmisión de datos es eficaz y confiable y provee la información técnica necesaria para el control, la facturación confiable y oportuna, control remoto del servicio eléctrico, etc.

RECOMENDACIONES

Hacer varias pruebas del hardware diseñado antes de construirlo como circuito impreso, se pueden corregir errores que se presentan en todos los circuitos electrónicos. El momento de diseñar la tarjeta electrónica se debe tener en cuenta la posición de los componentes para utilizar el menos espacio posible.

Hacer un software lo más sencillo posible para que cualquier persona lo pueda entender y manejar, todo esto, comprobando el funcionamiento del prototipo individualmente (Hardware y Software), en conjunto y su comportamiento luego de transcurrir varias horas.

Continuar desarrollando el proyecto, hay mucho por hacer, por ejemplo, marcar alertas de las variaciones de voltaje, así se podrá mejorar el servicio para los clientes, se puede tomar el consumo total de cada transformador, para determinar si existe robo de energía, una manera muy sencilla de cortar el servicio de los abonados para hacer mantenimiento, de ser implementada con una pantalla LCD se podría mostrar datos de fechas de corte o el valor que debe ser cancelado en la planilla de cada mes.

BIBLIOGRAFÍA

Referencias Bibliográficas:

1. CORONEL Gutiérrez, Marco. (2011) “Estudio para la implementación del sistema AMI en la EMPRESA ELECTRICA REGIONAL CENTRO SUR C.A.”. Ingeniería Eléctrica. Universidad Politécnica Salesiana [Consulta: 8 de Noviembre de 2012]
2. OBANDO Morillo, Marco. PAZMIÑO Solís, Elizabeth. (2012) “Diseño e Implementación de un prototipo entrenador autómatas de ping pong controlado mediante comando de voz”. Ingeniería en Electrónica y Computación. Universidad Politécnica de Chimborazo. [Consulta: 4 de Diciembre de 2012]
3. CLAVIJO Mendoza, Juan Ricardo. (2011). Diseño y simulación de sistemas microcontrolados en lenguaje C. [En línea]: <http://www.mikroe.com/download/eng/documents/publications/other-books/libro_simulacion_mikroc.pdf>. [Consulta: 4 de Diciembre de 2012]
4. VALVERDE Rebazada, Jorge. (2007) El estándar inalámbrico zigbee. [En línea]: <<http://www.seccperu.org/files/ZigBee.pdf>>. [Consulta: 8 de Noviembre de 2012]
5. OBANDO Morillo, Marco. PAZMIÑO Solís, Elizabeth. (2012) “Diseño e Implementación de un prototipo entrenador autómatas de ping pong controlado mediante comando de voz”. Ingeniería en Electrónica y Computación. Universidad Politécnica de Chimborazo. [Consulta: 4 de Diciembre de 2012]
6. OYARCE Mino, Andrés. (2009). “Implementación del protocolo modbus sobre una tarjeta de desarrollo para su uso sobre una red”. Ingeniería Civil Electricista. Universidad de Chile. [Consulta: 3 de Enero de 2013]
7. PAZ, Santiago. (2008). “Diseño y construcción de un módulo transmisor receptor inalámbrico para el manejo de sensores con el protocolo estándar IEEE 802.15.4”. Ingeniería Electrónica. Universidad Tecnológica de la Mixteca. [Consulta: 3 de Enero de 2013]

8. MÉNDEZ Delgado, Gerardo. (2009). “Zigbee y sus aplicaciones en el monitoreo y control de procesos a distancia”. Ingeniería Electrónica. Universidad Autónoma de San Luis Potosí. [Consulta: 3 de Enero de 2013]
9. GONZALES, Carlos (2009). Conceptos de Telemetría, Telemando y Red GSM. [En línea]: <
<http://www.dspace.ups.edu.ec/bitstream/123456789/41/8/Capitulo2.pdf>>. [Consulta: 4 de Enero de 2013]
10. RAMOS Silvestre, Edgar. FUENTES Telleria, Ignacio. (2012) Implementación de un sistema de telemetría utilizando la tecnología inalámbrica WIFI. [En línea]: <
<http://www.slideshare.net/riftbol/implementacin-de-un-sistema-de-telemetry-utilizando-la-tecnologa-inalmbrica-wifi-14088259>>. [Consulta: 4 de Enero de 2013]

Referencias Electrónicas:

1. GUDIÑO, Jorge Luis. Sistemas de lectura remota. [En línea]: <
<http://neutron.ing.ucv.ve/revista-e/No7/Jorge%20Gudi%C3%B1o%5CSistemas%20de%20Lectura%20Remota.htm>>. [Consulta: 9 de Octubre de 2012]
2. Sistemas AMR. [En línea]: <
http://www.romatech.cl/Romatech/AMR_files/AMR%20Expo%202.pdf>. [Consulta: 9 de Octubre de 2012]
3. VIDRIO López, Gilberto. Infraestructura de medición avanzada ami en las redes inteligentes. [En línea]: <
<http://www.slideshare.net/FiiDEM/infraestructura-de-medicin-avanzada-ami-en-las-redes-inteligentes>>. [Consulta: 11 de Octubre de 2012]
4. ELECTRIC POWER RESEARCH INSTITUTE. Advanced metering infrastructure. 2007. [En línea]: <http://www.ferc.gov/eventcalendar/Files/20070423091846-EPRI%20-%20Advanced%20Metering.pdf>>. [Consulta: 11 de Octubre de 2012]

5. NATIONAL ENERGY TECHNOLOGY LABORATORY FOR THE U.S. DEPARTMENT OF ENERGY. Advanced metering infrastructure. 2008. [En línea]: < [http://www.netl.doe.gov/SmartGrid/referenceshelf/whitepapers/AMI%20White%20paper%20final%20021108%20\(2\)%20APPROVED_2008_02_12.pdf](http://www.netl.doe.gov/SmartGrid/referenceshelf/whitepapers/AMI%20White%20paper%20final%20021108%20(2)%20APPROVED_2008_02_12.pdf)>. [Consulta: 11 de Octubre de 2012]
6. Redes Smart Grid: eficiencia energética total. 2010. [En línea]: < <http://www.datati.es/2485/redes-smart-grid-eficiencia-energetica-total.html>>. [Consulta: 13 de Octubre de 2012]
7. Relé de estado sólido con triac. [En línea]: <<http://www.forosdeelectronica.com/proyectos/rele-estado-solido.htm>>. [Consulta: 5 de Noviembre de 2012]
8. SABACA, Mariano. Relé estado sólido. 2006. [En línea]: <http://www.soloelectronica.net/rele_solido.htm>. [Consulta: 5 de Noviembre de 2012]
9. Microcontrolador. 2007. [En línea]: < <http://es.wikipedia.org/wiki/Microcontrolador>>. [Consulta: 10 de Noviembre de 2012]
10. Microcontrolador. [En línea]: < <http://microcontroladores-e.galeon.com/>> [Consulta: 10 de Noviembre de 2012]
11. DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD DE VALENCIA. Microcontrolador pic18f4550. [En línea]: < <http://es.scribd.com/doc/57978444/PIC18F4550>>. [Consulta: 5 de Noviembre de 2012]
12. Configuración y conceptos básicos xbee s2b. 2012. [En línea]: < <http://plataformaszigbee.blogspot.com/2012/05/practica-1-configuracion-y-conceptos.html>>. [Consulta: 6 de Noviembre de 2012]
13. CORRIENTE CONTINUA. 2010. [En línea]: < <http://introduccionelectronica.blogspot.com/2010/06/corriente-continua.html>>. [Consulta: 6 de Noviembre de 2012]

14. GARCÍA SANTAMARÍA, GILBERTO. Fuente de corriente directa. [En línea]: http://www.ecured.cu/index.php/Fuente_de_corriente_directa. [Consulta: 6 de Noviembre de 2012]
15. Filtro por condensador. [En línea]: < http://educativa.catedu.es/44700165/aula/archivos/repositorio//3000/3079/html/44_filtro_por_condensador.html>. [Consulta: 6 de Noviembre de 2012]
16. PÉREZ VALDEZ, DAMIÁN. Que son las bases de datos. 2007. [En línea]: <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>>. [Consulta: 2 de Diciembre de 2012]
17. BASES DE DATOS. [En línea]: http://es.wikipedia.org/wiki/Base_de_datos>. [Consulta: 2 de Diciembre de 2012]
18. BASES DE DATOS. 2002. [En línea]: http://es.wikipedia.org/wiki/Base_de_datos>. [Consulta: 2 de Diciembre de 2012]
19. MYSQL.2010. [En línea]: <http://es.wikipedia.org/wiki/MySQL>>. [Consulta: 2 de Diciembre de 2012]
20. NATIONAL INSTRUMENTS CORPORATION. Modules and toolkits. 2012. [En línea]: http://zone.ni.com/reference/en-XX/help/370051V-01/cvi/cviaddon_toolkits/>. [Consulta: 3 de Diciembre de 2012]
21. NATIONAL INSTRUMENTS CORPORATION. Database connectivity toolset user manual. 2001. [En línea]: <http://www.ni.com/pdf/manuals/321525c.pdf>>. [Consulta: 3 de Diciembre de 2012]

GLOSARIO

MySQL.- Es un sistema de software libre que ayuda en la gestión de base de datos relacional, multiusuario y multihilo, es decir, que realiza varias tareas a la vez.

Comandos AT.- Son el lenguaje de programación para la configuración de los Xbee.

X-CTU.- Es un programa que nos permite programar de manera fácil y rápida cualquier tipo de modulo inalámbrico de la familia de los Xbee

ODBC.- Es la herramienta que nos permite ingresar a una base de datos desde cualquier aplicación.

MikroC.- Es un software con una amplia variedad de herramientas y ayudas que facilitan la creación de proyectos está basado en el lenguaje de programación C.

USART.- Es un acrónimo de Trasmisor y Receptor Síncrono/Asíncrono Universal. Se trata de un puerto que utiliza el formato serial para transmitir datos.

AMR.- Automatic Meter Reading, se refiere a la tecnología de lectura remota de medidores de agua luz y teléfono.

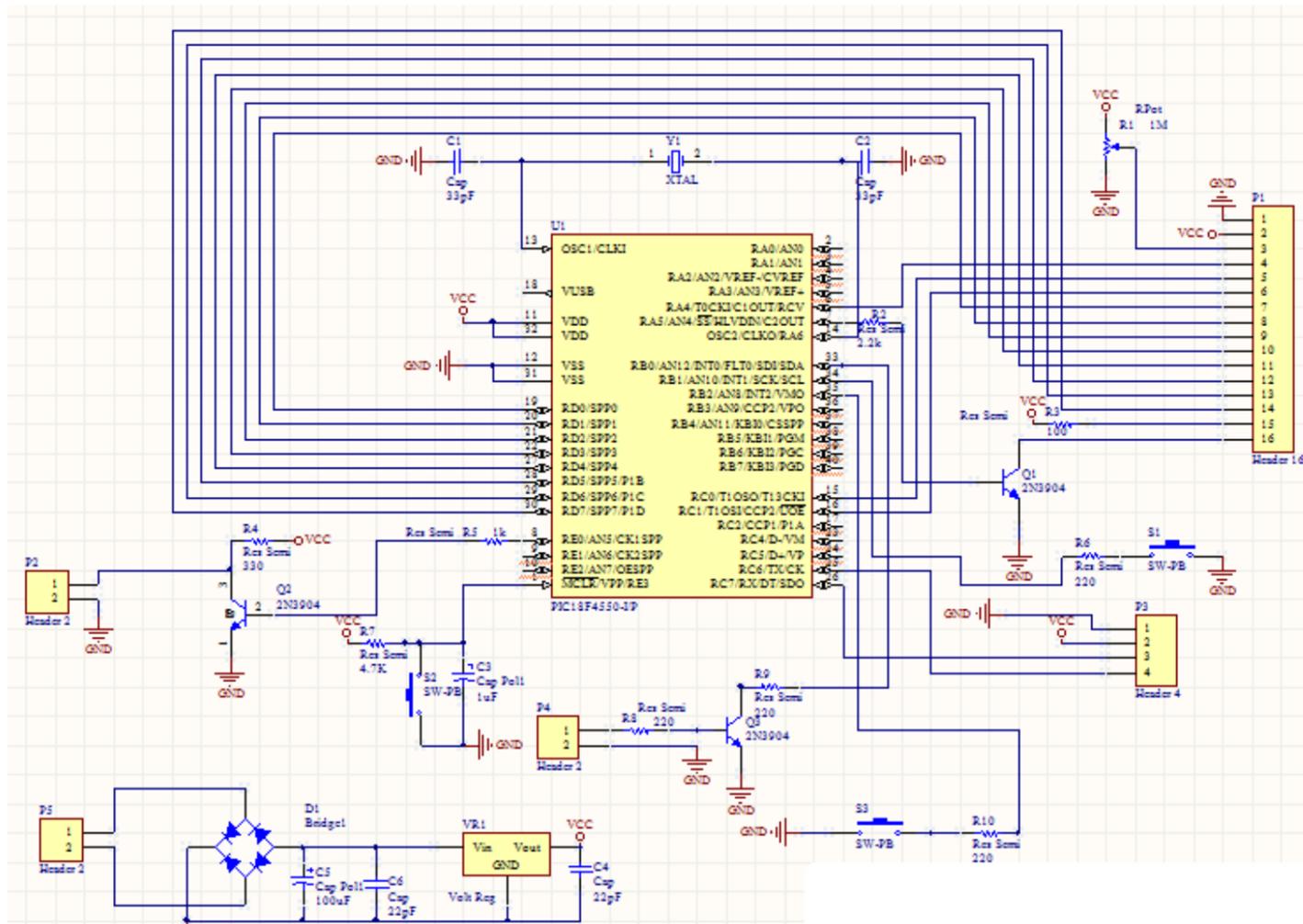
EEPROM.- Se refiere a la memoria interna de un microcontrolador que se caracteriza por poder guardar y borrar información dentro de ella.

Xbee.- es un módulo que sirve para comunicaciones inalámbricas que utiliza la frecuencia libre de los 2.4 GHz.

ANEXO

En los anexos incluimos el diseño del circuito electrónico y el programa completo desarrollado en LabVIEW, puesto que, son diagramas de gran tamaño y necesitamos un espacio más grande para su mejor visualización y entendimiento. Se anexa hojas de tamaño A3 con lo que se visualizara mejor las imágenes.

Anexo 1 Diseño del circuito final



Anexo 2: Diagrama de bloques de Toma de lecturas y control del servicio

