

Universidad Tecnológica de la Mixteca

Codificador de Pulsos KYZ bajo el Protocolo de Comunicaciones MODBUS para Medidores Electrónicos de Energía Eléctrica

TESIS

para obtener el título de:
Ingeniero en Electrónica

Presenta:
Enmanuel Aparicio Velázquez

Director:
M. C. Fermín Hugo Ramírez Leyva

Huajuapán de León, Oaxaca. Octubre de 2004.

Tesis presentada el 6 de noviembre de 2004 ante los siguientes sinodales:

M.C. José Antonio Moreno Espinosa
M.C. Esteban Oswaldo Guerrero Ramírez
C. DR. Felipe Santiago Espinosa

Bajo la dirección de:
M.C. Fermín Hugo Ramírez Leyva

Dedicatoria

Dedico este trabajo a las personas que siempre han estado conmigo a pesar de la distancia que nos separa físicamente. Con todo mi amor:

A MIS PADRES

Fresvinda Velázquez Herrera y Josafat Aparicio Carrasco

A MIS HERMANOS

Nancy, Alexander, Gladys y Yuridia Aparicio Velázquez

La familia que tanto amo.

Enmanuel.

Agradecimiento especial

A mi profesor y gran amigo Fermín Hugo Ramírez Leyva, sin ningún título que lo distinga más que su excelente calidad humana. Mil gracias por todo su apoyo, sus consejos, pero sobre todo por su valiosa amistad.

Enmanuel.

Agradecimientos

El día de hoy tengo mucho que agradecer a tantas personas que no quisiera omitir a nadie. Por eso, de antemano agradezco a todas las que han contribuido a la construcción de mi persona como hijo, hermano, amigo, estudiante, alumno, compañero y, por consecuencia, como profesionalista, esposo y padre.

Agradezco a DIOS por acompañarme siempre en todo lo que hago y permitirme llegar a este dichoso momento.

A mi madre por darme la vida, el amor y el aliento que necesito para seguir adelante.

A mi padre porque me ha enseñado con el ejemplo que sólo con trabajo y entrega se consiguen nuestros anhelos. Gracias papá por sus consejos y porque nunca ha dejado de apoyarme.

A Nancy, Alexander, Gladys y Yuridia porque juntos crecimos, peleamos, sufrimos, lloramos, pero sobre todo reímos y compartimos grandes momentos de felicidad. Los amo hermanos. Muchas gracias por todo.

A mis abuelitas Eloísa y Josafat porque son el vivo ejemplo de sacrificio, lucha y trabajo constantes para que los hijos crezcan sin carencias. Las adoro abuelitas lindas.

A todos mis tíos y tías, sin excepción alguna. De manera especial a mis “tíos-hermanos” Antonio, Julio, Amando y Genoveva Aparicio Carrasco, y a Elena, Irene, Josefa, Eufrosina y Juan Velázquez Herrera. Muchas gracias por todo su apoyo.

A todos mis profesores, gracias por contribuir a mi preparación profesional. De manera especial a quienes considero mis amigos: Fermín Hugo Ramírez Leyva, José Antonio Moreno Espinosa, Esteban Oswaldo Guerrero Ramírez y Felipe Santiago Espinosa, no sólo por el tiempo dedicado a la revisión de este documento y sus valiosas sugerencias para que éste fuera un buen trabajo, sino también por la gran amistad y los consejos que me brindaron durante mi carrera; a Jacob Javier Vásquez Sanjuán porque fue el primero que me dio motivación al iniciar mi carrera y juntos convivimos en un proyecto al final de la misma; a Heriberto I. Hernández Martínez por sus palabras llenas de motivación y el apoyo incondicional que me ofreció para que este trabajo llegara a buen término; a Gabriel Gerónimo Castillo por su alentadora amistad. A todos ellos, sin ningún otro título más que el de mis amigos, muchas gracias.

Al M.C. Mario Moreno Rocha y al DR. Christian Sturm les agradezco su valiosa colaboración y el tiempo dedicado a la revisión y mejoramiento de la interfaz de usuario de este trabajo. Asimismo agradezco a Christian su amistad, consejos y apoyo incondicional que me ha brindado.

A doña Palmira y a todos los integrantes de las familias Palafox Alvarado, Salazar Montesinos, Oledo Ramos, Sibaja Martínez, Sánchez Rojas, Rueda López y a doña Elia y Marisol porque todos ellos me hicieron sentir parte de su familia y a su lado nunca me sentí solo.

A la generación 1998-2003 de la Ingeniería en Electrónica, de la cual me siento muy orgulloso de ser parte porque juntos emprendimos nuevos retos y pasamos grandes momentos.

A las bellas chicas del Racimo: Adriana, Fabiola, Yazmín, Berenice, Nube y Mónica gracias por ser unas amigas tan lindas.

A mis grandes amigos del Rebaño Sagrado (bautizados así por el Racimo): Seco, Fox, Memo y Julio. Gracias por su enorme amistad, su carnal "AVE" los quiere y siempre podrán contar con él. Ahora van ustedes.

A mis hermanas adoptivas Lucy, Alejandra, Sahadi y Rebeca muchas gracias por su cariño, amistad y apoyo incondicional.

A Keina por representar el primer bello sentimiento que nunca se olvida, a Quetzaly, Adriana y Rose Mary gracias por su cariño, siempre las recordaré.

A Cinthya Arelly porque es mi ángel de la guarda y aún sigue viviendo en mi corazón y en mi pensamiento. Gracias por acompañarme y cuidarme en todo momento.

A ti Yutzin, porque has venido a darle luz a mi vida, sentido a mi existencia y por ser la más grande razón que tengo para seguir adelante y construir un buen futuro para los que vienen y nos acompañarán.

TE AMO.

Ya ti también Tlatoani, porque te espero con todas las ansias de mi corazón y me motivas a crecer como persona para darte el mejor ejemplo del mundo.

A todos ustedes: mil gracias.

Los quiere.

Enmanuel.

Índice general

Índice general	xi
Índice de figuras	xv
Índice de tablas	xix
Resumen	xxi
Abstract	xxiii
Introducción	xxv
Objetivo general	xxviii
Objetivos específicos	xxviii
Contenido del documento de tesis	xxix
1. Costo y medición de la energía eléctrica	1
1.1. Variables eléctricas de una instalación trifásica	1
1.2. Facturación de la energía eléctrica en México	5
1.2.1. Reducción de costos por consumo de energía eléctrica	6
1.3. Demanda máxima	6
1.3.1. Controladores de demanda máxima	8
1.4. Medidores de energía eléctrica	9
1.4.1. Medidor electrónico con salidas a relevador KYZ	10
1.4.1.1. Medición de los pulsos KYZ	11
1.4.1.2. Constante de energía Ke	11
1.4.1.3. Uso de los pulsos KYZ y la constante de energía Ke	11
2. El protocolo de comunicaciones MODBUS	15
2.1. Generalidades	15
2.2. Protocolo MODBUS de Línea Serial	16
2.2.1. Descripción del protocolo	16
2.2.1.1. Modo dirigido	17
2.2.1.2. Modo de difusión	18
2.2.2. Estructura de la trama MODBUS RTU	18
2.2.2.1. Delimitador de tramas	20
2.2.2.2. Campo de dirección	20
2.2.2.3. Campo de función	20
2.2.2.4. Campo de datos	21
2.2.2.5. Campo de comprobación de error	21
2.2.2.6. Trama de las respuestas de excepción	21
2.2.3. Modo de transmisión RTU	22
2.2.3.1. Generación del campo CRC en modo RTU	22

2.2.4. Códigos de función MODBUS	23
2.2.4.1. Función 0x03 (Lectura de registros mantenidos)	24
2.2.4.2. Función 0x04 (Lectura de registros de entrada)	25
2.2.4.3. Función 0x06 (Escribir un registro mantenido)	25
2.2.4.4. Función 0x10 (Escribir múltiples registros mantenidos)	26
2.2.4.5. Función 0x17 (Leer y escribir múltiples registros)	26
2.2.4.6. Función 0x08 (Diagnóstico del dispositivo)	27
2.2.4.6.1. Subfunción 0x0000 (Solicitud de eco)	27
2.2.4.6.2. Subfunción 0x0001 (Reinicio de comunicación)	28
2.2.4.6.3. Subfunción 0x0002 (Devolver el registro de diagnóstico)	28
2.2.4.6.4. Subfunción 0x0004 (Activar el MSE)	28
2.2.4.6.5. Subfunción 0x000A (Reinicio de contadores de eventos de comunicación)	28
2.2.4.6.6. Subfunciones 0x000B a 0x000F (Devolver contador de evento)	28
2.2.4.7. Función 0x0B (Devolver el contador de eventos de comunicación)	29
2.2.4.8. Función 0x2B-0x0D (Leer datos de identificación)	29
2.2.5. Códigos de excepción MODBUS	30
2.3. MODBUS en la capa física	30
2.3.1. MODBUS RS-485	31
2.3.1.1. Configuración MODBUS 2W	32
2.3.1.2. Configuración MODBUS 4W	32
2.3.1.3. Requerimientos del sistema multipunto	33
2.3.2. MODBUS RS-232	34
3. Diseño y desarrollo de un codificador de pulsos KYZ	37
3.1. Descripción general del sistema	37
3.2. Módulo de alimentación	39
3.3. Módulo de acoplamiento	39
3.4. Módulo de comunicaciones	40
3.4.1. Puertos de comunicación alambrada	40
3.4.1.1. Puerto RS-232	41
3.4.1.2. Puerto RS-485	42
3.4.2. Radio-módem MaxStream de 900 MHz	43
3.5. Módulo digital	45
3.5.1. Detección de los pulsos KYZ	46
3.5.2. Cuantificación de los pulsos KYZ	47
3.5.3. Codificación de los pulsos KYZ	48
3.5.4. Implementación de MODBUS en el AT90S2313	49
3.5.4.1. Uso de la memoria EEPROM	49
3.5.4.2. Uso de la memoria RAM	50
3.5.4.2.1. Memoria para recepción de tramas MODBUS	51
3.5.4.2.2. Memoria para registros mantenidos y de entrada	52
3.5.4.2.3. Memoria para registros contadores de eventos de comunicación	53
3.5.4.2.4. Memoria para transmisión de tramas MODBUS	53
3.5.4.3. Atención a un mensaje MODBUS	53
3.5.4.3.1. Validación de los campos de función y subfunción	54
3.5.4.3.2. Rango de direcciones válidas	55
3.5.4.3.3. Validación de los valores de registros	56
3.5.4.3.4. Códigos de excepción	57
3.5.4.3.5. Ejecución de la función	57
3.5.5. Descripción del firmware del CODKYZ	57
4. Desarrollo del software de pruebas y configuración del CODKYZ	61
4.1. Lenguaje de programación	61
4.2. Modelo de desarrollo	62
4.3. Requerimientos del Soft-KYZ	62

4.3.1. Requerimiento general	63
4.3.1.1. Requerimientos específicos:	63
4.4. Análisis de los requerimientos	64
4.4.1. Análisis modular del Soft-KYZ	64
4.4.2. Bloques del Soft-KYZ	64
4.4.2.1. Módulos del bloque de presentación	64
4.4.2.2. Módulos del bloque de procesamiento	65
4.4.2.3. Módulos del bloque de reporte	65
4.5. Diseño del Soft-KYZ	65
4.5.1. Diseño de los datos	65
4.5.2. Diseño del archivo de reporte	66
4.5.3. Diseño de la arquitectura	66
4.5.4. Diseño de la interfaz de usuario	66
4.5.5. Diseño de los componentes	67
4.6. Codificación del Soft-KYZ	67
4.6.1. Descripción de la tarea Monitoreo de los pulsos KYZ	68
4.6.1.1. Módulo Crea Archivo	68
4.6.1.2. Módulo Monitoreo	70
4.6.1.3. Módulo Tareas MODBUS	71
4.6.1.4. Módulo Escribe Respuesta	72
4.6.1.5. Módulo Calcula Variables	72
4.6.1.6. Módulo Escribe Archivo	73
4.6.2. Descripción de la tarea Registro de las transacciones MODBUS	73
4.6.2.1. Módulo Transacciones	74
4.6.2.2. Módulo Incrementa Datos	75
4.6.2.3. Módulo Solicita Contadores	75
4.6.3. Descripción de la tarea Configuración de los dispositivos	76
4.6.3.1. Módulo Configuración	76
4.6.3.2. Módulo Detecta CODKYZ	78
4.6.3.3. Módulo Configura Puerto	78
4.6.4. Descripción de la tarea Demostración del protocolo MODBUS	78
4.6.4.1. Módulo MODBUS	80
4.6.5. Estructura jerárquica de control del Soft-KYZ	80
4.7. Pruebas del Soft-KYZ	81
4.7.1. Pruebas de ejecución e integración de módulos	81
4.7.2. Pruebas de funcionamiento	82
4.7.3. Pruebas de usabilidad	83
5. Pruebas y resultados	85
5.1. Procedimiento de pruebas	86
5.2. Pruebas de laboratorio	86
5.2.1. Instalación del sistema para las pruebas de laboratorio	86
4.1.1.1. Elementos de las pruebas de laboratorio	88
5.2.1.1.1. Emulador de relevadores KYZ	89
5.2.2. Resultados de las pruebas de laboratorio	90
5.2.2.1. Resultados empleando el puerto serial RS-232	90
5.2.2.1.1. Detección y configuración del CODKYZ	90
5.2.2.1.2. Comprobación del registro de los Pulsos KYZ	90
5.2.2.1.3. Monitoreo de pulsos KYZ	92
5.2.2.1.4. Datos de las transacciones MODBUS hechas con el puerto RS-232	94
5.2.2.2. Resultados empleando el puerto serial RS-485	95
5.2.2.2.1. Datos de las transacciones MODBUS hechas con el puerto RS-485	95
5.2.3. Problemas presentados en las pruebas de laboratorio	96
5.3. Pruebas de campo	97

5.3.1. Instalación del sistema para pruebas de campo	97
5.3.1.1. Elementos de las pruebas de campo	97
5.3.1.2. Instalación de campo	98
5.3.2. Resultados de las pruebas de campo	100
5.3.2.1. <i>Monitoreo</i> de pulsos KYZ	101
5.3.2.2. Datos de las transacciones MODBUS hechas con los radio-módems	105
Conclusiones	107
Trabajos futuros	109
Referencias	111
Internet	113
Apéndice A. El microcontrolador AT90S2313	A-1
A.1. Características funcionales	A-1
A.2. Características programables	A-1
Apéndice B. Manual de usuario del CODKYZ	B-1
Introducción	B-1
B.1. Descripción general del CODKYZ	B-2
B.1.1. Características físicas	B-2
B.1.2. Características eléctricas	B-2
B.1.3. Identificación de componentes	B-2
B.2. Instalación y conexión	B-3
B.2.1. Dónde instalar el CODKYZ	B-3
B.2.2. Conexión de los pulsos KYZ	B-4
B.2.3. Conexión de los puertos de comunicaciones	B-5
B.2.3.1. Uso y conexión del puerto RS-232	B-5
B.2.3.2. Uso y conexión del puerto RS-485	B-6
B.2.3.2.1. Conexión al bus de 4 hilos	B-6
B.2.3.2.2. Conexión al bus de 2 hilos	B-6
B.2.3.3. Conexión del radio-módem 9XStream	B-8
B.2.3.4. Conexión e instalación de la antena	B-8
B.3. Soporte del protocolo MODBUS	B-9
B.3.1. Parámetros de comunicación MODBUS	B-9
B.3.2. Funciones MODBUS soportadas	B-9
B.3.3. Subfunciones de diagnóstico MODBUS	B-9
B.3.4. Códigos de excepción MODBUS	B-10
B.3.5. Registros mantenidos	B-10
B.3.6. Valores por omisión de los registros mantenidos	B-10
B.3.7. Formato comprimido de fecha y hora	B-10
B.3.8. Registros de entrada	B-11
B.3.9. Descripción de los LEDES indicadores	B-11
B.4. Diagrama esquemático	B-12
B.5. Lista de componentes	B-13
B.6. Diagrama de circuito impreso	B-13
B.7. Solución de problemas comunes	B-14

Índice de figuras

<i>Figura i.1.</i> Red MODBUS RS-485: <i>monitoreo</i> , automatización y control de procesos industriales. xvii	
<i>Figura 1.1.</i> Diagrama unifilar de una red eléctrica trifásica. _____	4
<i>Figura 1.2.</i> a) Banco de capacitores instalados para corregir el factor de potencia, b) conexión típica. _____	6
<i>Figura 1.3.</i> Consumo de energía de las fábricas A y B [11]. _____	7
<i>Figura 1.4.</i> Perfil de la curva de demanda eléctrica de una planta en un día [URL6]. _____	8
<i>Figura 1.5.</i> Clasificación de los medidores de energía eléctrica. _____	9
<i>Figura 1.6.</i> Funcionamiento de un relevador tipo A [URL7]. _____	10
<i>Figura 1.7.</i> Funcionamiento de un relevador KYZ [URL7]. _____	10
<i>Figura 1.8.</i> Relación entre la constante de energía y los pulsos KYZ de los medidores electrónicos. _____	13
<i>Figura 2.1.</i> Representación del modo dirigido de los mensajes MODBUS. _____	18
<i>Figura 2.2.</i> Representación del modo de difusión de los mensajes MODBUS. _____	18
<i>Figura 2.3.</i> Estructura general de un mensaje MODBUS. _____	19
<i>Figura 2.4.</i> Representación gráfica de un ciclo solicitud-respuesta (transacción) MODBUS. _____	19
<i>Figura 2.5.</i> Delimitadores de tiempo entre las tramas de mensajes MODBUS. _____	20
<i>Figura 2.6.</i> Estructura de la trama para respuestas de excepción. _____	21
<i>Figura 2.7.</i> Distribución de los bits por cada dato transmitido utilizando el modo RTU. _____	22
<i>Figura 2.8.</i> Secuencia de los bits del número 0x6A en la señal de transmisión RTU. _____	22
<i>Figura 2.9.</i> Categorías de función MODBUS. _____	23
<i>Figura 2.10.</i> Estructura de la interconexión de dispositivos utilizando la configuración 2W. _____	32
<i>Figura 2.11.</i> Estructura de la interconexión de dispositivos utilizando la configuración 4W. _____	33
<i>Figura 2.12.</i> Configuración punto a punto MODBUS RS-232. _____	34
<i>Figura 3.1.</i> Diagrama a bloques del hardware del CODKYZ. _____	37
<i>Figura 3.2.</i> Señales de entrada y salida de los bloques que integran el hardware del CODKYZ. _____	38
<i>Figura 3.3.</i> Diagrama esquemático de la fuente fija de voltaje de CD que alimenta el CODKYZ. _____	39
<i>Figura 3.4.</i> Diagrama esquemático de la etapa de acoplamiento. _____	40
<i>Figura 3.5.</i> Diagrama a bloques de los puertos de comunicaciones seriales del CODKYZ. _____	40
<i>Figura 3.6.</i> Diagrama de conexiones del CI DS275 [25]. _____	41
<i>Figura 3.7.</i> Conector DB9 hembra de 9 pines empleado para la salida del puerto RS-232. _____	41
<i>Figura 3.8.</i> Diagrama de conexiones del CI MAX489 [26]. _____	42
<i>Figura 3.9.</i> Terminales con tornillos empleadas para conexión del puerto RS-485. _____	43
<i>Figura 3.10.</i> Diagrama a bloques del radio-módem MaxStream. _____	43
<i>Figura 3.11.</i> Módulo 9Xtream: a) tarjeta de interfaz, b) radio-módem, c) antena externa. _____	44
<i>Figura 3.12.</i> Diagrama funcional de la etapa digital implementada en el AT90S2313. _____	45
<i>Figura 3.13.</i> Lectura de los pulsos KYZ por medio de los pines del microcontrolador. _____	47
<i>Figura 3.14.</i> Cuantificación de los pulsos KYZ y su respaldo en memoria RAM. _____	47
<i>Figura 3.15.</i> Representación de los pulsos KYZ y el momento en que se actualizan los registros. _____	48
<i>Figura 3.16.</i> Mapa de memoria del CODKYZ. _____	51

Figura 3.17. Diagrama de flujo del algoritmo para calcular el CRC-16 de una trama MODBUS.	52
Figura 3.18. Diagrama de flujo de la función principal del programa implementado en el CODKYZ.	58
Figura 3.19. Diagrama de flujo de la función <i>Atiende_Trama_MODBUS</i> .	59
Figura 4.1. Representación de las fases del modelo incremental aplicado al desarrollo del Soft-KYZ.	62
Figura 4.2. Fases del desarrollo del software.	63
Figura 4.3. Bloques funcionales del Soft-KYZ.	65
Figura 4.4. Diseño arquitectónico del Soft-KYZ.	67
Figura 4.5. Diagrama de flujo de la tarea <i>Monitoreo</i> de los pulsos KYZ.	69
Figura 4.6. Icono de entradas y salidas del módulo <i>Crea Archivo</i> .vi.	70
Figura 4.7. Icono de entradas y salidas del módulo <i>Monitoreo</i> .vi.	71
Figura 4.8. Icono de entradas y salidas del módulo <i>Tareas MODBUS</i> .vi.	72
Figura 4.9. Icono de entradas y salidas del módulo <i>Escribe Respuesta</i> .vi.	72
Figura 4.10. Icono de entradas y salidas del módulo <i>Calcula Variables</i> .vi.	72
Figura 4.11. Icono de entradas y salidas del módulo <i>Escribe Archivo</i> .vi.	73
Figura 4.12. Diagrama de flujo de la tarea <i>Registro de las transacciones MODBUS</i> .	74
Figura 4.13. Icono de entradas y salidas del módulo <i>Transacciones</i> .vi.	75
Figura 4.14. Icono de entradas y salidas del módulo <i>Incrementa Datos</i> .vi.	75
Figura 4.15. Icono de entradas y salidas del módulo <i>Solicita Contadores</i> .vi.	76
Figura 4.16. Diagrama de flujo de la tarea <i>Configuración de los dispositivos</i> .	77
Figura 4.17. Icono de entradas y salidas del módulo <i>Tareas de Configuración</i> .vi.	78
Figura 4.18. Icono de entradas y salidas del módulo <i>Detecta CODKYZ</i> .vi.	78
Figura 4.19. Icono de entradas y salidas del módulo <i>Configura Puerto</i> .vi.	78
Figura 4.20. Diagrama de flujo de la tarea <i>Demostración del protocolo MODBUS</i> .	79
Figura 4.21. Icono de entradas y salidas del módulo <i>MODBUS</i> .vi.	80
Figura 4.22. Estructura jerárquica de control del Soft-KYZ.	81
Figura 4.23. Pruebas de caja blanca y caja negra ejecutadas en dos módulos del Soft-KYZ.	82
Figura 4.24. Interfaz gráfica de usuario del Soft-KYZ.	83
Figura 5.1. Diagrama de flujo del procedimiento de pruebas.	87
Figura 5.2. Diagrama de instalación del sistema para las pruebas de laboratorio.	87
Figura 5.3. Instalación del sistema para las pruebas de laboratorio usando el puerto RS-232.	88
Figura 5.4. Instalación del sistema para las pruebas de laboratorio usando el puerto RS-485.	88
Figura 5.5. a) Diagrama esquemático y b) fotografía del emulador de relevadores KYZ.	89
Figura 5.6. a) Detección y b) reporte de datos de identificación del CODKYZ.	91
Figura 5.7. Datos de cuantificación de los pulsos KYZ reportados por el CODKYZ.	91
Figura 5.8. Datos del <i>monitoreo</i> y resultados finales de los parámetros de consumo eléctrico.	92
Figura 5.9. Curvas de potencia reportadas por el Soft-KYZ durante el <i>monitoreo</i> .	93
Figura 5.10. Curvas de potencia del <i>monitoreo</i> completo dibujadas en Microsoft Excel.	94
Figura 5.11. Curvas de consumo de energía <i>monitoreo</i> completo dibujadas en Microsoft Excel.	94
Figura 5.12. Datos finales de las transacciones hechas con el puerto RS-232.	95
Figura 5.13. Datos finales de las transacciones hechas con el puerto RS-485.	96
Figura 5.14. Diagrama de instalación del sistema para las pruebas de campo.	98
Figura 5.15. Aspecto de la instalación del sistema de pruebas de campo.	98
Figura 5.16. Ubicación de la antena en la instalación de campo.	99
Figura 5.17. Conexión de los componentes centrales: PM650, CODKYZ y radio-módem.	100
Figura 5.18. Aspecto real de la instalación hecha en sala de control.	100
Figura 5.19. Datos del <i>monitoreo</i> hecho para las pruebas de campo.	101
Figura 5.20. Gráficas reportadas por el Soft-KYZ durante el <i>monitoreo</i> .	102
Figura 5.21. Curva de potencia activa hecha en Microsoft Excel a partir del archivo de reporte.	102
Figura 5.22. Curva de consumo de energía activa realizada en Microsoft Excel.	103
Figura 5.23. Curvas de demanda medida y demanda máxima realizada en Microsoft Excel.	103
Figura 5.24. Encabezado, registro de demanda máxima y parte final del archivo de reporte.	104
Figura 5.25. Resultados finales del <i>monitoreo</i> reportados por el Soft-KYZ.	104
Figura 5.26. Resultados finales del <i>monitoreo</i> reportados por el medidor PM650.	105
Figura 5.27. Datos finales de las transacciones hechas con el enlace inalámbrico.	105
Figura A.1. Configuración de los pines del AT90S2313.	A-2

Figura B.1. Ubicación de los componentes del CODKYZ. _____	B-3
Figura B.2. Conexión de las terminales K-Y del PM650 a las entradas PKWh del CODKYZ. _____	B-4
Figura B.3. Ubicación de los puentes que seleccionan el estándar de comunicaciones. _____	B-5
Figura B.4. Conexión del CODKYZ y la computadora a través del puerto RS-232. _____	B-6
Figura B.5. Conexión del CODKYZ a una red MODBUS 4W a través del puerto RS-485 de 4 hilos. _____	B-7
Figura B.6. Conexión del CODKYZ a una red MODBUS 2W a través del puerto RS-485 de 2 hilos. _____	B-7
Figura B.7. Conexión del CODKYZ al radio módem a través del puerto RS-4232. _____	B-8
Figura B.8. Diagrama esquemático del CODKYZ. _____	B-12
Figura B.9. Diagrama de circuito impreso del CODKYZ. _____	B-13

Índice de tablas

<i>Tabla i.1.</i> Consumo de energía eléctrica en México de enero a septiembre de 2003. _____	xxv
<i>Tabla 1.1.</i> Efecto del factor de potencia en la facturación eléctrica. _____	5
<i>Tabla 2.1.</i> MODBUS ubicado en las capas del modelo OSI. _____	16
<i>Tabla 2.2.</i> Organización de las direcciones en MODBUS. _____	20
<i>Tabla 2.3.</i> Códigos de función MODBUS implementados en dispositivos sobre línea serial. _____	24
<i>Tabla 2.4.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x03. _____	25
<i>Tabla 2.5.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x04. _____	25
<i>Tabla 2.6.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x06. _____	26
<i>Tabla 2.7.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x10. _____	26
<i>Tabla 2.8.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x17. _____	27
<i>Tabla 2.9.</i> Subfunciones de diagnóstico de la función 0x08. _____	28
<i>Tabla 2.10.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x08, 0x00. _____	29
<i>Tabla 2.11.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x0B. _____	29
<i>Tabla 2.12.</i> Categorías de los datos de identificación de un dispositivo MODBUS. _____	30
<i>Tabla 2.13.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x2B, 0x0D. _____	31
<i>Tabla 2.14.</i> Códigos de excepción implementados en dispositivos sobre línea serial. _____	31
<i>Tabla 2.15.</i> Definición de las líneas de la configuración 2W. _____	32
<i>Tabla 2.16.</i> Definición de las líneas de la configuración 4W. _____	33
<i>Tabla 2.17.</i> Señales definidas por el estándar EIA/TIA-232 para una comunicación punto a punto. _____	35
<i>Tabla 3.1.</i> Señales eléctricas del CI MAX489 de MAXIM [26]. _____	42
<i>Tabla 3.2.</i> Características principales de operación del radio-módem modelo 9XStream [27]. _____	44
<i>Tabla 3.3.</i> Características principales de la antena ocupada para enlace inalámbrico. _____	44
<i>Tabla 3.4.</i> Registros mantenidos de configuración del CODKYZ. _____	50
<i>Tabla 3.5.</i> Valores por defecto de los registros mantenidos de configuración del CODKYZ. _____	50
<i>Tabla 3.6.</i> Registros de entrada del CODKYZ. _____	53
<i>Tabla 3.7.</i> Registros contadores de eventos de comunicación del CODKYZ. _____	53
<i>Tabla 3.8.</i> Códigos de funciones MODBUS que soporta el CODKYZ. _____	55
<i>Tabla 3.9.</i> Códigos de subfunciones MODBUS que soporta el CODKYZ. _____	55
<i>Tabla 3.10.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x65. _____	55
<i>Tabla 3.11.</i> Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x69. _____	55
<i>Tabla 3.12.</i> Direcciones lógicas, físicas y del campo de datos de los registros mantenidos. _____	56
<i>Tabla 3.13.</i> Direcciones lógicas, físicas y del campo de datos de los registros de entrada. _____	56
<i>Tabla 3.14.</i> Códigos de excepción soportados por el CODKYZ. _____	57
<i>Tabla 4.1.</i> Tareas MODBUS más comunes entre el Soft-KYZ y el CODKYZ. _____	71
<i>Tabla 4.2.</i> Datos de las transacciones MODBUS que registra el Soft-KYZ. _____	73
<i>Tabla 5.1.</i> Valores de registros de pulsos KYZ esperados y los reportados por el CODKYZ. _____	90
<i>Tabla 5.2.</i> Cálculos y valores esperados del <i>monitoreo</i> hecho en las pruebas de laboratorio. _____	93

Tabla 5.3. Comparación de la eficiencia del estándar RS-485 a 4 y 2 hilos. _____	96
Tabla 5.4. Comparación de los resultados del <i>monitoreo</i> reportados por el Soft-KYZ y el PM650. __	105
Tabla B.1. Características eléctricas del CODKYZ. _____	B-2
Tabla B.2. Conexión entre los relevadores KYZ del PM650 y las entradas de pulsos del CODKYZ. B-5	B-5
Tabla B.3. Parámetros de comunicación MODBUS del CODKYZ. _____	B-9
Tabla B.4. Funciones MODBUS soportadas por el CODKYZ. _____	B-9
Tabla B.5. Subfunciones de diagnóstico (función 08) MODBUS soportadas por el CODKYZ. _____	B-9
Tabla B.6. Códigos de excepción MODBUS soportadas por el CODKYZ. _____	B-10
Tabla B.7. Registros mantenidos (configuración) del CODKYZ. _____	B-10
Tabla B.8. Valores por omisión de los registros mantenidos del CODKYZ. _____	B-10
Tabla B.9. Registros de entrada del CODKYZ. _____	B-11
Tabla B.10. LEDS indicadores de funcionamiento del CODKYZ. _____	B-11
Tabla B.11. Lista de componentes del CODKYZ. _____	B-13
Tabla B.12. Problemas más comunes que se presentan al usar el CODKYZ y cómo se resuelven. B-14	B-14

Resumen

En este documento de tesis se describe el diseño y desarrollo de un dispositivo electrónico, llamado Codificador de Pulsos KYZ (CODKYZ), el cual codifica y almacena la información de las salidas pulsadas KYZ de los medidores electrónicos de energía eléctrica que se instalan en las empresas para propósitos de facturación. El elemento central es el microcontrolador AT90S2313 de la firma Atmel, en el cual se ha implementado el soporte del protocolo de comunicaciones MODBUS-RTU para que el dispositivo pueda conectarse en una red industrial MODBUS.

Asimismo, se ha desarrollado un instrumento virtual en LabVIEW que cumple con las características básicas de una Estación Maestra MODBUS. Con este software se configura el CODKYZ y se prueba su funcionamiento consultando los datos que el hardware almacena y con ellos se calculan y grafican las variables de consumo eléctrico más importantes de la planta. La transmisión de datos se hace bajo MODBUS empleando un par de radio-módems modelo 9XStream de 900MHz mediante un enlace inalámbrico, o bien, sobre un bus de línea serial usando los estándares RS-232 o RS-485 de 2 ó 4 hilos.

Además, este software proporciona una herramienta que muestra el funcionamiento del protocolo MODBUS mediante la generación, transmisión y recepción de tramas interactuando con dispositivos que soporten dicho protocolo.

Abstract

This thesis describes the design and development of an electronic device called CODKYZ (*Codificador de Pulsos KYZ*) which encodes and stores information about output KYZ pulses of electronic energy meters used for billing in companies. The main component is the Atmel AT90S2313 microcontroller which has been programmed to support MODBUS-RTU communications protocol so that the device can be included in a MODBUS industrial network.

In this project, a Virtual Instrument has also been developed in LabVIEW in order to configure and test the CODKYZ. This software works as a basic MODBUS Master Station and has a routine that requests CODKYZ's data which is used to compute and plot the most important electrical variables of energy consumption. The data can be transmitted either wirelessly by means of a pair of 900 MHz radio-modems 9XStream model, or over a serial line using RS-232 or RS-485 electrical communication standards.

This software also provides a tool to demonstrate how MODBUS protocol works. It permits the user to construct, transmit and receive MODBUS frames in order to communicate with any device that supports this protocol.

Introducción

La energía eléctrica es un recurso artificial vital para el desarrollo social y económico de cualquier país. Su importancia es tal, que diariamente se consume energía eléctrica en todo el mundo para aplicaciones domésticas, comerciales, servicios y sobre todo industriales.

En el período comprendido de enero a septiembre de 2003, en México se consumió un total de 129,454 gigavatios-hora (GWh, *Gigawatts-hour*) de energía eléctrica generada por la CFE (*Comisión Federal de Electricidad*) [URL1]. El 76.92% de esta energía corresponde al consumo público detallado en la *tabla i.1*.

Tabla i.1. Consumo de energía eléctrica en México de enero a septiembre de 2003.

Sector de consumo	Número de clientes	Total de Consumo (GWh)	Consumo promedio por mes (GWh)
Doméstico	18,544,055	24,916	2.768
Comercial	2,195,812	6,738	0.749
Industrial	135,127	58,611	6.512
Servicios	139,350	3,198	0.355
Agrícola	99,234	6,118	0.680
Totales	21,113,578	99,581	11.064

En la tabla anterior se observa que el sector industrial consume la mayor parte de la energía destinada al consumo público. Por lo tanto, debido a los costos de inversión que la CFE realiza para suministrarla, es fácil comprender que la energía eléctrica es un bien necesario que puede resultar muy caro para una empresa industrial si no tiene un control adecuado de su uso, sobre todo para las fábricas que trabajan con maquinas de alto consumo eléctrico porque incrementan el valor de los parámetros que la CFE usa para calcular el recibo eléctrico. Dichos parámetros son: consumo de energía, factor de potencia (FP) y demanda máxima [URL1].

Al tomar medidas para corregir alguno de estos parámetros, se consigue un ahorro significativo en la electricidad que demanda la empresa y por lo tanto, en el importe que debe pagar. Por ejemplo, existen controladores que ocupan bancos de capacitores y sistemas de conexión y desconexión automática de maquinaria que

ayudan a mantener el FP y la demanda máxima en valores preestablecidos por el consumidor.

Afortunadamente, la tecnología aplicada a la industria facilita a las empresas instalar sistemas completos de vigilancia (*monitoreo*) para tener un mejor control del consumo eléctrico y de otros procesos industriales de manera automática. Dentro de este conjunto de técnicas se encuentran las redes de comunicaciones industriales (RCI), los buses de campo, las comunicaciones inalámbricas y el software o programas de adquisición de datos y control [1, 2].

Las RCI se emplean para conocer y controlar el estado de los procesos que se ejecutan en lugares distantes dentro o fuera de la empresa, por ejemplo, el consumo de energía eléctrica de la planta [3]. Lo anterior se logra midiendo las variables fundamentales del proceso industrial con instrumentos instalados en campo, éstos a su vez reportan la información a una sala de control a través de los llamados buses de campo, los cuales se constituyen por los protocolos de comunicaciones y las interfaces eléctricas que establecen las reglas de comunicación entre los dispositivos de la red [2]. Los principales buses de campo que se utilizan actualmente en la industria son: *Profibus*, *CAN*, *DeviceNet*, *MODBUS*, *Fieldbus Foundation* y *HART* [2, 4, 5].

Por otro lado, el desarrollo de las comunicaciones inalámbricas ha mejorando las aplicaciones de acceso remoto, de modo que los sistemas inalámbricos constituyen un método seguro, flexible y de fácil adaptación a los cambios realizados en la estructura física de las empresas (mantenimiento, reparación y reubicación de maquinaria y equipo) [6]. Actualmente existen módulos de bajo costo para comunicación inalámbrica en las bandas de frecuencias reservadas para aplicaciones industriales, médicas y científicas (*ISM Band*, *Industrial and Scientific and Medical Band*). Además, no se requiere contar con una licencia para usar estos equipos que operan en los rangos de frecuencias de 902-928 Mhz y 2.40-2.4835 GHz [URL2].

Adicionalmente, el software utilizado para la adquisición de datos y control de los dispositivos remotos, es variado. Los programas más usados en este ambiente presentan una interfaz gráfica que permite a los usuarios utilizarlos como si estuvieran operando los equipos directamente. A este tipo de programa se le denomina instrumento virtual (VI, *Virtual Instrument*) [7].

Entre los lenguajes de programación que permiten crear VIs destacan LabVIEW desarrollado por *National Instruments* [URL3]; VEEPro, por *Agilent Technologies* [URL4]; Intellution de *Emerson Process Management* y lenguajes visuales como MS Visual C++ y MS Visual Basic.

Por todo lo anterior, resulta útil que las empresas utilicen esta tecnología para instalar RCI que incluyan dispositivos para analizar el comportamiento de las variables de consumo eléctrico: energía eléctrica, demanda máxima, FP y factor de carga (FC) [8]. Conociendo esta información, es posible hacer un diagnóstico del consumo actual de la planta y con ello emprender medidas correctivas para hacer más eficiente el uso de la electricidad.

En el mercado existe una gran variedad de equipos de medición que facilitan el análisis de las variables eléctricas. Destacan, por ejemplo, los de las marcas *Yokogawa*, *PowerSight*, *Aeroflex* y *SquareD* [URL5]. Estos instrumentos requieren

de transformadores de corriente y/o de voltaje que reducen la potencia que entra al equipo de medición, ocupan un microprocesador para llevar un registro de todos los parámetros que miden y facilitar la transferencia de datos hacia una computadora. Además, poseen pantallas para mostrar la información y un software para realizar estudios de comportamiento de la planta. El costo de un sistema de este tipo varía en función del número de mediciones que soportan y del área que puede cubrir.

Por otro lado, los medidores de energía que la CFE instala para propósitos de facturación, tienen la capacidad de medir las energías activa y reactiva, el factor de potencia y la demanda máxima de la planta. Adicionalmente, poseen salidas a relevador para generar una señal de sincronía cada cinco minutos [9] y para reportar el consumo actual de energías activa y reactiva. Contando el número de pulsos y midiendo el tiempo de duración, es posible llevar un registro de consumo de energía y demanda de potencia que presenta una planta.

Sin embargo, la mayoría de veces, el medidor de CFE se instala en la entrada de las fábricas o edificios y se necesitan al menos cuatro cables para llevar las señales pulsadas hasta una sala de control. En ocasiones, esto no es práctico ni factible. Una solución a este problema consiste en instalar un codificador de los pulsos que genera el medidor. Así, este dispositivo se encarga de cuantificar y codificar los pulsos para que, de forma inalámbrica, transmita la información del consumo eléctrico hasta una sala de control, la cual puede formar parte de una red de *monitoreo*, automatización y control de procesos industriales a través de una conexión cableada o inalámbrica.

La *figura i.1* muestra el esquema una RCI que utiliza el bus de campo MODBUS para interconectar diversos dispositivos. Este bus de campo se constituye por el uso del protocolo de comunicaciones MODBUS y la interfaz eléctrica EIA/TIA-485 para comunicar todos los elementos que pertenecen a la red. Lo anterior se efectúa bajo los lineamientos de conexión y cableado que establece la norma de MODBUS. Al contar con una red de este tipo, es posible instalar diferentes dispositivos que obtengan información o realicen acciones de control, cuya recopilación de datos y envío de órdenes se realice mediante una computadora central desde una sala de control.

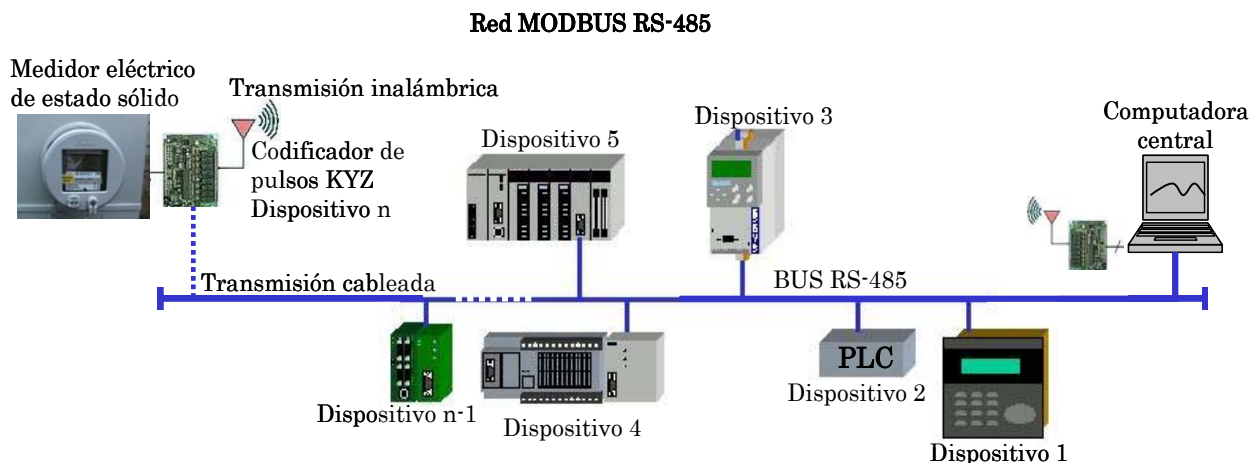


Figura i.1. Red MODBUS RS-485: *monitoreo*, automatización y control de procesos industriales.

En este trabajo de tesis se plantea el diseño y la construcción de un codificador de pulsos KYZ, el cual se conecta directamente a las salidas pulsadas del medidor de energía de la CFE (o cualquier otro de este tipo) y reporta información sobre el consumo eléctrico de la planta en forma inalámbrica o cableada. El componente central es el microcontrolador AT90S2313 de Atmel (*apéndice A*), en el que se implementa el protocolo de comunicaciones MODBUS para ser integrado en una red industrial MODBUS RS-485. La conexión a la red puede ser inalámbrica a través del Radio-Módem MaxStream de 900 MHz o alambrada, utilizando el puerto RS-485 de 2 ó 4 hilos.

Como software de pruebas se desarrolló un programa de computadora en LabVIEW, el cual se encarga de enviar tramas de solicitud de datos al dispositivo siguiendo el protocolo de comunicaciones MODBUS. Es así como se obtiene la información sobre el consumo eléctrico que el dispositivo remoto ha adquirido del medidor. Con esta información se realizan los cálculos necesarios para crear gráficas de comportamiento de consumo de energía, FP y de demanda de potencias activa, reactiva y aparente que registra la planta analizada.

Además, permite configurar el dispositivo remoto desde una sala de control u oficina. Esto facilita a la gerencia de una empresa, enterarse de cómo se está utilizando la electricidad y así poder aplicar medidas de control sin necesidad de verificarlo directamente en campo [3].

Por todo lo anterior, a continuación se presentan los objetivos que se persiguen con este trabajo de tesis.

Objetivo general

Diseñar y construir un sistema cuantificador y codificador de los pulsos KYZ generados por los medidores electrónicos de energía eléctrica, de modo que la información que representan pueda ser transferida a una computadora personal de forma inalámbrica o cableada, utilizando el protocolo de comunicaciones MODBUS.

Objetivos específicos

- Diseñar y construir un módulo de acoplamiento de 5 V de corriente directa (CD) para generar pulsos KYZ en niveles TTL empleando los relevadores de salida de los medidores electrónicos de energía.
- Implementar el soporte del protocolo de comunicaciones MODBUS en el microcontrolador AT90S2313, para que el codificador tenga la capacidad de incluirse como Esclavo en una RCI que opere bajo este protocolo.
- Adaptar el radio-módem MaxStream de 900 MHz en el sistema para tener la capacidad de comunicación inalámbrica bajo el estándar RS-232.
- Implementar los puertos de comunicación serial bajo los estándares RS-232 y RS-485 de 2 y 4 hilos para enlaces cableados punto-punto y multipunto, respectivamente.
- Elaborar la tarjeta de circuito impreso del hardware.
- Desarrollar un software de configuración y pruebas en LabVIEW que opere bajo MODBUS para solicitar y modificar los datos del codificador y verificar

su funcionamiento correcto, así como para crear gráficas de FP y de demanda de potencias activa, reactiva y aparente.

- Implementar en el software de pruebas la capacidad de mostrar el funcionamiento del protocolo de comunicaciones MODBUS.

Contenido del documento de tesis

Este documento de tesis se divide en cinco capítulos descriptivos del sistema y en las conclusiones obtenidas.

En el capítulo uno se describen los parámetros que repercuten en la facturación eléctrica y se explica la importancia de su *monitoreo* y control a través de los medidores electrónicos proporcionados por la CFE, haciendo hincapié en la relevancia que tiene controlar la demanda máxima de potencia.

El capítulo dos contiene la descripción del protocolo de comunicaciones MODBUS para su implementación en equipos que reportan datos a través de una línea serial. De manera particular se explican las funciones que el dispositivo propuesto puede ejecutar.

En el capítulo tres se describe el diseño y la implementación del hardware del sistema. Además, se detalla el programa realizado en el microcontrolador AT90S2313.

En el capítulo cuatro se describe el desarrollo del software de pruebas del sistema realizado en LABVIEW.

El capítulo cinco explica las pruebas realizadas al sistema completo y los resultados finales.

Por último se incluyen las conclusiones que se han obtenido al finalizar el trabajo de tesis descrito en este documento y se mencionan los trabajos futuros que se pueden realizar tomando éste como punto de partida.

1. Costo y medición de la energía eléctrica

En la mayoría de los países, las empresas se preocupan seriamente por el consumo de energía eléctrica, principalmente las del sector industrial, pues los gastos ocasionados por emplear este recurso tanto para fines de producción como de operación, llegan a ser muy elevados.

Tomando en cuenta lo anterior, en este capítulo se definen las variables eléctricas de una instalación trifásica, se citan los aspectos de mayor importancia de la facturación de energía eléctrica para las empresas en México y se describen también las medidas de prevención que deben tomarse para evitar gastos elevados en electricidad. Además, se profundiza en el tema de la demanda máxima y sus controladores automáticos, se describe el funcionamiento de los medidores electrónicos empleados para registrar el consumo de energía eléctrica y el uso de sus salidas pulsadas para implementar sistemas de *monitoreo* y control de los parámetros eléctricos de una planta.

1.1. Variables eléctricas de una instalación trifásica

En lo que se refiere al sector industrial, la mayoría de las empresas se abastecen de energía eléctrica a través de una instalación trifásica que instala la empresa de suministro eléctrico.

Una red eléctrica trifásica esta compuesta por tres señales sinusoidales de voltaje con una frecuencia de 60 Hz, las cuales se encuentra desfasadas 120° entre cada una de ellas. Normalmente, éstas provienen de un transformador de potencia que baja el voltaje de las líneas de distribución de 13.2 KV a su valor nominal, 120 ó 220 Vrms respecto a una cuarta línea llamada neutro.

Cuando una carga eléctrica se alimenta con una sola fase (arbitraria) del transformador, ésta proporciona una señal sinusoidal de voltaje $v(t)$ y una de corriente $i(t)$ de modo que a la carga se le suministra una potencia instantánea en kilovatios (KW, *kilowatts*) igual a [11]:

Debido a que $p(t)$ varía en el tiempo, es necesario obtener la potencia promedio que se suministra a la carga eléctrica en un intervalo de tiempo determinado a partir de la siguiente expresión:

$$P = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} v(t) \cdot i(t) \cdot dt \quad (2)$$

en la cual:

P = potencia promedio suministrada a la carga en el intervalo de tiempo [KW]

$v(t)$ = voltaje sinusoidal proveniente de una fase del transformador [V],

$i(t)$ = corriente sinusoidal demandada por la carga eléctrica [A],

t_0, t_1 = límites inferior y superior, respectivamente, del intervalo de tiempo [s].

Sin embargo, para efectos de facturación, la empresa de suministro eléctrico instala medidores de energía que calculan el voltaje eficaz (V_{ef}), la corriente eficaz (I_{ef}) y el ángulo (θ) por el cual el voltaje adelanta a la corriente en cada fase del transformador porque éstos son los parámetros que se ocupan para calcular la energía que una planta consume al final del periodo de facturación.

Estos medidores generalmente son electrónicos y se conectan a la salida del transformador trifásico y reciben las tres señales de voltaje y corriente a través de transformadores que aíslan las señales de entrada y las disminuyen a niveles que estos dispositivos pueden manejar.

Para obtener el ángulo de desfase θ entre el voltaje y la corriente, los medidores electrónicos comparan las señales y miden el tiempo que la señal de corriente tarda en pasar por cero después de la señal de voltaje.

Los valores eficaces de voltaje y corriente se obtienen a través de la integral numérica de los datos medidos durante un ciclo de la señal sinusoidal a partir de las siguientes ecuaciones:

$$V_{ef} = \sqrt{\frac{1}{T} \int_0^T v^2(t) \cdot dt} \quad (3)$$

$$I_{ef} = \sqrt{\frac{1}{T} \int_0^T i^2(t) \cdot dt} \quad (4)$$

en donde:

V_{ef} = valor eficaz de voltaje [Vrms],

I_{ef} = valor eficaz de corriente [Arms],

T = periodo de la señal sinusoidal de voltaje o corriente.

Idealmente, las señales $v(t)$ e $i(t)$ son sinusoidales puras, por lo tanto, sus valores eficaces se calculan de manera idéntica a partir de su valor pico (V_p) como:

$$V_{ef} = \frac{V_p}{\sqrt{2}} = 0.7071 \cdot V_p \quad (5)$$

Con estos parámetros, los medidores pueden calcular las potencias activa (P) en KW; reactiva (Q), en Kilovoltios-amperios reactivos (KVAR); y aparente (S), en Kilovoltios-amperios (KVA) que la carga eléctrica está demandando a la fase del transformador. Estas variables están dadas por las ecuaciones 6, 7 y 8, respectivamente.

$$P = V_{ef} I_{ef} \cos \theta \quad (6)$$

$$Q = V_{ef} I_{ef} \sen \theta \quad (7)$$

$$S = V_{ef} I_{ef} \quad (8)$$

Cuando la carga eléctrica ha exigido *potencia* a la fase del transformador durante un período de tiempo Δt , se dice que la planta ha consumido cierta cantidad de *energía eléctrica*. De esta manera, el consumo de energía activa (Ea) representa la cantidad de potencia activa que exigen los dispositivos para producir trabajo efectivo durante un tiempo determinado, la energía reactiva (Er) es la que ocupan las cargas inductivas, como motores, para generar campos magnéticos necesarios en su funcionamiento y la energía aparente (Eap) es la que se consume para producir las energías activa y reactiva.

La energía consumida por la carga, ya sea activa en kilovatios-hora (KWh), reactiva en KVARh o aparente en KVAh, se calcula multiplicando la potencia correspondiente por el tiempo de operación. Por ejemplo, la energía activa está dada como:

$$Ea = P\Delta t \quad (9)$$

donde:

Ea = Energía reactiva consumida [KWh],

P = Potencia activa promedio exigida por la carga eléctrica durante el intervalo [KW],

Δt = El intervalo de tiempo de consumo [h].

De manera análoga se calculan las energías reactiva y aparente.

Otro parámetro importante de una instalación eléctrica trifásica es el factor de potencia (FP), el cual es una medida de la eficiencia con la que se aprovecha la energía eléctrica, ya que permite saber qué porción de la potencia aparente está formada por potencia activa y qué porción, por reactiva, es decir, permite conocer qué cantidad de potencia se está ocupando realmente para producir trabajo efectivo.

Matemáticamente, el FP se define como el coseno del ángulo θ entre el voltaje y la corriente (ecuación 10) cuando se consume de energía reactiva; o bien, como la razón entre P y S (ecuación 11). El FP cantidad numérica sin unidad de medida.

$$FP = \cos \theta \quad (10)$$

$$FP = \frac{P}{S} \quad (11)$$

Las ecuaciones del 3 al 11, representan las operaciones matemáticas que efectúan los medidores electrónicos para calcular las variables eléctricas de consumo de energía para una sola fase del transformador, de modo que la energía total consumida por una planta eléctrica es la suma de la energía consumida en cada fase del transformador trifásico. De esta manera, los totales de energías activa (Ea), reactiva (Er) y aparente (Eap) entregados por el sistema eléctrico trifásico a la planta son:

$$Ea_{total} = Ea_{fase1} + Ea_{fase2} + Ea_{fase3} \quad (12)$$

$$Er_{total} = Er_{fase1} + Er_{fase2} + Er_{fase3} \quad (13)$$

$$Eap_{total} = Eap_{fase1} + Eap_{fase2} + Eap_{fase3} \quad (14)$$

donde el subíndice $fase_n$ ($n = 1, 2$ y 3 o a, b y c) representa la fase del transformador que suministró la energía correspondiente. De igual manera, el FP de la planta se calcula de manera continua como:

$$FP_{total} = \frac{FP_{fase1} + FP_{fase2} + FP_{fase3}}{3} \quad (15)$$

Sin embargo, al final de un período de facturación, el valor de FP sobre el cual se va a realizar el cobro a la empresa consumidora se calcula como:

$$FP_{facturable} = \frac{Ea_{total}}{Eap_{total}} \quad (11)$$

Los medidores electrónicos poseen un módulo de conexión externa que ofrece al consumidor un medio para calcular estas variables y conocer su consumo de energía eléctrica (*apartado 1.4.1*).

La *figura 1.1* muestra el diagrama unifilar de una red trifásica típica donde se observa la ubicación del transformador de potencia, los interruptores termomagnéticos que protegen las diferentes cargas eléctricas alimentadas y el punto donde el proveedor de electricidad coloca los medidores electrónicos para facturar el consumo de energía eléctrica.

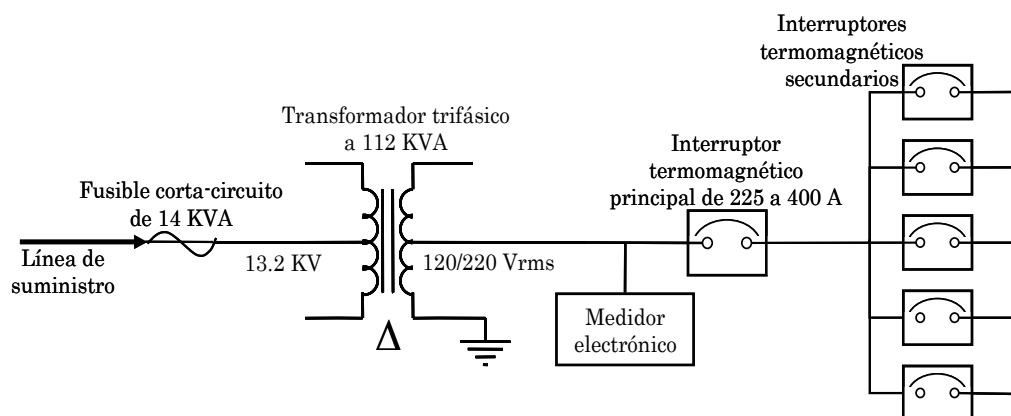


Figura 1.1. Diagrama unifilar de una red eléctrica trifásica.

1.2. Facturación de la energía eléctrica en México

En México, los principales cargos que la CFE¹ establece para consumidores con demanda mayor a 25 KW² son [URL1]:

- Consumo de energía
- Bonificación por alto FP o cargo por bajo FP
- Demanda máxima medida

El costo de la electricidad depende directamente de la cantidad de energía eléctrica consumida al término de cada período de facturación³. De acuerdo con la tarifa de cobro y la región en donde se encuentre la empresa, la CFE cobra un importe específico por cada KWh de energía consumido. Sin embargo, aún cuando el usuario no consume energía, debe pagar un cargo por servicio mínimo, pues mantenerlo conectado al suministro eléctrico representa un costo para la empresa suministradora de energía [URL1].

Cuando una fábrica utiliza máquinas de corriente alterna, como motores de inducción y transformadores, absorbe potencia reactiva para generar sus campos magnéticos [10, 11]. Cuando esto ocurre, se dice que las máquinas tienen un FP menor a la unidad y por lo tanto, también el consumo eléctrico de la empresa.

Los límites del FP son 0 y 1, su efecto en la facturación depende de si su valor es menor o mayor a 0.9. Cuando su valor está por debajo de 0.9 se establece una penalización sobre el valor del recibo por bajo FP y una bonificación en caso de ser mayor, ambos casos se detallan en la *tabla 1.1*.

Tabla 1.1. Efecto del factor de potencia en la facturación eléctrica.

Condición	Penalización o bonificación	Valores Máximos
FP < 0.9	Penalización por bajo FP $\text{Penalización}(\%) = \frac{3}{5} \times \left[\frac{0.9}{FP} - 1 \right] \times 100$	120%
FP > 0.9	Bonificación por alto FP $\text{Bonificación}(\%) = \frac{1}{4} \times \left[1 - \frac{0.9}{FP} \right] \times 100$	2.50%

Por otro lado, la CFE está interesada en que las empresas disminuyan la demanda de potencia eléctrica, porque el hecho de mantener un suministro de energía con una demanda excesiva le representa costos de inversión grandes. Debido a esto, la CFE establece un cargo por demanda máxima de potencia, es decir, cobra un precio específico por cada KW de demanda máxima registrada al final del período de facturación respetando también la región donde se encuentra el usuario. El concepto de demanda máxima se explica con mayor detalle en el *subcapítulo 1.3*.

¹ Empresa gubernamental que genera, transmite, distribuye y suministra la electricidad en México.

² A partir de esta demanda de potencia, la CFE considera una tarifa de servicio general en el que se incluyen los sectores comercial e industrial.

³ En México, generalmente es de un mes para el sector productivo (industria, comercio, etc).

1.2.1. Reducción de costos por consumo de energía eléctrica

Debido a que las empresas del sector productivo hacen un uso extenso de la electricidad, sobre todo las grandes fábricas de producción necesitan reducir los cargos generados por hacer mal uso de ella.

En este tipo de empresas, el FP y la demanda máxima son los principales cargos, ya que el consumo de energía eléctrica es un recurso necesario difícil de reducir porque la maquinaria debe operar de forma continua la mayor parte del tiempo.

Para mejorar el uso de la electricidad es necesario realizar una inversión de capital, la cual resulta económicamente factible si el ahorro anual, en costos de electricidad, supera o iguala la cantidad invertida [11]. No obstante lo anterior, en ocasiones el consumidor no tiene opción, sobre todo por el FP que es altamente penalizado si tiene un valor bajo.

Las empresas que utilizan máquinas con bajo FP instalan bancos de capacitores en la entrada del servicio eléctrico para corregirlo [11]. Sin embargo, debido a que el FP varía conforme se enciende o se apaga la maquinaria, resulta más útil instalar un banco de capacitores con control automático para conmutar continuamente entre ellos (*figura 1.2*). De este modo, las empresas buscan mantener un FP superior a 0.9.

Para controlar la demanda máxima, los usuarios instalan dispositivos que *monitorean* el consumo de energía eléctrica y, eventualmente, apagan equipos cuya operación no es trascendental si detectan que la demanda de potencia se acerca a un valor fijo preestablecido.

1.3. Demanda máxima

El costo mensual de electricidad suministrada a los usuarios no sólo depende de la energía total consumida, sino también de la eficiencia con que se consume. Para entender mejor este concepto se cita el siguiente ejemplo:

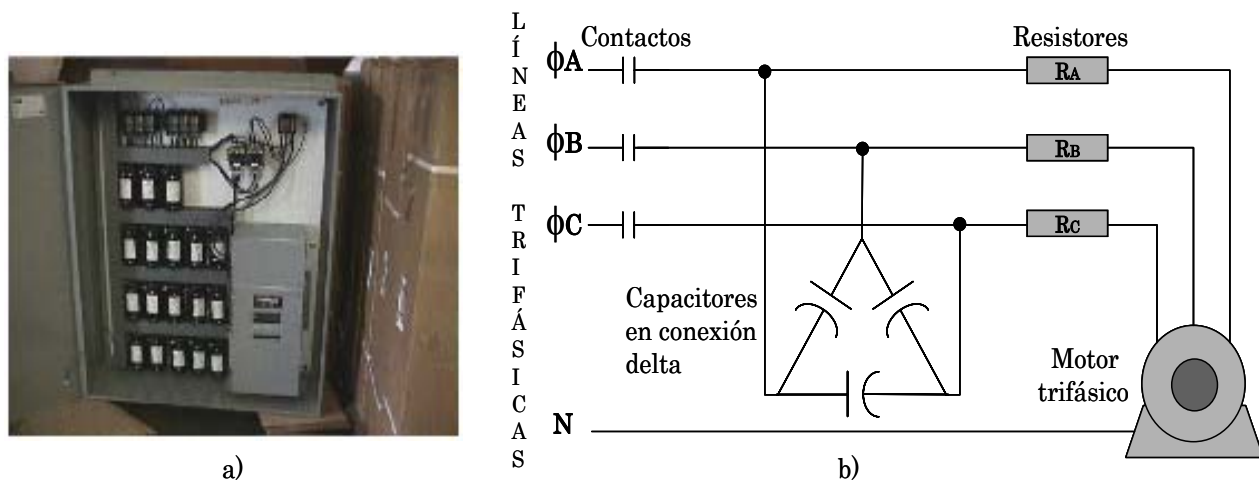


Figura 1.2. a) Banco de capacitores instalados para corregir el factor de potencia, b) conexión típica.

Dos fábricas, A y B, están conectadas respectivamente a una línea de alto voltaje a través de los transformadores T_A y T_B (figura 1.3). La fábrica A opera a plena carga día y noche todos los días de la semana, solicitando una potencia de 1100 KW a la línea de suministro, tal y como se ilustra en la figura 1.3a. Al final del mes (720 h), habrá consumido un total de $1100 \text{ KW} \times 720 \text{ h} = 792,000 \text{ KWh}$ de energía.

Supóngase que la fábrica B consume la misma cantidad de energía (792,000 KWh), pero su valor de carga varía continuamente haciendo oscilar su potencia de operación entre 100 y 4000 KW, como se muestra en la figura 1.3b. Por esta razón, la capacidad del transformador y la línea de transmisión que suministran a la fábrica B, debe ser mayor que la de los que suministran a la fábrica A. Por lo tanto, la empresa de suministro eléctrico debe invertir más capital para servir a la fábrica B. En consecuencia, esta fábrica debe pagar más por el mal uso de la electricidad.

No obstante, en la figuras 1.3a y 1.3b se pueden apreciar pequeños picos de potencia requeridos al encender motores o turbinas eléctricos, pero éstos no duran lo suficiente como para necesitar que se instale equipo de suministro de mayor capacidad y que deba cobrarse por ello. Ante tal situación, se establece un intervalo de tiempo para el cual se hace significativa la exigencia de potencia y por tanto, el consumo de energía.

Este lapso de tiempo se denomina *intervalo de demanda* y puede ser variable. Sin embargo, en la práctica se utilizan valores de 60, 30, 15 10, 3 ó 1 minutos, los cuales son submúltiplos de una hora [12]. Los medidores electrónicos calculan el promedio de potencia solicitada durante cada intervalo de demanda. A este promedio de potencia se le llama *demanda medida*. Consecuentemente, al valor más grande de *demanda medida* registrado en un período de facturación se le denomina *demanda máxima*. En el caso de las fabricas A y B del ejemplo anterior, las demandas máximas son 1100KW y 4000 KW, respectivamente. Al respecto, la CFE hace referencia a la demanda máxima como sigue:

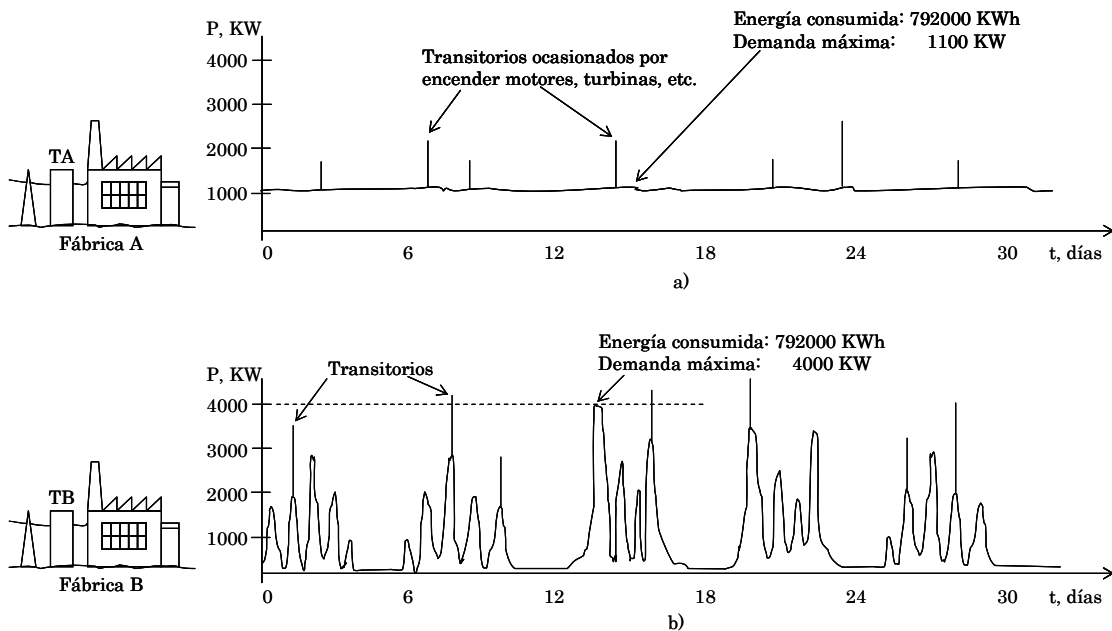


Figura 1.3. Consumo de energía de las fábricas A y B [11].

“Las Demandas Máximas medidas en los distintos periodos se determinarán mensualmente por medio de instrumentos de medición, que indican la demanda media en kilowatts, durante cualquier intervalo de 15 (quince) minutos del periodo en el cual el consumo de energía eléctrica sea mayor que en cualquier otro intervalo de 15 (quince) minutos en el periodo correspondiente” [URL1].

Esta es la razón por la que CFE cobra por cada KW de la demanda máxima registrada al final del período de facturación. En la *figura 1.4* se muestra una gráfica con los perfiles usuales de demanda de potencia eléctrica que tiene una planta en un día. Como se observa en ella, el consumo no es homogéneo, pero para efectos de facturación es el valor máximo el que se cobra.

1.3.1. Controladores de demanda máxima

Dada la definición de demanda máxima, es necesario tener un control para que su valor no sea elevado y por consiguiente, su costo disminuya. Como se mencionó en el *apartado 1.2.1*, existen dispositivos denominados controladores de demanda máxima (CDM) cuya definición es la siguiente:

“Dispositivo que mide la energía eléctrica que una planta está demandando y predice su comportamiento al final del intervalo de medición y, en caso de ser necesario, apaga las cargas eléctricas que no son prioritarias para la producción, de modo que la demanda máxima no exceda un umbral de consumo preestablecido” [11, 13, URL6].

Una parte importante de los CDM consiste en un sistema lector de información de la energía eléctrica que está consumiendo la planta. Dicha información la proporcionan los medidores de energía que instala la empresa de suministro, a través de un módulo con salidas para conexión externa. En el siguiente apartado se describen este tipo de medidores y sus salidas.

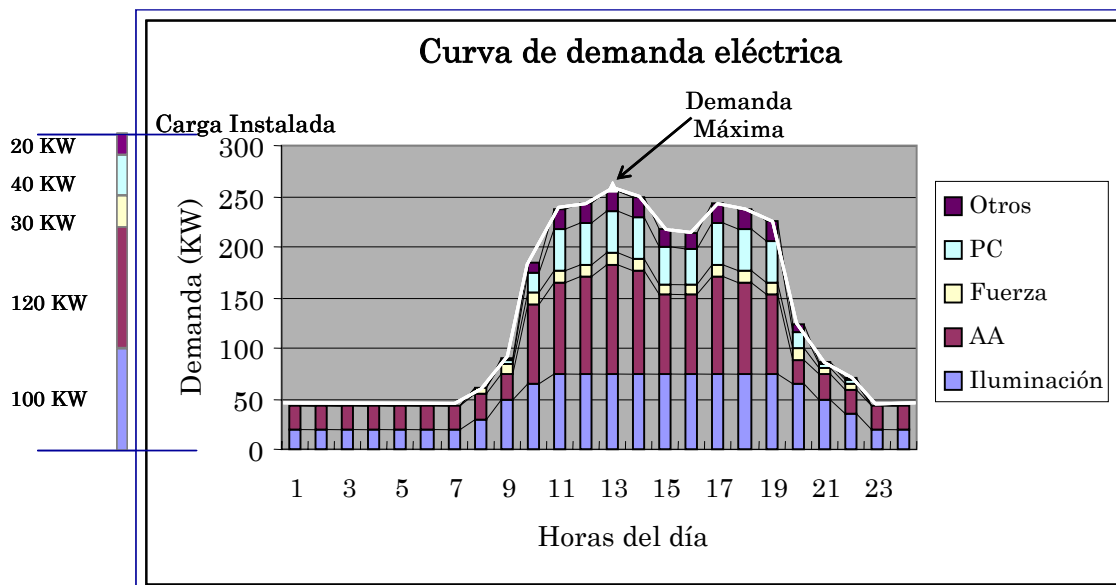


Figura 1.4. Perfil de la curva de demanda eléctrica de una planta en un día [URL6].

1.4. Medidores de energía eléctrica

La unidad de medida de la energía en el Sistema Internacional es el julio (J, joule). Sin embargo, por muchos años se ha usado el kilovatio-hora (KWh, *Kilowatt-hour*) para medir la energía eléctrica suministrada a la industria y usuarios particulares. Un KWh es exactamente igual a 3.6 MJ [11].

Los medidores de energía eléctrica están diseñados para multiplicar la potencia por el tiempo y son llamados vatíohorímetros o wathhorímetros, pues reportan, entre otros parámetros eléctricos, el consumo de energía eléctrica en KWh.

Por su principio de funcionamiento, los medidores de energía se clasifican en electromecánicos y electrónicos. En ambos tipos existen medidores de una y tres fases de voltaje, sólo que los electromecánicos son prácticamente el único tipo empleado para mediciones residenciales y casi todos los utilizados en la industria son electrónicos. Sin embargo, existen excepciones en las que se utilizan medidores electrónicos en instalaciones residenciales y electromecánicos, en la industria. Esto depende del consumo mensual de energía y de la precisión con la que se quiere medir [11]. En la *figura 1.5* se muestra un ejemplo de los medidores de energía electromecánicos y electrónicos que se utilizan en la actualidad.

Los elementos principales de los medidores electromecánicos son dos bobinas, un imán permanente, un sistema de engranes y un disco rotatorio soportado por un eje vertical. Cuando estos medidores se conectan a la instalación eléctrica, las dos bobinas y el imán permanente generan un campo magnético de tal manera que el sistema de engranes produce un torque en el disco haciéndolo girar. El número de vueltas del disco es proporcional a la energía (en KWh) suministrada a la carga [11].

En cambio, los medidores electrónicos basan sus mediciones en el procesamiento digital. Por lo tanto, las señales de corriente y voltaje que entran al medidor se digitalizan a través de convertidores analógico a digital (ADC, *Analog to Digital Converters*), de modo que los datos obtenidos son procesados digitalmente para calcular el valor de las variables eléctricas de una red trifásica. Contienen también una pantalla de cristal líquido (LCD, *Liquid Cristal Display*) para exhibir el valor actual de los parámetros de medición.

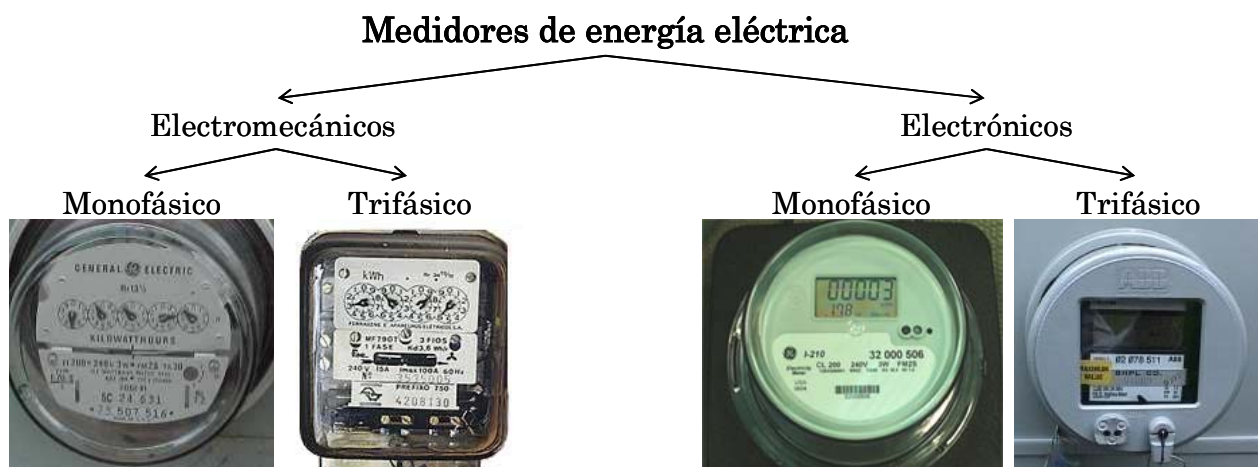


Figura 1.5. Clasificación de los medidores de energía eléctrica.

Puesto que el sistema codificador que se desarrolla en este trabajo de tesis adquiere sus señales de entrada específicamente de los medidores electrónicos, a continuación se explica el módulo de salida que se ocupa para adquirir la información de energía que este tipo de medidores registra.

1.4.1. Medidor electrónico con salidas a relevador KYZ

Los medidores electrónicos de energía tienen un módulo de salida adicional para indicar que la planta ha consumido cierta cantidad de energía.

La salida del medidor es un relevador de contacto seco, el cual puede ser electromecánico o de estado sólido y conmuta a una velocidad proporcional al consumo de energía. Existen dos tipos de relevadores que se distinguen por la forma en que conmutan:

Relevador tipo A: es un interruptor del tipo un polo un tiro (SPST, *Single Pole, Single Throw*). Su conmutación es alternada. Cada vez que el medidor registra un consumo de energía predefinido, el contacto se cierra por un intervalo de 100 ms, transcurrido este tiempo, se abre nuevamente y permanece en ese estado hasta que se vuelve a consumir otra cantidad de energía idéntica a la anterior (*figura 1.6*) [URL7].

Relevador tipo C: también es llamado relevador KYZ. Es un interruptor del tipo un polo dos tiros (SPDT, *Single Pole Double Throw*) [10]. La terminal K es el *común* del relevador; la terminal Y, el contacto *normalmente cerrado* y Z, el contacto *normalmente abierto*. La terminal K conmuta de estado entre *abierto* (K) y *cerrado* (Z) cada vez que se ha consumido una cantidad de energía predefinida (*figura 1.7*). Cada cambio de estado del relevador es llamado pulso KYZ. Los medidores electrónicos de energía que cuentan con este tipo relevadores en su salida son llamados *medidores con salida de pulsos KYZ* [URL7, URL8].

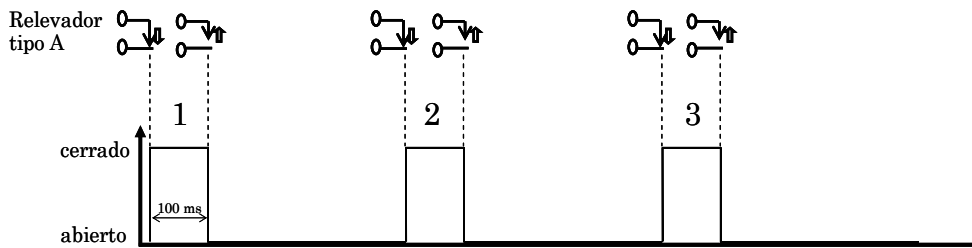


Figura 1.6. Funcionamiento de un relevador tipo A [URL7].

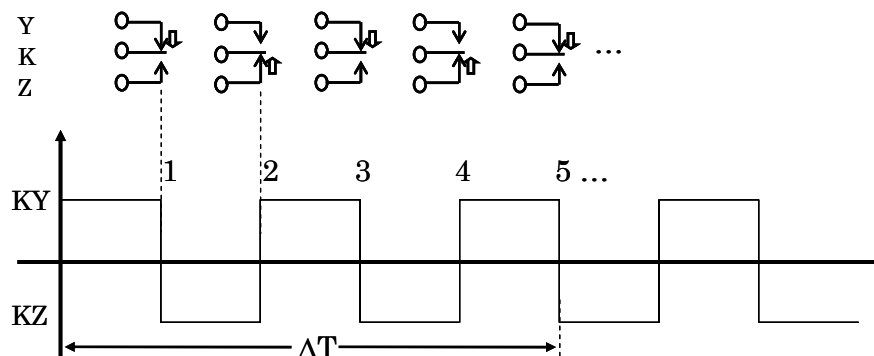


Figura 1.7. Funcionamiento de un relevador KYZ [URL7].

El sistema propuesto en esta tesis está enfocado sólo a los medidores con salida de pulsos KYZ debido a que son más comunes que los de relevador tipo A [URL8].

1.4.1.1. Medición de los pulsos KYZ

La mayoría de las entradas digitales de los sistemas de gestión de energía de 2 hilos sólo utilizan dos de los tres hilos de un relevador de pulsos KYZ. Esta aplicación se conoce como iniciador de pulso de 2 hilos [15].

La *figura 1.7* muestra una secuencia de pulsos generados con un relevador KYZ para una aplicación de este tipo, en ella se puede apreciar lo siguiente:

- En una aplicación de 2 hilos, la secuencia de pulsos es similar a los estados alternados de apertura y cierre de un contacto de forma A [10, URL7].
- La mayoría de las aplicaciones de pulso KYZ de 2 hilos utilizan un contacto de forma C, pero sólo conectan una de las partes del contacto, KY o KZ.
- El pulso KYZ se define como el tiempo que permanece una parte del relevador de forma C en estado abierto más la transición de abierto a cerrado o viceversa.
- En la *figura 1.7*, las transiciones se marcan como 1, 2, 3, etc., cada transición representa el tiempo en que el relevador cambia de KZ a KY y de KY a KZ. El receptor debe contar un pulso en los puntos que se muestran numerados.

1.4.1.2. Constante de energía K_e

Los medidores electrónicos utilizan un parámetro que indica la cantidad de energía eléctrica consumida por cada pulso que generan. Este valor es conocido como constante de energía de pulsos KYZ (K_e) [11, 14] y es específica para cada medidor o puede ser configurable si el medidor lo permite. Su unidad de medida es el kilovatio-hora por pulso (KWh/pulso), pero su valor numérico se emplea para generar pulsos KYZ tanto de energía activa en KW, como de energía reactiva KVARh.

1.4.1.3. Uso de los pulsos KYZ y la constante de energía K_e

Los medidores de la CFE obtienen la demanda eléctrica que consume la planta como la potencia promedio integrada en intervalos de 15 minutos, los cuales se forman de subintervalos de 5, es decir, cada 5 minutos se actualiza el valor de la demanda dentro de un intervalo de 15 minutos (10 minutos anteriores y los 5 minutos transcurridos recientemente). A este procedimiento de cálculo de la demanda se le conoce como *integración rolada* [12].

La *potencia promedio* que demanda una planta durante un intervalo de tiempo Δt se puede calcular con base en los pulsos KYZ que proporciona el medidor de energía. Para ello se cuenta el número de pulsos que transcurren en el intervalo Δt y se obtiene la energía registrada por el medidor a partir de la siguiente ecuación [15]:

$$E = Ke(Pulsos) \quad (16)$$

en la cual:

E = Energía consumida por la planta en el intervalo Δt ,

Ke = Constante de energía [KWh/pulso] y,

$Pulsos$ = Número de pulsos generados por el medidor en el intervalo Δt .

Mediante las ecuaciones 9 y 16 se obtiene:

$$P = \frac{(Ke)(pulsos)}{\Delta t} \quad (17)$$

donde:

P = Potencia promedio exigida por la planta en el intervalo Δt [kW], es decir, la demanda medida si Δt = intervalo de demanda del medidor.

Este procedimiento se realiza para calcular tanto la potencia activa como reactiva demandadas en el mismo intervalo de tiempo. Con estos valores se obtienen las demás variables de consumo eléctrico presentadas en el *subcapítulo 1.1*.

Tomando en cuenta lo anterior, se

concluye que contando el número de pulsos KYZ y midiendo su duración en tiempo es posible llevar un registro de consumo de energía y demanda de potencia de una planta.

La *figura 1.8* ilustra los conceptos y la relación que existe entre la constante de energía (Ke) y los pulsos KYZ.

Para la *figura 1.8a*, supóngase que una planta hipotética sólo tiene conectado un motor cuya potencia de operación es de 0.5 KW y que la constante de energía del medidor se configura a 0.5 KWh/pulso. De este modo, el medidor eléctrico generará un pulso KYZ cada hora. Si se conecta un molino de 1.5 KW, la potencia consumida por ambos equipos será de 2 KW, por lo tanto, los pulsos KYZ se generarán cuatro veces más rápido, es decir, cada quince minutos.

En la *figura 1.8b*, se presenta la misma situación, sólo que ahora el medidor tiene una Ke de 0.25 KWh/pulso, la mitad del caso anterior. Bajo estas condiciones, la razón de los pulsos KYZ será el doble del caso anterior a pesar de que la planta consume energía exactamente igual. Esto se debe a que el medidor registra un consumo de energía Ke dos veces más rápido. Cuando sólo trabaja el motor, cada 30 minutos se registra un consumo de 0.25 KWh y cuando trabajan juntos el motor y el molino, cada 7.5 minutos.

En síntesis, la energía eléctrica y la demanda máxima, así como los demás parámetros eléctricos de una red trifásica, se pueden obtener a través de la cuantificación de los pulsos KYZ generados por las salidas a relevador de los medidores electrónicos. Teniendo esta información se pueden crear sistemas automáticos de *monitoreo* y control de dichos parámetros.

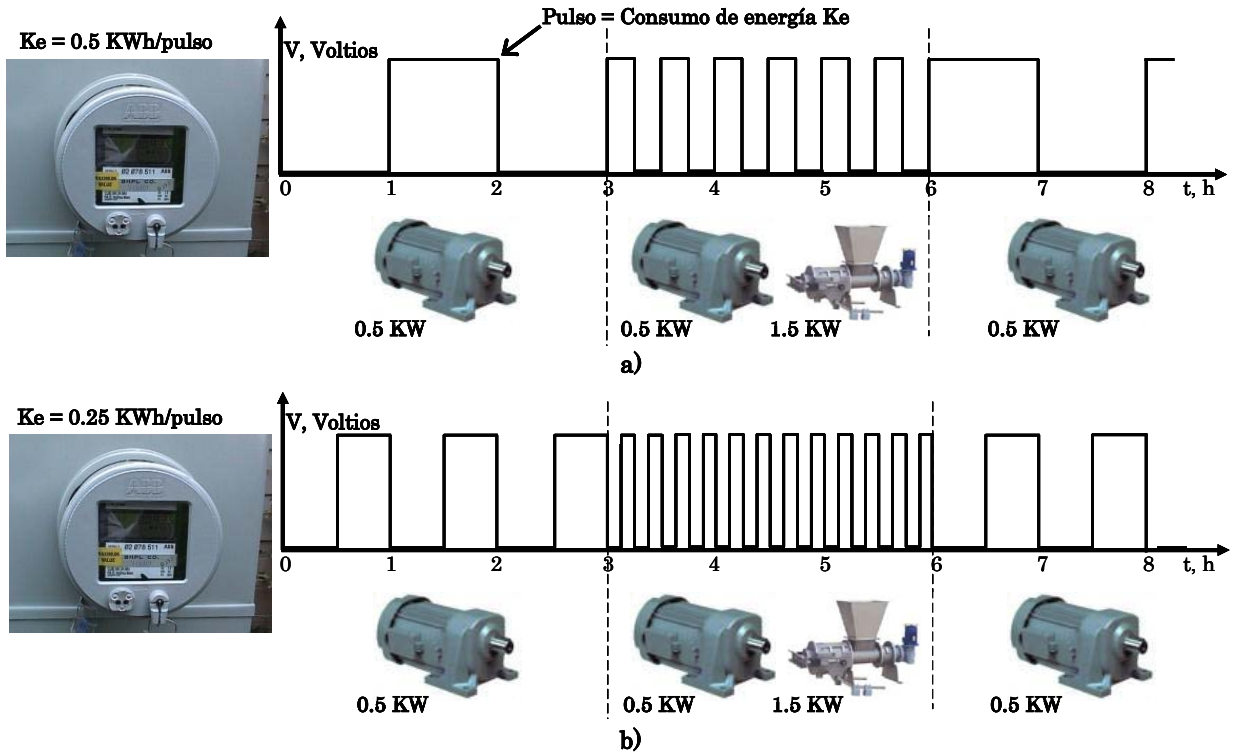


Figura 1.8. Relación entre la constante de energía y los pulsos KYZ de los medidores electrónicos.

2. El protocolo de comunicaciones MODBUS

La importancia de utilizar un protocolo de comunicaciones en este trabajo se debe al hecho de que el sistema propuesto hace uso de un microcontrolador y una computadora personal que intercambian información y porque puede ser incluido en una RCI [1]. En este caso en particular, el sistema emplea el protocolo de comunicaciones MODBUS para reportar datos de consumo de energía eléctrica y permite tener un control automático de este parámetro en forma remota.

2.1. Generalidades

Para controlar procesos industriales e integrar cada uno de los instrumentos de campo en una RCI, es necesario tener un estándar para que puedan comunicarse entre ellos. La necesidad de establecer las reglas y procedimientos digitales mediante los cuales se pueden comunicar los elementos de una red, dio origen a los protocolos de comunicaciones. PROFIBUS, MODBUS, HART, INTERBUS, SINEC L1, POWERLOGIC y WORLDFIP, son algunos de los protocolos de comunicaciones creados por los fabricantes de dispositivos utilizados en la industria [4].

El protocolo de comunicaciones MODBUS (Modicon Bus) fue puesto en el mercado en 1979 por la marca Modicon del Grupo Schneider⁴. Inicialmente se usó para establecer una red de comunicación multipunto entre sus controladores lógicos programables (PLC, *Programmable Logic Controller*), los cuales se utilizan para controlar procesos industriales distantes. Ahora, el uso de MODBUS se ha extendido a TCP/IP sobre Ethernet, pero sobre todo se emplea en la industria como un bus de campo para *monitorear* equipos de medición [16, URL9].

Un bus de campo se constituye cuando un protocolo de comunicaciones se usa conjuntamente con especificaciones de comunicación, cableado y conexión, uso de equipo y la estructura de una red orientados a la transmisión de datos [3].

Actualmente, existe una organización mundial de usuarios particulares y de empresas de automatización y control, quienes usan el protocolo de comunicaciones

⁴ Corporativo industrial dedicado a la automatización y control de equipos de medición.

MODBUS en equipos de medición instalados en campo. MODBUS se considera un bus de campo, pues además del protocolo de comunicaciones, esta organización especifica los lineamientos que lo conforman como tal [18, URL9].

A continuación se describe el protocolo de comunicaciones MODBUS y se explican las normas que deben respetarse cuando se implementa en un dispositivo programable de control. Asimismo, se describen los estándares y lineamientos que definen el uso de MODBUS como un bus de campo para la transferencia de información entre equipos remotos de medición y control.

2.2. Protocolo MODBUS de Línea Serial

El estándar MODBUS define un protocolo de mensajes en la capa de aplicación que se ubica en el nivel 7 del modelo OSI [17]. En esta capa establece una comunicación Cliente-Servidor entre dispositivos conectados en diferentes tipos de buses o redes. Sin embargo, también establece las normas de un protocolo específico *sobre una línea serial* en el que un Maestro y varios Esclavos intercambian información. Este protocolo se denomina **MODBUS de Línea Serial** y toma lugar en el nivel 2 del modelo OSI (*tabla 2.1*), es decir, en la capa de enlace de datos [18].

Tabla 2.1. MODBUS ubicado en las capas del modelo OSI.

Capa	Modelo ISO/OSI	MODBUS
7	Aplicación	Protocolo de aplicación MODBUS
6	Presentación	Vacía
5	Sesión	Vacía
4	Transporte	Vacía
3	Red	Vacía
2	Enlace de Datos	Protocolo MODBUS de Línea Serial
1	Física	EIA/TIA-485 (o EIA/TIA-232)

El protocolo MODBUS de Línea Serial utiliza dos modos de transmisión: RTU (*Remote Transmission Unit*) y ASCII (*American Standard Code for Information Interchange*). Ambos modos establecen cómo se codifican los bits en los mensajes transmitidos sobre la línea serial, cómo se generan los paquetes de información y cómo se decodifican los datos una vez recibidos por el nodo destino.

En este trabajo se implementa el protocolo MODBUS en un microcontrolador como nodo Esclavo y en una PC como nodo Maestro mediante un programa desarrollado en LABVIEW. La comunicación se establece sobre una red con enlace serial y se utiliza la transferencia inalámbrica de datos. El modo RTU es más eficiente para la transmisión inalámbrica debido a su mayor densidad de datos en las tramas respecto al modo ASCII [18]. Por tal motivo, a lo largo de este capítulo se describen las características del protocolo MODBUS de Línea Serial bajo las especificaciones de uso en modo de transmisión RTU.

2.2.1. Descripción del protocolo

El protocolo MODBUS de Línea Serial define dos tipos de estaciones que se conectan bajo una sola configuración de red y que intercambian información mediante un ciclo solicitud-respuesta.

Las estaciones o nodos de una red MODBUS son:

- **Maestro:** es el encargado de controlar el intercambio de información en la red. Sólo existe un Maestro en una red MODBUS. Las tramas generadas por el Maestro se denominan *solicitudes*.
- **Esclavo:** funciona bajo el control de la Estación Maestra (Maestro). En una red MODBUS pueden existir más de un Esclavo⁵. Las tramas generadas por el Esclavo se denominan *respuestas*.

Los nodos de la red se conectan bajo una configuración de **bus serial compartido**. En MODBUS esto significa que sólo un Maestro y uno o varios Esclavos se conectan a una sola línea serial al mismo tiempo para intercambiar información.

Cuando el Maestro desea iniciar un intercambio de información con un Esclavo de la red, le envía una solicitud. El esclavo ejecuta la tarea indicada en la solicitud y le devuelve una respuesta. A este proceso se le llama **transacción o ciclo solicitud-respuesta MODBUS**.

Sólo el dispositivo Maestro debe iniciar una transacción MODBUS. Los dispositivos Esclavos nunca transmitirán datos sin haber recibido una solicitud del Maestro y nunca podrán comunicarse entre ellos.

Cuando el Maestro envía un mensaje de solicitud a un Esclavo, espera recibir una respuesta normal. Sin embargo, sólo uno de cuatro posibles eventos ocurre después de una solicitud del Maestro:

- Si el Esclavo recibe la solicitud sin errores de comunicación y puede atenderla, envía una respuesta normal.
- Si el Esclavo no recibe la solicitud debido a un error de comunicación, no habrá respuesta. Por lo tanto, el Maestro debe manejar un tiempo de espera máximo de respuesta a su solicitud.
- Si el Esclavo recibe la petición, pero detecta un error de comunicación (paridad o CRC), no enviará respuesta. Por lo tanto, el Maestro debe manejar un tiempo de espera máximo de respuesta a su solicitud.
- Si el Esclavo recibe la petición sin error de comunicación, pero no puede manejarla, por ejemplo, si la solicitud es leer un registro inexistente, el Esclavo devolverá una *respuesta de excepción* informando al Maestro de la naturaleza del error.

Existen dos modos mediante los cuales el Maestro envía mensajes MODBUS a un Esclavo: el modo dirigido y el modo de difusión, los cuales se explican en seguida.

2.2.1.1. Modo dirigido

Con el modo *dirigido*, el Maestro se comunica con un solo Esclavo. En este modo una transacción MODBUS consiste en 2 mensajes: uno es la *solicitud* que el Maestro envía al Esclavo para que ejecute una función específica y el otro es la *respuesta* del Esclavo después de recibir y procesar la petición, lo anterior se ilustra en la *figura 2.1*.

⁵ Hasta un máximo de 247 usando repetidores en la línea.

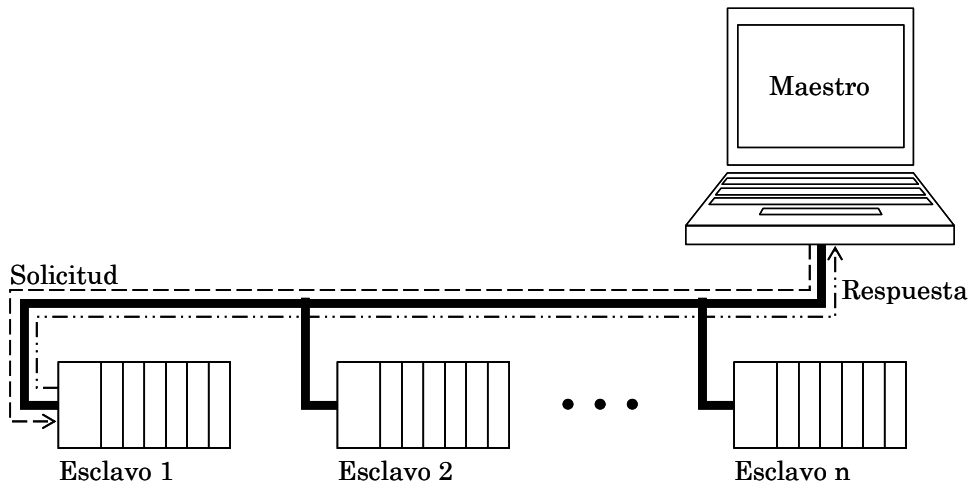


Figura 2.1. Representación del modo dirigido de los mensajes MODBUS.

Es necesario que cada Esclavo tenga una sola dirección diferente a todas las demás. Así, el Maestro puede dirigirse a un solo Esclavo de manera independiente.

2.2.1.2. Modo de difusión

Con el modo de *difusión*, el Maestro envía una petición a todos los Esclavos conectados a la red, para lo cual se utiliza la dirección 0 que se reserva para este caso. En este modo ninguno de los Esclavos responde a la solicitud del Maestro. Todos los mensajes de difusión deben ser necesariamente solicitudes de escritura y todos los Esclavos de la red deben reconocerlas y ejecutarlas. En la *figura 2.2* se muestra el concepto de un mensaje MODBUS en el modo de difusión.

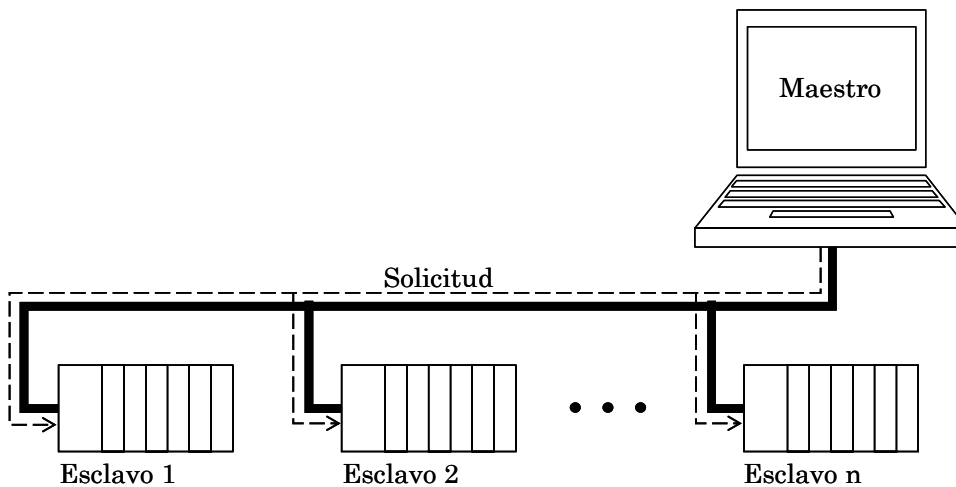


Figura 2.2. Representación del modo de difusión de los mensajes MODBUS.

2.2.2. Estructura de la trama MODBUS RTU

En la *figura 2.3* se muestra la estructura general de una trama MODBUS RTU. La trama puede ser una solicitud del Maestro en modo dirigido o de difusión, o bien, una trama de respuesta del Esclavo.

Todas las tramas constan de una unidad de datos de aplicación (ADU, *Application Data Unit*) formada por los siguientes campos:

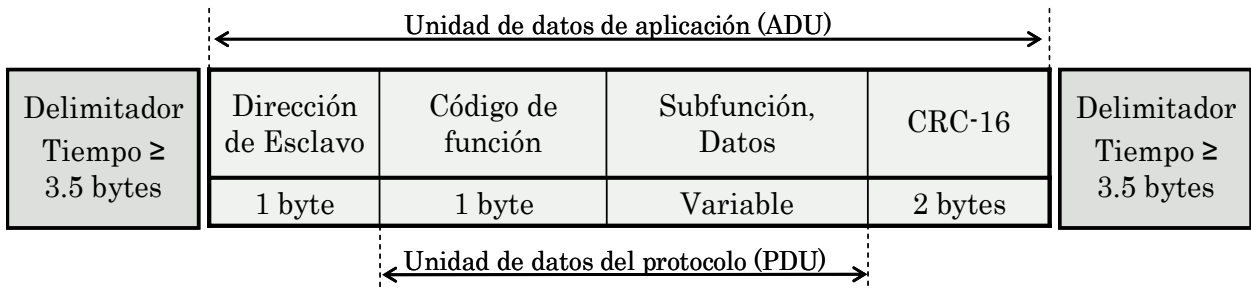


Figura 2.3. Estructura general de un mensaje MODBUS.

- *Dirección del Esclavo* con el que establece comunicación el Maestro.
- *Código de la función* que el Esclavo debe ejecutar.
- *Datos* necesarios para ejecutar la función (si los requiere).
- *Campo de comprobación de error* en la trama.

Los campos *código de función* y *datos*, en conjunto, se denominan unidad de datos del protocolo (PDU, *Protocol Data Unit*) y representa la sección de solicitud y respuesta de datos entre los dispositivos Esclavo y Maestro de MODBUS.

En el mensaje de *solicitud*, el Maestro coloca la dirección del dispositivo Esclavo con el que desea comunicarse; con un código de función le indica la acción que debe realizar; los bytes de datos contienen información adicional necesaria para ejecutar la función y un campo de comprobación de error proporciona un método para que el Esclavo valide la integridad del mensaje recibido (*figura 2.4*).

Si no hay errores, la trama de *respuesta* del Esclavo incluye su dirección y la función que ha ejecutado. Ambos campos son una réplica de la solicitud. Los datos incluyen la información que el Esclavo ha recolectado tras la ejecución de la orden indicada por el Maestro.

En caso de error, el Esclavo genera una *respuesta de excepción* que describe el error en los campos de función y de datos.

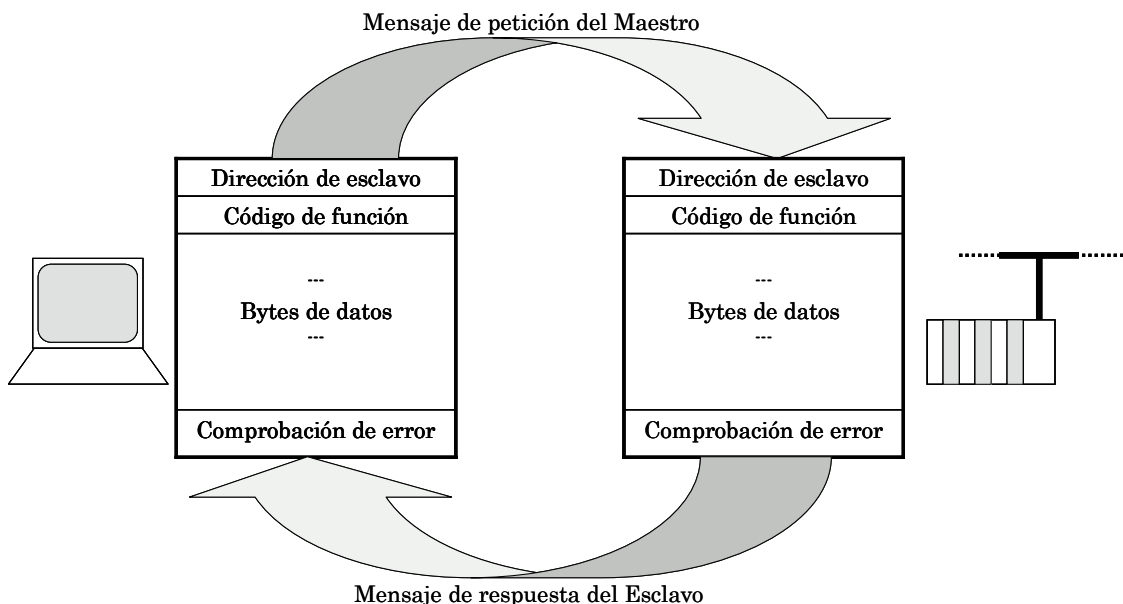


Figura 2.4. Representación gráfica de un ciclo solicitud-respuesta (transacción) MODBUS.

2.2.2.1. Delimitador de tramas

Tomando en cuenta la velocidad de transmisión (*baudrate*) utilizada, en el modo RTU las tramas son separadas por intervalos de tiempo de por lo menos 3.5 veces el tiempo de un caracter, es decir, los elementos de la red deben esperar este tiempo para iniciar la transmisión de una trama. Este tiempo representa el delimitador de las tramas MODBUS y es conocido como *tiempo de silencio*, el cual se representa como $t_{3.5}$ (figura 2.5).

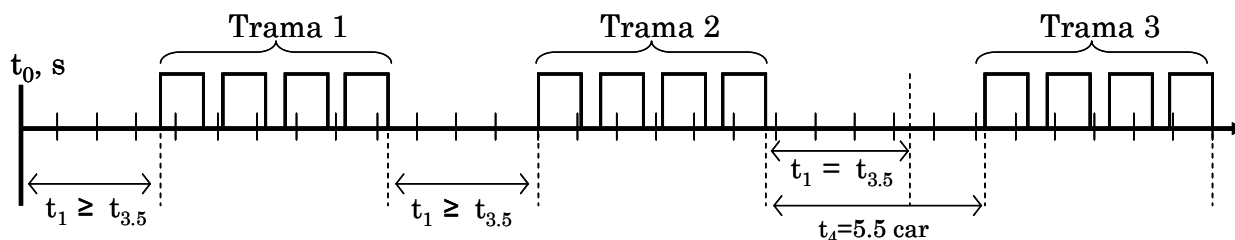


Figura 2.5. Delimitadores de tiempo entre las tramas de mensajes MODBUS.

El dispositivo transmisor coloca un mensaje MODBUS en una trama que tiene un punto de inicio y fin conocidos. La trama entera del mensaje debe ser transmitida como una cadena continua de caracteres. Si los caracteres que pertenecen a la trama están separados por un tiempo mayor o igual al *tiempo de silencio*, el receptor recibe el mensaje incompleto. Los mensajes incompletos deben detectarse y registrarse como errores de comunicación.

2.2.2.2. Campo de dirección

El campo de dirección identifica el Esclavo con el que el Maestro está intercambiando información. Tiene una longitud de 8 bits (1 byte), por lo tanto, el espacio de direccionamiento comprende 256 direcciones diferentes. La dirección 0 está reservada para solicitudes en modo de difusión. Los Esclavos utilizan una dirección en el rango de 1 a 247, las demás están reservadas para usos futuros.

En MODBUS, el Maestro no tiene una dirección específica, sólo los Esclavos poseen una dirección que es única en la red. Así, cualquier mensaje generado por un Esclavo lo recibe únicamente el Maestro y debe ser la respuesta a una petición. La tabla 2.2 muestra la organización de las direcciones en MODBUS.

Tabla 2.2. Organización de las direcciones en MODBUS.

Direcciones	Uso
0	Difusión
1-247	Dirigido
248-255	Reservadas para usos futuros

En una transacción sin errores, el campo de dirección de la solicitud es idéntico al de la respuesta.

2.2.2.3. Campo de función

En el campo de función el Maestro especifica el código de la tarea que el Esclavo va a ejecutar. La longitud es de 1 byte. Todos los códigos de función tienen el MSB en 0, por lo tanto, sus valores están por debajo de 128 (0x80).

En una transacción sin errores, el campo de función de la solicitud es idéntico al de la respuesta. En el apartado 2.2.4 se listan los códigos de función destinados a dispositivos que trabajen sobre una línea de comunicación serial.

2.2.2.4. Campo de datos

El campo datos de una solicitud, contiene información adicional que el Esclavo debe usar para tomar la acción definida por el código de función. Esto puede incluir partes como direcciones de registros, la cantidad de registros que han de ser manipulados y la cantidad de bytes de datos contenidos en el campo.

En una transacción sin errores, el campo datos de una respuesta contiene los datos solicitados. Si la función por ejecutar no requiere información adicional, el campo datos es inexistente (de longitud cero). Por ejemplo, para ejecutar el código de función 0x0B (*Devolver contador de comunicaciones exitosas*), el esclavo no requiere ninguna información adicional. El código de función por sí solo especifica la acción.

2.2.2.5. Campo de comprobación de error

Para el caso del modo de transmisión RTU, el campo de comprobación de error es el resultado de una comprobación de redundancia cíclica (CRC, Cyclical Redundancy Checking) de 16 bits sobre todos los bytes que integran el mensaje.

El valor del CRC es calculado por el dispositivo transmisor, el cual lo añade al mensaje. El dispositivo receptor del mensaje recalcula el CRC y compara el valor calculado con el valor actual recibido en el campo de CRC. Si los dos valores no son iguales, ha ocurrido un error.

2.2.2.6. Trama de las respuestas de excepción

En el caso de las respuestas de excepción, la estructura de la trama presenta ciertos cambios en los campos de función y de datos respecto a la de la figura 2.3. Estos cambios son importantes para describir el error ocurrido y se muestran en la figura 2.6.

En una respuesta de excepción, el MSB del código de función es puesto en 1. Esto hace que el código de función en una respuesta de excepción resulte 0x80 más alto de lo que sería para una respuesta normal.

Con el MSB del código de función activado, el programa de aplicación del Maestro reconoce la respuesta de excepción y después examina en el campo de datos el código de error.

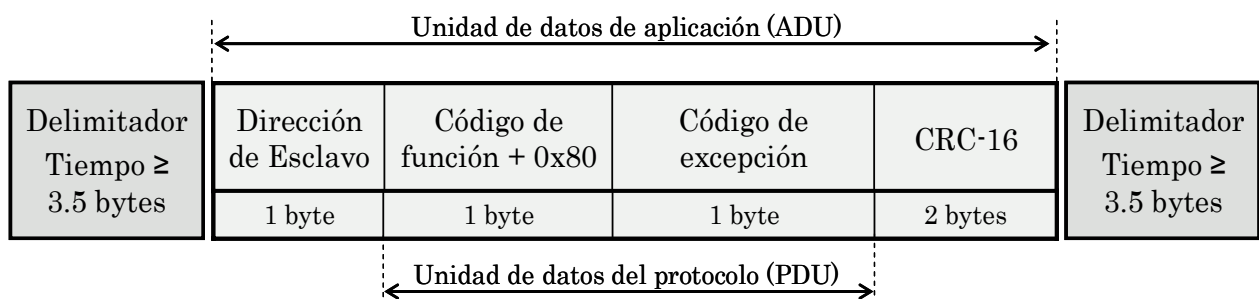


Figura 2.6. Estructura de la trama para respuestas de excepción.

En el campo de datos de una respuesta de excepción, el Esclavo retorna un código de excepción que define la condición del Esclavo que causó el error.

2.2.3. Modo de transmisión RTU

En una comunicación serial MODBUS que usa el modo RTU, cada dato de un mensaje se codifica y transmite en 11 bits de acuerdo con la distribución que se muestra en la *figura 2.7*.

Sistema binario de codificación RTU			
Inicio	Datos	Paridad	Fin
1 bit	8 bits	1 bit	1 bit

Figura 2.7. Distribución de los bits por cada dato transmitido utilizando el modo RTU.

Se requiere el uso de paridad par. Sin embargo, con la finalidad de establecer compatibilidad con otros dispositivos, es necesario soportar paridad impar y de no paridad.

Los bits de datos forman un octeto de bits (*byte*) que contiene dos dígitos hexadecimales de 4 bits, los cuales son transmitidos de izquierda a derecha: primero el bit menos significativo (LSB, *Less Significant Bit*) y por último el bit más significativo (MSB, *Most Significant Bit*).

La *figura 2.8* muestra cómo sería la secuencia de bits para el número 106 decimal (6A hexadecimal, 01101010 binario) usando el modo de transmisión RTU con distribución de 1 bit de inicio, 8 bits de datos, sin paridad y un bit de fin.

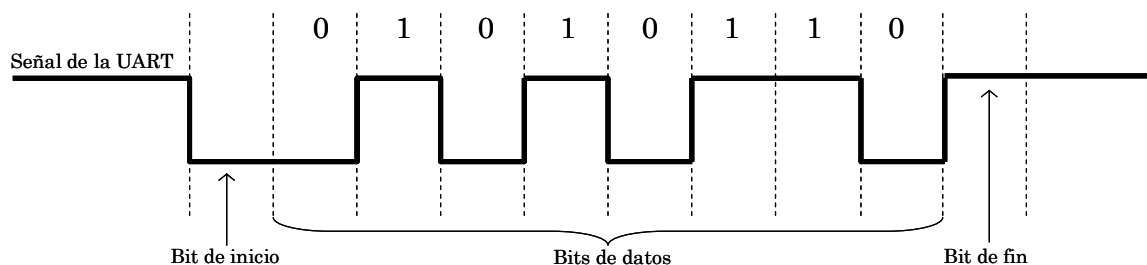


Figura 2.8. Secuencia de los bits del número 0x6A en la señal de transmisión RTU.

2.2.3.1. Generación del campo CRC en modo RTU

Para calcular el valor del campo CRC de una trama MODBUS RTU, primero se precarga un registro de 16 bits con todos ellos en 1; entonces inicia un proceso que obtiene y opera los sucesivos bytes del mensaje con el contenido del registro y actualiza éste con el resultado obtenido. Sólo los 8 bits de dato de cada carácter son utilizados para generar el CRC. Los bits de inicio, paro y paridad, no afectan al CRC.

El procedimiento propuesto por la norma del protocolo MODBUS para generar el registro CRC es:

1. Cargar un registro de 16 bits denominado registro CRC, con FFFF (todos 1).
2. XOR del primer byte del mensaje con el byte de orden bajo del registro CRC de 16 bits, colocando el resultado en el registro CRC.

3. Desplazar el registro CRC un bit a la derecha (hacia el LSB) rellenando con un cero el MSB; el LSB se extrae y examina.
4. Si el LSB es 0, repetir paso 3 (otro desplazamiento); si no (LSB es 1), hacer XOR entre el registro CRC y el valor polinómico 0xA001.
5. Repetir los pasos 3 y 4 hasta que se hayan efectuado 8 desplazamientos. Una vez hecho esto, se habrá procesado un byte completo.
6. Repetir los pasos 2 al 5 para el próximo byte del mensaje; continuar haciendo esto hasta que todos los bytes hayan sido procesados.

El contenido final del registro CRC es el valor del campo CRC de 16 bits buscado. Cuando el CRC de 16 bits (2 bytes) es transmitido en el mensaje, el byte de orden bajo se transmitirá primero, seguido por el byte de orden alto. Por ejemplo, si el valor del CRC es 0x1241, al final de la trama se insertará como 0x4112.

2.2.4. Códigos de función MODBUS

En la *figura 2.9* se muestran las tres categorías de códigos de función MODBUS en el rango de 1 a 127:

- **Códigos de función públicos:** son códigos de función definidos, validados, documentados y garantizados en ser únicos por la comunidad MODBUS.ORG [URL9]. Existen tres rangos de códigos públicos: de 1 a 64, de 73 a 109 y de 111 a 127. Dentro de estos tres rangos existen códigos aún no definidos y que están reservados para usos futuros.
- **Códigos de función definidos por el usuario:** son códigos libres para que el usuario implemente alguna función específica para su aplicación. Existen dos rangos de códigos definidos por el usuario: de 65 a 72 y de 100 a 110.
- **Códigos de función reservados:** son códigos que sólo algunas empresas utilizan en sus productos, pero no están disponibles para uso público.

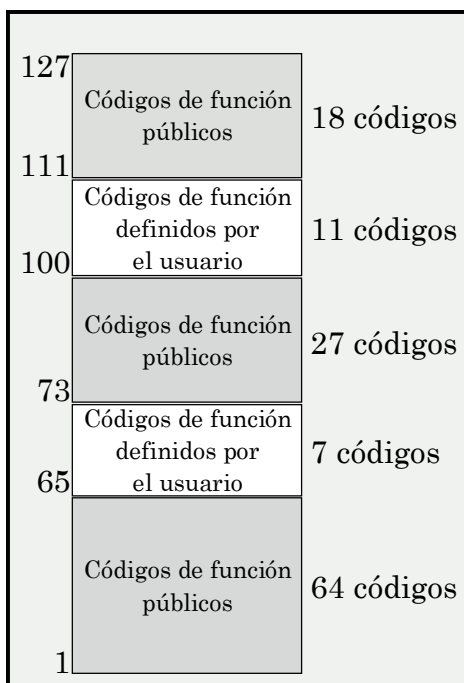


Figura 2.9. Categorías de función MODBUS.

En la *tabla 2.3* se muestran los códigos de función públicos que son usados con mayor frecuencia en dispositivos conectados a una red de enlace serial [17].

Tabla 2.3. Códigos de función MODBUS implementados en dispositivos sobre línea serial.

Código (dec)	Código (hex)	Descripción
1	0x01	Leer estado de bobinas ⁶
2	0x02	Leer entradas digitales
3	0x03	Leer registros mantenidos
4	0x04	Leer registros de entrada
5	0x05	Cambiar el estado de una bobina
6	0x06	Escribir un registro mantenido
8	0x08	Diagnósticos del dispositivo
11	0x0B	Devolver contador de comunicaciones exitosas
15	0x0F	Cambiar el estado de múltiples bobinas
16	0x10	Escribir múltiples registros mantenidos
23	0x17	Leer y escribir múltiples registros mantenidos
43	0x2B	Leer datos de identificación del dispositivo

Algunos códigos de función cuentan con subfunciones de 16 bits que indican al Esclavo qué tarea en específico debe ejecutar dentro de un conjunto de ellas. La código de subfunción elegido por el Maestro se coloca en el campo de datos de la solicitud. De la *tabla 2.3*, solamente los códigos de función 8 y 43 poseen subfunciones, las cuales se explican más adelante. A continuación se describen los códigos de función que se han implementado en el CODKYZ⁷.

2.2.4.1. Función 0x03 (Lectura de registros mantenidos)

Los *registros mantenidos* son aquellos utilizados generalmente para configurar las características de operación del dispositivo tales como velocidad de transmisión, fecha y hora, entre otros. Por lo regular estos registros son de lectura/escritura, sin embargo, en ocasiones se utilizan para almacenar datos de sólo lectura.

El código de función 3 (0x03) se usa para leer el contenido de un bloque contiguo de 1 hasta 125 *registros mantenidos* de un dispositivo Esclavo. La PDU especifica la dirección del registro de inicio y el número de registros por leer. Los registros se numeran a partir de 0, así por ejemplo, los *registros mantenidos* del 1 al 16 se numeran del 0 al 15 en la PDU.

En la respuesta, los datos de los registros son enviados en dos bytes por registro. Para cada registro, el primer byte contiene los bits de mayor orden y el segundo, los de menor orden. Ambos de izquierda a derecha.

La *tabla 2.4* muestra un ejemplo del contenido de las tramas de solicitud y respuesta para leer los *registros mantenidos* 5, 6 y 7 de un dispositivo Esclavo. Al igual que en las tablas de ejemplo subsecuentes, se debe tomar en cuenta que los registros de 16 bits del campo de datos, son separados en sus bytes alto y bajo.

⁶ Se refiere al estado encendido o apagado de salidas digitales que generalmente controlan un relevador.

⁷ Para mayor detalle sobre las otras funciones, consúltense las referencias empleadas del protocolo MODBUS [17].

El campo *contador de bytes* es de 8 bits y su valor representa la cantidad de bytes en el campo de datos de la PDU.

Tabla 2.4. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x03.

Solicitud		Respuesta	
Nombre del Campo	Valor (hex)	Nombre del Campo	Valor (hex)
Función	03	Función	03
Dirección de inicio (byte alto)	00	Contador de bytes	06
Dirección de inicio (byte bajo)	04	Valor del registro 5 (byte alto)	7F
No. de registros (byte alto)	00	Valor del registro 5 (byte bajo)	10
No. de registros (byte bajo)	03	Valor del registro 6 (byte alto)	A0
		Valor del registro 6 (byte bajo)	12
		Valor del registro 7 (byte alto)	3D
		Valor del registro 7 (byte bajo)	0C

2.2.4.2. Función 0x04 (Lectura de registros de entrada)

Los *registros de entrada* de un dispositivo Esclavo contienen información que éste obtiene del exterior, generalmente los valores de las variables físicas que mide o por contabilización de eventos externos. Estos registros son de sólo lectura.

Con la función 4 (0x04) se leen desde 1 hasta 125 *registros de entrada* contiguos de un dispositivo Esclavo. La PDU especifica la dirección del registro de inicio y el número de registros por leer. Los registros se numeran a partir de 0, así por ejemplo, los *registros mantenidos* del 1 al 16 se numeran del 0 al 15 en la PDU.

En la respuesta, los datos de los registros son enviados en dos bytes por registro. Para cada registro, el primer byte contiene los bits de mayor orden y el segundo byte, los de menor orden. Ambos de izquierda a derecha. En la *tabla 2.5* se ejemplifica una transacción MODBUS utilizando la función 0x04 para leer los *registros de entrada* 15 y 16 de un dispositivo Esclavo.

Tabla 2.5. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x04.

Solicitud		Respuesta	
Nombre del Campo	Valor (hex)	Nombre del Campo	Valor (hex)
Función	04	Función	04
Dirección de inicio (byte alto)	00	Contador de bytes	04
Dirección de inicio (byte bajo)	0E	Valor del registro 15 (byte alto)	7F
No. de registros (byte alto)	00	Valor del registro 15 (byte bajo)	10
No. de registros (byte bajo)	02	Valor del registro 16 (byte alto)	A0
		Valor del registro 16 (byte bajo)	12

2.2.4.3. Función 0x06 (Escribir un registro mantenido)

La función 6 (0x06) se utiliza para modificar el valor de un solo *registro mantenido* de un dispositivo Esclavo. En la solicitud se especifica la dirección del registro por escribirse y su valor. Para la dirección, también se considera una numeración a partir de 0. Una respuesta normal es un eco de la solicitud y se retorna una vez escrito el *registro mantenido*, tal situación se muestra en la *tabla 2.6*, la cual se ejemplifica la escritura del valor 12033 en el registro 27.

Tabla 2.6. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x06.

Solicitud		Respuesta	
Nombre del Campo	Valor (hex)	Nombre del Campo	Valor (hex)
Función	06	Función	06
Dirección del registro (byte alto)	00	Dirección del registro (byte alto)	00
Dirección del registro (byte bajo)	1A	Dirección del registro (byte bajo)	1A
Valor del registro (byte alto)	2F	Valor del registro (byte alto)	2F
Valor del registro (byte bajo)	01	Valor del registro (byte bajo)	01

2.2.4.4. Función 0x10 (Escribir múltiples registros mantenidos)

La función 16 (0x10) se utiliza para escribir un bloque contiguo de registros en un dispositivo Esclavo (desde 1 hasta 120 aproximadamente).

Los registros por escribirse se especifican en el campo de datos de la solicitud. Los datos se envían en dos bytes por registro. La respuesta normal devuelve el código de función, la dirección de inicio y la cantidad de registros escritos. En el ejemplo de la *tabla 2.7* se modifica el valor de los registros 15 y 16 de un Esclavo con los valores 1291 y 7953, respectivamente.

Tabla 2.7. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x10.

Solicitud		Respuesta	
Nombre del Campo	Valor (hex)	Nombre del Campo	Valor (hex)
Función	10	Función	10
Dirección de inicio (byte alto)	00	Dirección del registro (byte alto)	00
Dirección de inicio (byte bajo)	0E	Dirección del registro (byte bajo)	0F
No. de registros (byte alto)	00	No. de registros (byte alto)	00
No. de registros (byte bajo)	02	No. de registros (bytebajo)	02
Contador de bytes	04		
Valor de registro (byte alto)	05		
Valor de registro (byte bajo)	0B		
Valor de registro (byte alto)	1F		
Valor de registro (byte bajo)	11		

2.2.4.5. Función 0x17 (Leer y escribir múltiples registros)

La función 23 (0x17) ejecuta una combinación de una operación de lectura y una de escritura en una sola transacción MODBUS. La escritura se lleva a cabo antes de la lectura.

Los registros se numeran a partir de 0. La solicitud incluye la dirección de inicio y el número de registros por leer, así como la dirección de inicio, el número de registros y los datos por escribir. El campo contador de bytes especifica el número de bytes contenidos en el campo de datos de escritura.

Una respuesta normal a esta función contiene el grupo de registros que fueron leídos. El campo contador de bytes especifica el número de bytes contenidos en el campo de datos de lectura. En la *tabla 2.8* se muestra un ejemplo de la función 23 en el que se solicita leer los registros del 1 al 4 y escribir los valores 23602 y 4399 en los registros 15 y 16, respectivamente.

2.2.4.6. Función 0x08 (Diagnóstico del dispositivo)

La función 8 (0x08) de MODBUS es exclusiva de los dispositivos con bus serial. Proporciona una serie de pruebas para verificar el sistema de comunicación entre un dispositivo Maestro y un Esclavo o para verificar condiciones de error interno.

Tabla 2.8. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x17.

Solicitud		Respuesta	
Nombre del Campo	Valor (hex)	Nombre del Campo	Valor (hex)
Función	17	Función	17
Dir. de inicio de lectura (byte alto)	00	Contador de bytes	08
Dir. de inicio de lectura (byte bajo)	00	Valor del registro 01 (byte alto)	2A
No. de registros (byte alto)	00	Valor del registro 01 (byte bajo)	10
No. de registros (byte bajo)	04	Valor del registro 02 (byte alto)	1B
Dir. de inicio de escritura (byte alto)	00	Valor del registro 02 (byte bajo)	AA
Dir. de inicio de escritura (byte bajo)	0E	Valor del registro 03 (byte alto)	5A
No. de registros (byte alto)	00	Valor del registro 03 (byte bajo)	0C
No. de registros (byte bajo)	02	Valor del registro 04 (byte alto)	A7
Contador de bytes	04	Valor del registro 04 (byte bajo)	C2
Valor de registro (byte alto)	5C		
Valor de registro (byte bajo)	32		
Valor de registro (byte alto)	11		
Valor de registro (byte bajo)	2F		

Utiliza un campo de subfunción de dos bytes en la solicitud para definir el tipo de diagnóstico que se ejecutará. El Esclavo devuelve tanto el código de función como el de subfunción en una respuesta normal; además, algunos diagnósticos incluyen datos adicionales que se envían en el campo de datos.

En general, la función de diagnóstico no afecta la ejecución de las tareas del dispositivo, ya que los bits y registros de datos y configuración no son accedidos. Sin embargo, el Esclavo puede ser forzado a operar en el modo sólo escuchar (MSE), con el cual se *monitorean* los mensajes del bus de comunicaciones, pero no se responden. Lo anterior puede afectar al programa de aplicación en el Esclavo si depende del intercambio de información con el Maestro, pero sirve para llevar a cabo una orden de difusión si se desea que un determinado Esclavo no la ejecute.

De manera opcional, algunas subfunciones reinician los contadores de error en el Esclavo y otras devuelven los contadores de eventos de comunicación. La *tabla 2.9* contiene las subfunciones más utilizadas por los dispositivos que implementan la función de diagnóstico en el modo MODBUS RTU. A continuación se explica cada una de ellas.

2.2.4.6.1. Subfunción 0x0000 (Solicitud de eco)

Esta subfunción responde con un mensaje idéntico a la solicitud. En el campo de datos se incluye cualquier valor de 16 bits que se desee (consulte el ejemplo de la *tabla 2.10*).

Tabla 2.9. Subfunciones de diagnóstico de la función 0x08.

DEC	HEX	Descripción de la Subfunción
0	00	Eco (devolver los datos de solicitud)
1	01	Reinicio de comunicación y contadores de eventos de comunicación
2	02	Devolver el registro de diagnóstico
4	04	Activar el modo sólo escuchar (MSE)
10	0A	Reinicio de contadores de eventos de comunicación
11	0B	Devolver el registro contador de mensajes detectados en el BUS
12	0C	Devolver el registro contador de errores de comunicación (CRC)
13	0D	Devolver el registro contador de repuestas de excepción
14	0E	Devolver el registro contador de mensajes ejecutados por el Esclavo
15	0F	Devolver el registro contador de mensajes sin respuesta (de difusión)

2.2.4.6.2. Subfunción 0x0001 (Reinicio de comunicación)

Se ocupa para reiniciar el puerto de comunicaciones del dispositivo Esclavo y todos los contadores de eventos de comunicación y opcionalmente el registro cronológico de eventos de comunicación si existe. Esta subfunción es la única que elimina el MSE del dispositivo. Si el MSE está activado, reinicia la comunicación y los contadores, pero no devuelve respuesta. Si el MSE está inactivo, devuelve una respuesta normal antes de realizar el reinicio del dispositivo.

En la solicitud, el campo de datos debe ser 0x00FF si se desea que el registro cronológico de eventos sea borrado, en caso contrario debe ser 0x0000. En caso de respuesta, ésta debe ser una réplica de la solicitud.

2.2.4.6.3. Subfunción 0x0002 (Devolver el registro de diagnóstico)

Con esta subfunción se solicita el registro de diagnóstico de 16 bits si el dispositivo Esclavo posee uno. En el mensaje de petición, el campo de datos debe ser 0x0000 y en la respuesta se incluye el registro de diagnóstico.

2.2.4.6.4. Subfunción 0x0004 (Activar el MSE)

Activa el MSE para que el dispositivo detecte todos los mensajes dirigidos y de difusión, pero no se toma ninguna acción al detectarlos ni devuelve respuesta. La única función que será procesada una vez activado el MSE es la de reinicio de comunicación (Función 0x08, subfunción 0x00).

2.2.4.6.5. Subfunción 0x000A (Reinicio de contadores de eventos de comunicación)

Con esta subfunción se reinician todos los contadores y el registro de diagnóstico que posee el dispositivo. Estos registros deben reiniciarse también al encenderlo.

2.2.4.6.6. Subfunciones 0x000B a 0x000F (Devolver contador de evento)

Cuando se ejecuta cualquiera de las subfunciones 0x0B, 0x0C, 0x0D, 0x0E o 0x0F, el campo de datos del mensaje de solicitud debe ser 0x0000. En la respuesta, el campo de datos debe contener el valor actual del registro contador del evento de comunicación solicitado de acuerdo a la *tabla 2.9*.

El valor actual del contador solicitado se considera desde el último encendido, reinicio de comunicación o reinicio de contadores de eventos del dispositivo Esclavo.

La *tabla 2.10* muestra el ejemplo de una solicitud de eco (función 0x08, subfunción 0x00) al Esclavo 8 de la red MODBUS.

Tabla 2.10. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x08, 0x00.

Solicitud		Respuesta	
Nombre del Campo	Valor (hex)	Nombre del Campo	Valor (hex)
Función	08	Función	08
Sunfunción (byte alto)	00	Sunfunción (byte alto)	00
Subfunción (byte bajo)	00	Subfunción (byte bajo)	00
Dato de Solicitud (byte alto)	66	Dato de Respuesta (byte alto)	66
Dato de Solicitud (byte bajo)	AA	Dato de Respuesta (byte bajo)	AA

2.2.4.7. Función 0x0B (Devolver el contador de eventos de comunicación)

Esta función es utilizada para obtener el registro de estado y el contador de eventos de comunicación (CEC) del dispositivo Esclavo. Capturando el valor de este contador antes y después de una serie de mensajes, el Maestro puede determinar si el Esclavo ha operado los mensajes de manera normal.

El CEC es incrementado cada vez que un mensaje es completado con éxito. No se incrementa para respuestas de excepción, mensajes de eco, ni por obtener otros contadores de eventos. Se puede reiniciar por medio de la función de diagnóstico 0x08, con las subfunciones Reinicio de comunicaciones (código 0x0001) o Borrar contadores y registro de diagnóstico (código 0x000A).

La respuesta normal a esta solicitud contiene un registro de estado de dos bytes y el registro contador de eventos de comunicaciones exitosas también de dos bytes. El registro de estado será de todos unos (0xFFFF) si existe una condición de *ocupado*, es decir, si el Esclavo está ejecutando alguna tarea propia o atendiendo una solicitud previa; si no es el caso, será de todos ceros (0x0000) (*tabla 2.11*).

Tabla 2.11. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x0B.

Solicitud		Respuesta	
Nombre del Campo	Valor (hex)	Nombre del Campo	Valor (hex)
Función	0B	Función	0B
		Registro de estado (byte alto)	FF
		Registro de estado (byte bajo)	FF
		Contador de comunicaciones exitosas (byte alto)	11
		Contador de comunicaciones exitosas (byte bajo)	B7

2.2.4.8. Función 0x2B-0x0D (Leer datos de identificación)

Por regla general, los dispositivos de una red MODBUS cuentan con una interfaz encapsulada (MEI, *MODBUS Encapsulated Interface*) de transporte que define algunos tipos de consulta que se les puede realizar: el procesamiento de alguna transacción, error de manipulación de datos, entre otras definidas por cada dispositivo en particular. La más común es la lectura de datos de identificación (LDI), con ella los dispositivos reportan información adicional referente al fabricante, descripción física y funcional y la versión o revisión del dispositivo.

La consulta 14 (0x0D) de la función 43 (0x2B) de MODBUS permite al Maestro obtener esta información de un dispositivo Esclavo. La LDI es modelada

como un espacio de direcciones compuesta por un conjunto de elementos. Los elementos son llamados objetos y poseen un número de identificación.

La interfaz consiste en tres categorías de objetos que se enuncian a continuación y se detallan en la *tabla 2.12*:

- *Identificación básica del dispositivo.* La implementación de todos los objetos de esta categoría son obligatorios: Fabricante, Código del producto y número de revisión.
- *Identificación regular del dispositivo.* Estos objetos proporcionan identificación y descripción opcional del dispositivo. La implementación de los objetos definidos en esta categoría es opcional.
- *Identificación extendida del dispositivo.* Además de los objetos de identificación regular, se puede incluir objetos que proporcionen información de datos privados que son dependientes de cada dispositivo.

2.2.5. Códigos de excepción MODBUS

Los códigos de excepción son empleados por el Esclavo para indicarle al Maestro que recibió su solicitud, pero no pudo completar la función indicada por alguna razón en particular. En la *tabla 2.14* se describen los códigos de excepción que se implementan en dispositivos de enlace serial.

Tabla 2.12. Categorías de los datos de identificación de un dispositivo MODBUS.

Id. del objeto	Nombre/Descripción del objeto	Tipo	Ob/Op	Categoría
0x00	Fabricante	Trama ASCII	Obligatorio	Básica
0x01	Código del producto	Trama ASCII	Obligatorio	
0x02	Revisión	Trama ASCII	Obligatorio	
0x03	URL del fabricante	Trama ASCII	Opcional	Regular
0x04	Nombre del producto	Trama ASCII	Opcional	
0x05	Nombre del modelo	Trama ASCII	Opcional	
0x06	Nombre de la aplicación de usuario	Trama ASCII	Opcional	
0x07...0x7F	Reservado			
0x80...0xFF	Los objetos opcionales deben definirse	Definible		Extendida

2.3. MODBUS en la capa física

Para utilizar el Protocolo MODBUS de Enlace Serial en una RCI, se debe implementar una interfaz eléctrica de acuerdo con el estándar de comunicaciones EIA/TIA-485 [19]. Este estándar fue desarrollado conjuntamente por la Asociación de Industrias Electrónicas (EIA, *Electronic Industries Association*) y la Asociación de la Industria de Telecomunicaciones (TIA, *Telecommunications Industry Association*).

El RS-485 (*Recommended Standard 485*), como también se le conoce a este tipo de interfaz, permite la conexión de sistemas punto a punto y multipunto. Sin embargo, algunos dispositivos pueden implementar solamente la interfaz regida por el estándar EIA/TIA-232 para comunicaciones punto a punto.

Tabla 2.13. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x2B, 0x0D.

Solicitud		Respuesta	
Nombre del Campo	Valor (hex)	Nombre del Campo	Valor (hex)
Función	2B	Función	2B
Subfunción MEI	1E	Subfunción MEI	0E
Código de lectura del dispositivo	01	Código de lectura del dispositivo	01
Objeto por leer	00	Nivel de conformidad	01
		Siguientes	00
		Identificador del siguiente objeto	00
		Número de objetos	03
		Identificador del objeto	00
		Longitud del objeto	0C
		Valor del objeto	"UTM-FHRL-ave"
		Identificador del objeto	01
		Longitud del objeto	06
		Valor del objeto	"CODKYZ"
		Identificador del objeto	02
		Longitud del objeto	04
		Valor del objeto	"V1.0"

Tabla 2.14. Códigos de excepción implementados en dispositivos sobre línea serial.

DEC	HEX	Descripción del código
1	01	Se ha solicitado al dispositivo ejecutar una función que no soporta
2	02	Se ha intentado leer o escribir un dato con una dirección ilegal
3	03	El valor que se desea escribir en un registro no es válido
4	04	Ha ocurrido un error mientras el Esclavo ejecutaba la función solicitada
5	05	La función solicitada se está atendiendo pero tomará un tiempo considerablemente largo terminarla
6	06	El esclavo está atendiendo una tarea que le impedirá ejecutar la función solicitada

De esta manera se forma un sistema MODBUS en el que un dispositivo Maestro y uno o varios Esclavos se comunican a través de un enlace serial pasivo.

Las velocidades de transmisión requeridas son 9600 y 19200 bits por segundo (bps). Otras velocidades que pueden ser implementadas son 1200, 2400, 4800... 38400 bps, 56 y 115 Kbps, aunque existen aplicaciones con velocidades mayores.

Para la conexión de los cables se usan terminales con tornillos o conectores RJ45 o DB9.

2.3.1. MODBUS RS-485

Una red MODBUS RS-485 se forma cuando se utiliza éste estándar como interfaz eléctrica de comunicación. Bajo este esquema, se pueden construir redes de comunicación multipunto empleando una configuración "de dos hilos" (2W, *Two Wires*) o una "de cuatro hilos" (4W, *Four Wires*), las cuales se explican enseguida.

2.3.1.1. Configuración MODBUS 2W

Si la red de trabajo MODBUS RS-485 opera con la configuración 2W, todos los dispositivos se conectan en paralelo en un cable troncal constituido por tres conductores. Dos de ellos, D0 y D1, forman un par trenzado balanceado sobre los cuales se transmiten datos de manera bidireccional, esta es la razón del nombre “dos hilos”, un tercer cable (común, masa o tierra) que sirve de referencia interconecta todos los dispositivos. En la *figura 2.10* se pueden apreciar los detalles de esta configuración y en la *tabla 2.15* se listan las características de las líneas.

Bajo esta configuración, sólo un dispositivo puede transmitir datos en un sólo tiempo.

2.3.1.2. Configuración MODBUS 4W

A su vez, la configuración 4W posee un bus de dos pares 2W de datos monodireccionales, tal y como se muestra en la *figura 2.11*. Los datos en el *par del Maestro* (RXD1-RXD0) son recibidos sólo por los Esclavos y el *par de los Esclavos* (TXD1-TXD0) son recibidos sólo por el único Maestro existente en la red. Al igual que en la configuración 2W, en cualquier momento sólo un dispositivo puede transmitir datos. En la *tabla 2.16* se listan las características de las líneas de esta configuración.

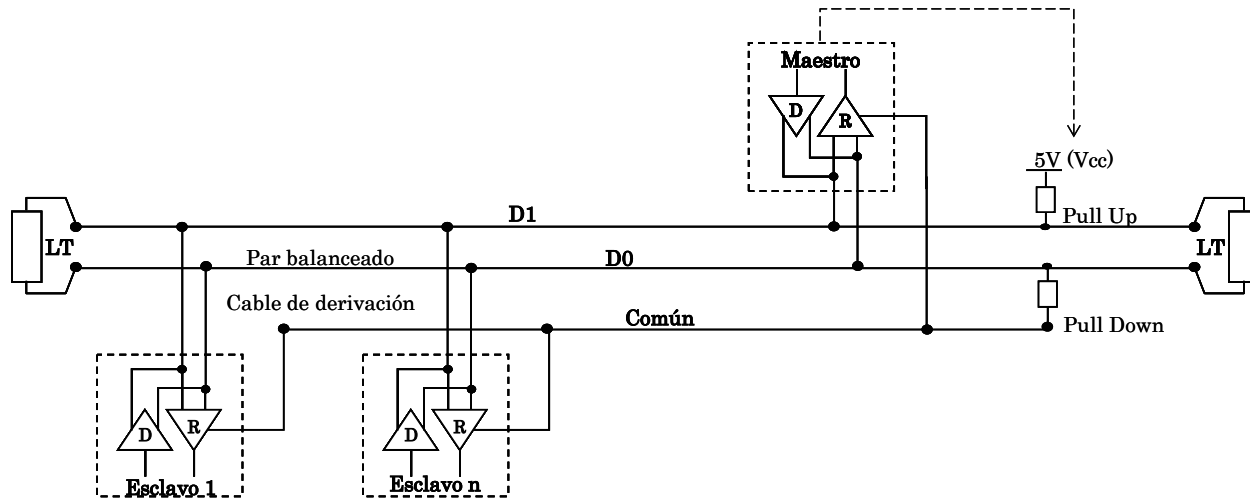


Figura 2.10. Estructura de la interconexión de dispositivos utilizando la configuración 2W.

Tabla 2.15. Definición de las líneas de la configuración 2W.

Líneas Requeridas	Tipo	Obligatoria	Nombre	Descripción
D1	E/S	Sí	B/B'	Terminal del Voltaje V1 (transceptor), donde $(V1-V0) > 200$ mV para un 1 binario o estado inactivo de la línea
D0	E/S	Sí	A/A'	Terminal del Voltaje V0 (transceptor), donde $(V0-V1) < 200$ mV para un 0 binario o estado activo de la línea
Común	REF	Sí	C/C'	Señal común de referencia

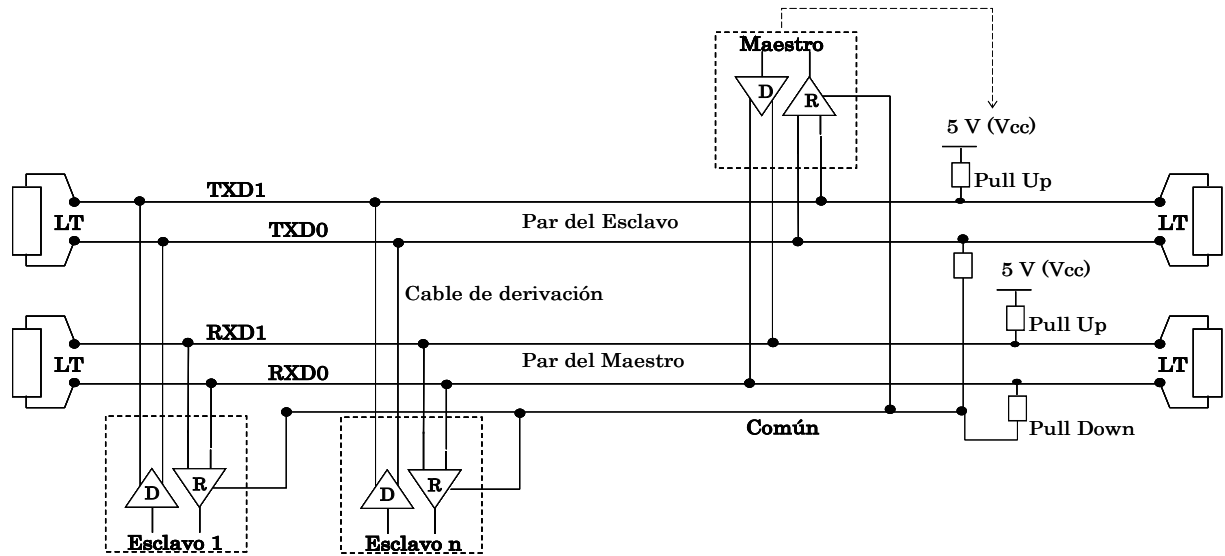


Figura 2.11. Estructura de la interconexión de dispositivos utilizando la configuración 4W.

Tabla 2.16. Definición de las líneas de la configuración 4W.

Líneas Requeridas	Tipo	Obligatoria	Nombre	Descripción
TXD1	E	Sí	B	Terminal del Voltaje Vb (transmisor), donde $(Vb-Va)>200$ mV para un 1 binario o estado inactivo de la línea
TXD0	E	Sí	A	Terminal del Voltaje Va (transmisor), donde $(Va-Vb)<200$ mV para un 0 binario o estado activo de la línea
RXD1	S	Sí	B'	Terminal del Voltaje Vb' (receptor), donde $(Vb'-Va')>200$ mV para un 1 binario o estado inactivo de la línea
RXD0	S	Sí	A'	Terminal del Voltaje Va' (receptor), donde $(Va'-Vb')<200$ mV para un 0 binario o estado activo de la línea
Común	REF	Sí	C/C'	Señal común de referencia

2.3.1.3. Requerimientos del sistema multipunto

El número máximo de dispositivos conectados en una red MODBUS RS-485, según lo establecido por la norma, es de 32 incluyendo un solo Maestro sin usar un repetidor. Lo anterior tomando en cuenta todas las posibles direcciones, la unidad de carga representada por los dispositivos y la polarización de la línea.

La configuración MODBUS RS-485 sin repetidor consta del cable troncal que interconecta todos dispositivos de la red de manera directa o utilizando cables de derivación.

La longitud máxima del cable troncal depende de la velocidad de transmisión, del cable (calibre, capacitancia o impedancia característica), del número de cargas conectadas directamente y de la configuración de la red (2W ó 4W). Por ejemplo, para una velocidad máxima de 9600 bps y un calibre 26 AWG, la longitud máxima del cable es de 1200 m. Los cables de derivación deben tener una longitud no mayor a 20 metros [18].

Cuando la línea está inactiva es susceptible al ruido externo o a interferencia. Para asegurar que la recepción permanezca en un estado constante cuando no existen señales de datos, se puede recurrir a la polarización de la línea conectando un par de resistores externos en un solo punto de todo el par balanceado RS-485:

- Un resistor de elevación (*pull-up*) se conecta de la línea D1 del par balanceado a un voltaje de 5 V.
- Un resistor de descenso (*pull-down*) se conecta de la línea D0 del par balanceado, a línea de referencia común.

El punto de conexión se elige generalmente en donde está conectado el Maestro y el voltaje de 5 V se toma de la alimentación del Maestro si su fuente de poder lo soporta, en caso contrario se emplea una fuente de poder externa.

Los valores de los resistores deben ser de 450 a 650 ohmios. El resistor de 650 ohmios puede permitir un número mayor de dispositivos conectados en la línea. Sin embargo, cuando se recurre a la polarización, el número máximo recomendable de dispositivos conectados a la red se reduce a 8.

En las *figuras 2.10 y 2.11* se ilustra la necesidad de conectar una *terminación de línea* (LT, *Line Termination*) de 150 ohmios en los extremos del cable troncal para minimizar reflexiones en la línea por desacoplo de impedancias. Cuando se polariza la línea, se recomienda usar un resistor de 120 ohmios (0.25 W) y un capacitor de 1 nF (10 V mínimo) en serie.

2.3.2. MODBUS RS-232

Una red de comunicaciones MODBUS RS-232 se implementa con dos dispositivos: un Maestro y un Esclavo, es decir, un Equipo Terminal de Datos (DTE, *Data Terminal Equipment*) y un Equipo de Comunicaciones de Datos (DCE, *Data Communication Equipment*) que se definen en el estándar EIA/TIA-232 como elementos de esta interfaz eléctrica.

La aplicación más común es aquella que utiliza una PC como Maestro (DTE) y algún sistema de tarea específica con base en un microprocesador o microcontrolador que opera como Esclavo (DCE), por ejemplo un PLC (*figura 2.12*).

La interconexión de la red es punto a punto y la longitud del cable debe ser menor que 20m. En la *tabla 2.17* se listan las señales que intervienen en este tipo de comunicaciones y que define el estándar EIA/TIA-232.

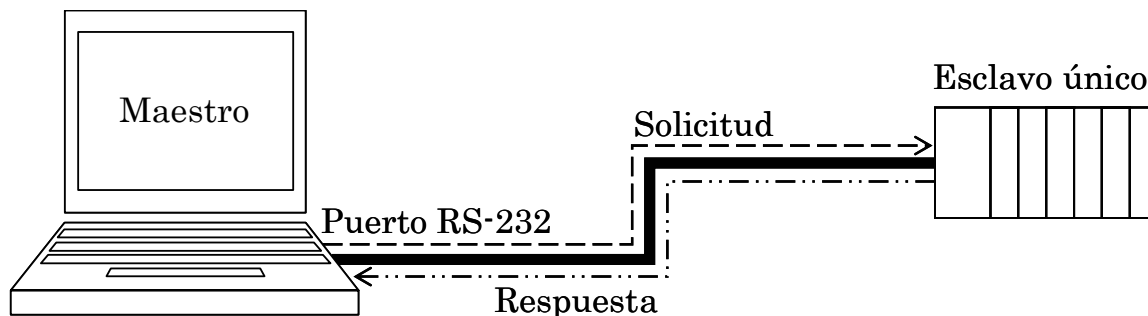


Figura 2.12. Configuración punto a punto MODBUS RS-232.

Tabla 2.17. Señales definidas por el estándar EIA/TIA-232 para una comunicación punto a punto.

Líneas Requeridas	Tipo	Obligatoria	Nombre	Descripción
COM	REF	Sí	Common	Señal común de referencia
CTS	E	No	Clear to Send	Señal de inicialización para enviar
DCD	S	No	Data Carrier Detect	Portadora de datos detectada
DSR	E	No	Data Set Ready	Indica que está listo para ser utilizado
DTR	S	No	Data Terminal Ready	Terminal de datos lista para enviar
RTS	S	No	Request to Send	Interrupción de transmisión
RXD	E	Sí	Received Data	Línea para recepción de datos
TXD	S	SÍ	Transmitted Data	Línea para transmisión de datos

3. Diseño y desarrollo de un codificador de pulsos KYZ

Para presentar una mejor descripción del sistema, primero se explica el diseño del hardware y después, el diseño del firmware⁸, es decir, primero se exponen las características de los circuitos y componentes electrónicos empleados para el acondicionamiento de las señales de entrada y salidas del sistema y después, se exponen los detalles considerados para implementar las funciones más importantes del programa del microcontrolador AT90S2313 [20], lo cual se llevó a cabo en el lenguaje ensamblador de Atmel (AVR-Assembler) [URL10].

3.1. Descripción general del sistema

La *figura 3.1* muestra el diagrama a bloques del CODKYZ, cuyas entradas principales son las conmutaciones de los relevadores KYZ de los medidores electrónicos de energía y las tramas de solicitud MODBUS. El hardware del CODKYZ se divide en cuatro módulos principales:

- Módulo de alimentación.
- Módulo de acoplamiento.
- Módulo de comunicaciones.
- Módulo digital.

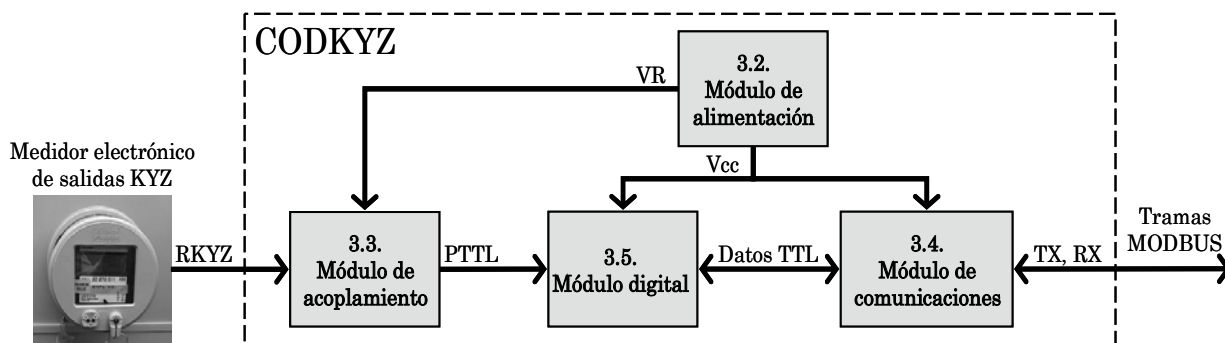


Figura 3.1. Diagrama a bloques del hardware del CODKYZ.

⁸ Software empotrado en un dispositivo programable que forma parte del hardware de un sistema electrónico.

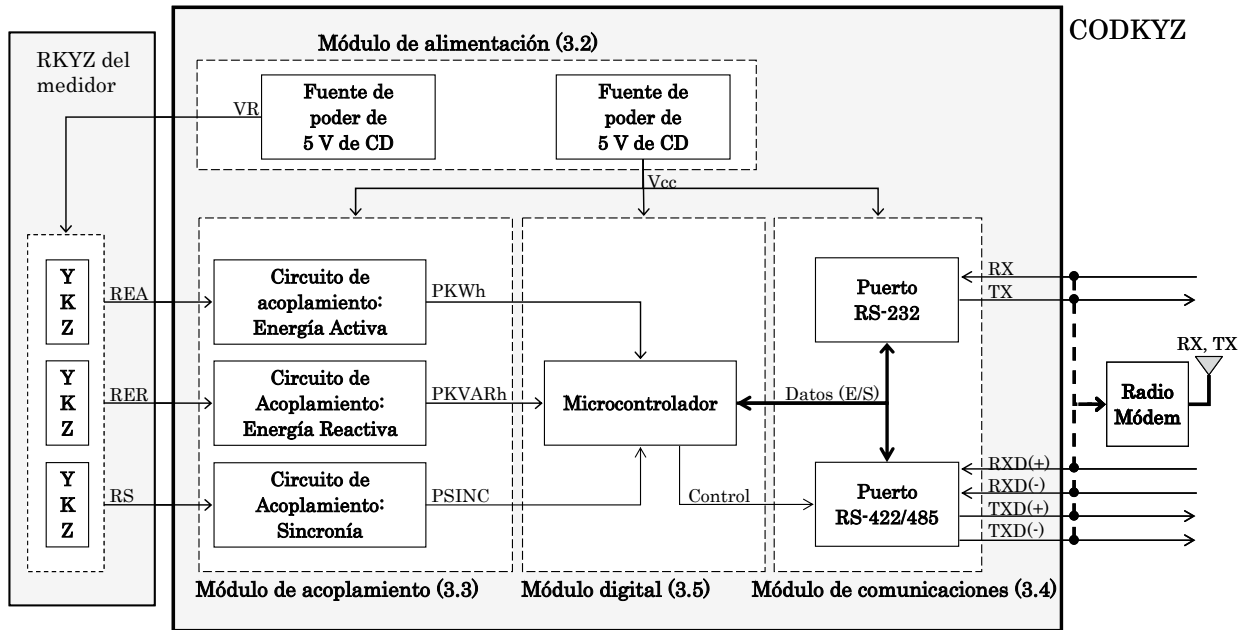


Figura 3.2. Señales de entrada y salida de los bloques que integran el hardware del CODKYZ.

En el diagrama de la figura 3.2 se muestran las señales de entrada y salida de cada módulo.

El *módulo de alimentación* posee dos fuentes de voltaje de CD que se emplean para energizar al sistema. Como puede observarse en la figura 3.2, ambas fuentes son de 5 V, una (VR) se conecta a los relevadores del medidor para producir los pulsos KYZ que recibe la etapa de acoplamiento y la otra (Vcc) alimenta a todos los componentes del CODKYZ.

Se requiere de dos fuentes de voltaje independientes, cada una con su respectivo transformador, para garantizar un aislamiento eléctrico entre la alimentación de corriente alterna (tomada de la subestación eléctrica), los relevadores de conmutación de los medidores electrónicos y la alimentación del CODKYZ. Esto ayuda a reducir los cambios de voltaje que pueden ocurrir en la alimentación de CD del sistema, ya que en el ambiente de operación existe intenso ruido electromagnético.

El *módulo de acoplamiento* consta de los circuitos necesarios para generar los pulsos KYZ a partir de un voltaje inicial y la conmutación de los relevadores de estado sólido de los medidores electrónicos. En sus salidas se producen niveles de voltaje de tipo TTL (*Transistor-Transistor Logic*) que son los adecuados para que el microcontrolador interprete los pulsos KYZ de energía activa (PKWh), energía reactiva (PKARh) y de sincronía (PSINC).

El *módulo de comunicaciones* lo constituyen los puertos de comunicaciones seriales del sistema, los cuales reciben las tramas de datos en niveles de voltaje TTL a través de la UART (*Universal Asynchronous Receiver and Transmitter*) del microcontrolador y los circuitos integrados (CI) de este módulo convierten los niveles de voltaje TTL a niveles especificados por las normas EIA/TIA-232 y EIA/TIA-485 para comunicaciones sobre una línea serial. También, como parte del módulo de comunicaciones, se incluye el radio-módem de la firma MaxStream para enviar datos de forma inalámbrica.

El *módulo digital* está conformado en su totalidad por el microcontrolador AT90S2313, el cual se encarga de: obtener, cuantificar y codificar los pulsos KYZ provenientes del módulo de acoplamiento. La codificación y transferencia de los datos se realiza siguiendo las reglas especificadas por el protocolo MODBUS, por esta razón, el microcontrolador posee un programa que realiza todas las funciones MODBUS necesarias para que el CODKYZ funcione adecuadamente y transfiera información por medio del módulo de comunicaciones.

Enseguida se explican con mayor detalle las características de todos los módulos citados anteriormente.

3.2. Módulo de alimentación

En la *figura 3.3* se ilustra el diagrama esquemático de las dos fuentes de voltaje de CD que alimentan el hardware del sistema. El diseño corresponde al de una fuente fija de voltaje regulado a través del CI lineal LM7805 [21] y con aislamiento de CA por medio de un transformador lineal [22]. Debido a que el hardware fue diseñado para que la exigencia de corriente fuera poca (refiérase al manual de usuario en el apéndice B), los transformadores que se ocupan en las dos fuentes son de 12 Vrms con 120 miliamperios (mA) de salida. Ambas fuentes proporcionan +5 V a su salida.

3.3. Módulo de acoplamiento

Para producir los PKWh, PKVARh y PSINC a partir de las conmutaciones de los relevadores del medidor de energía, se utilizan tres circuitos como el que se muestra en la *figura 3.4*, donde se observa que solamente se emplea el par de terminales K-Y del relevador KYZ, es decir, se ocupa la configuración de *iniciador de dos hilos* explicada en el *inciso 1.4.1.1*. Para ello se envía un voltaje de 5 V de CD hacia la terminal Y del relevador y su retorno se conecta de K hacia una resistencia de 470 Ω que limita el flujo de corriente en el LED (*Light Emitter Diode*) del optoacoplador 4N25 de la firma *Fairchild Semiconductors* [23], con el cual se consigue el aislamiento eléctrico entre las señales pulsadas del medidor y el módulo digital del sistema que trabaja con niveles TTL.

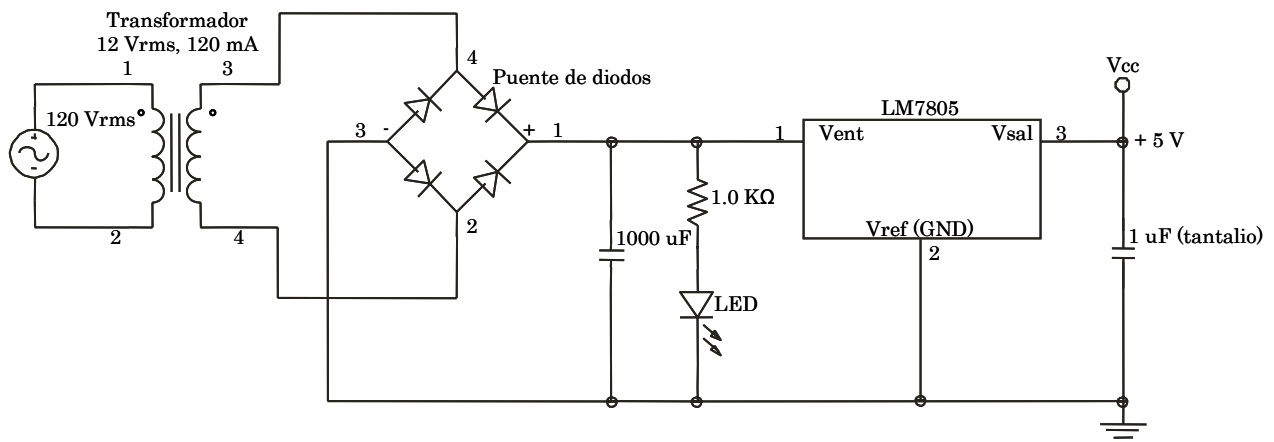


Figura 3.3. Diagrama esquemático de la fuente fija de voltaje de CD que alimenta el CODKYZ.

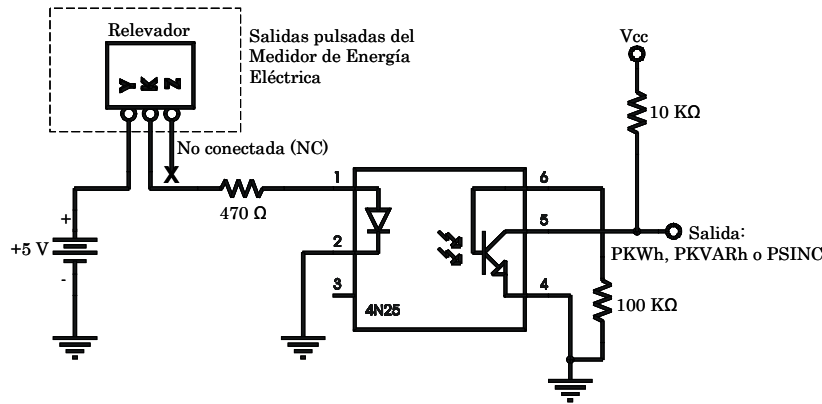


Figura 3.4. Diagrama esquemático de la etapa de acoplamiento.

Cuando las terminales K-Y se encuentran cerradas, se produce un flujo de corriente a través del optoacoplador que provoca la saturación de su transistor y a su salida se tiene un voltaje de 0 V (‘0’ lógico). Por otra parte, cuando el par K-Y está abierto, no hay flujo de corriente y el optotransistor se encuentra en estado de corte [24], por lo que a la salida se tienen 5 V (‘1’, lógico). Con la implementación de este arreglo, en la salida se generan señales de voltaje con niveles TTL que el microcontrolador usa para la lectura, cuantificación y codificación de los pulsos KYZ.

3.4. Módulo de comunicaciones

En esta etapa se establece el intercambio de datos entre el CODKYZ, como dispositivo Esclavo de la red y un Maestro que se encarga de consultarlo. La comunicación alamburada se lleva a cabo por medio de los puertos que posee el sistema bajo los estándares RS-232 y RS-485 de 2 ó 4 hilos. El radio-módem es el elemento que se usa para establecer el enlace inalámbrico.

3.4.1. Puertos de comunicación alamburada

Los puertos de comunicación alamburada se constituyen por los convertidores de niveles TTL de voltaje a niveles propios de los estándares de comunicación y los conectores que se ocupan para cada interfaz eléctrica, ya sea RS-232 o RS-485. Como se observa en la figura 3.5, el módulo de comunicaciones tiene señales de entrada y salida en niveles TTL que se conectan directamente a la UART del microcontrolador que constituye el módulo digital.

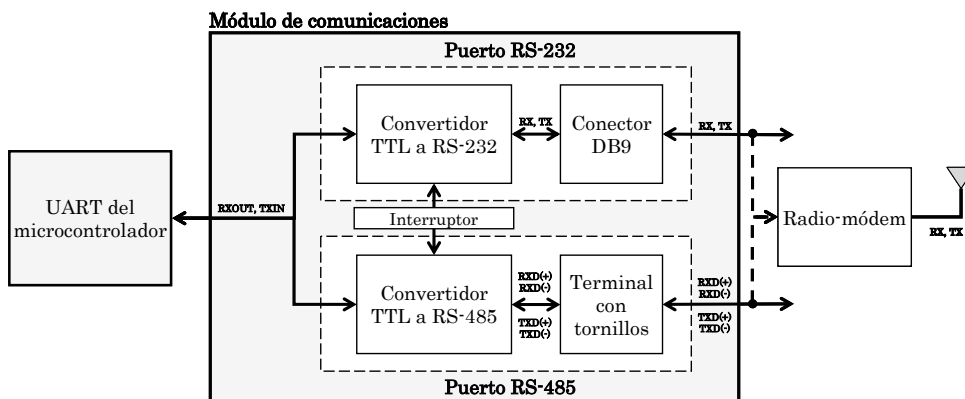


Figura 3.5. Diagrama a bloques de los puertos de comunicaciones seriales del CODKYZ.

3.4.1.1. Puerto RS-232

Para hacer la conversión de los niveles TTL de salida de la UART a niveles de voltaje RS-232 y viceversa, se emplea el CI DS275 de *DALLAS Semiconductor (MAXIM)*, cuyo diagrama eléctrico de conexiones se presenta en la *figura 3.6* [25]. Este componente se alimenta con 5 V de CD.

Como la cantidad de información que el CODKYZ intercambia en la comunicación serial es poca⁹, únicamente requiere de las señales de Recepción (RX) y Transmisión (TX) de datos. Además, el modo de comunicación del CODKYZ es semidúplex¹⁰ (o half-dúplex), es decir, puede transmitir y recibir datos, pero sólo una acción a la vez.

El DS275 está diseñado para soportar preferentemente este tipo de comunicación, para ello posee un interruptor que habilita automáticamente sólo una de las señales de intercambio de datos a la vez, RX o TX, la que inicie primero. La señal VDRV debe estar en alto (5V) para cumplir con esta función (*figura 3.6*).

Como interfaz de salida de este puerto se utiliza el conector DB9 hembra de 9 terminales (*pin*s) que se presenta en la *figura 3.7* y del cual se emplean sólo los *pin*s 2 (Tx), 3 (Rx) y 5 (GND) para las señales obligatorias que establece el estándar MODBUS RS-232 (*tabla 2.17*).

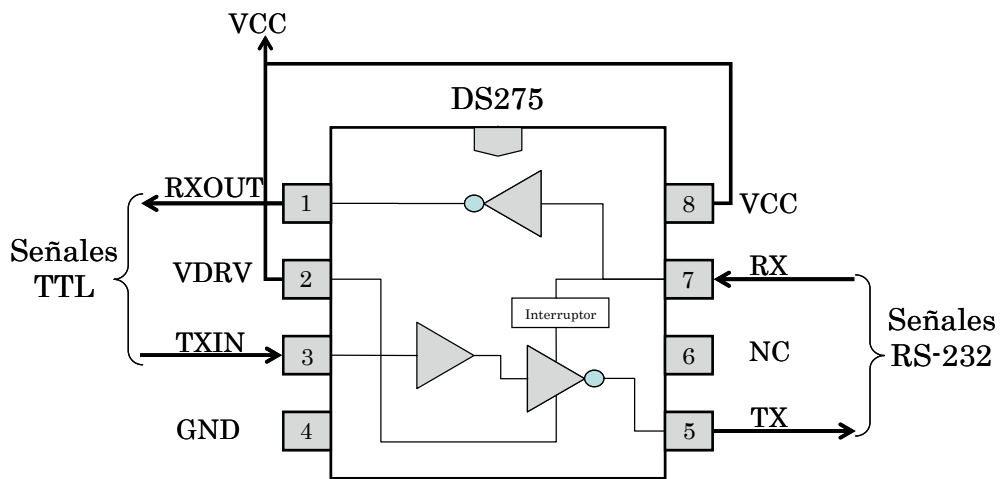


Figura 3.6. Diagrama de conexiones del CI DS275 [25].

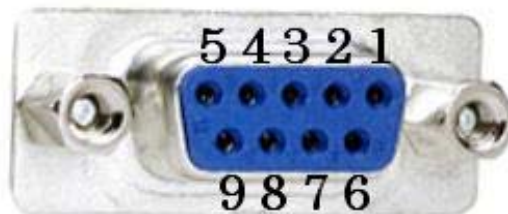


Figura 3.7. Conector DB9 hembra de 9 pines empleado para la salida del puerto RS-232.

⁹ Menos de 256 caracteres en tramas continuas a 9600 baudios, para cantidades mayores de datos se recomienda emplear señales de control de flujo de datos [7].

¹⁰ El término dúplex (o full-dúplex) se emplea para indicar una comunicación bidireccional en la que se puede transmitir y recibir datos de forma simultánea.

3.4.1.2. Puerto RS-485

Para adaptar los niveles TTL del microcontrolador y generar las señales del puerto RS-485 de 2 ó 4 hilos siguiendo la norma EIA/TIA-485, se usa el CI MAX489 también de la firma MAXIM [26]. En la *figura 3.8* se muestra su diagrama de conexiones y en la *tabla 3.1*, la descripción de sus señales eléctricas.

Cuando el MAX489 opera con 4 hilos, se usan el par de señales A-B para recepción y Y-Z para transmisión de manera independiente. Las señales de habilitación RE y DE pueden activarse a la vez para establecer una comunicación dúplex; sin embargo, debido a que la topología Maestro Esclavo de MODBUS requiere una comunicación semidúplex, estas señales se interconectan y son manipuladas por una señal de *control* que habilita la recepción o transmisión sólo una a la vez por ser excluyentes (activos en bajo y alto, respectivamente).

Cuando el transmisor o el receptor se encuentran deshabilitados, sus salidas y entradas se ponen en alta impedancia; esta característica se aprovecha interconectando las señales A-Y y B-Z para construir el puerto RS-485 de 2 hilos. De esta manera, ya sea con 2 ó 4 hilos, cuando el CODKYZ espera datos, habilita el receptor del CI MAX489 activando la señal RE (en bajo) y desactiva la señal DE. Algo muy parecido sucede cuando va a transmitir activando la señal DE (en alto). Todo lo anterior se ilustra en la *figura 3.8*.

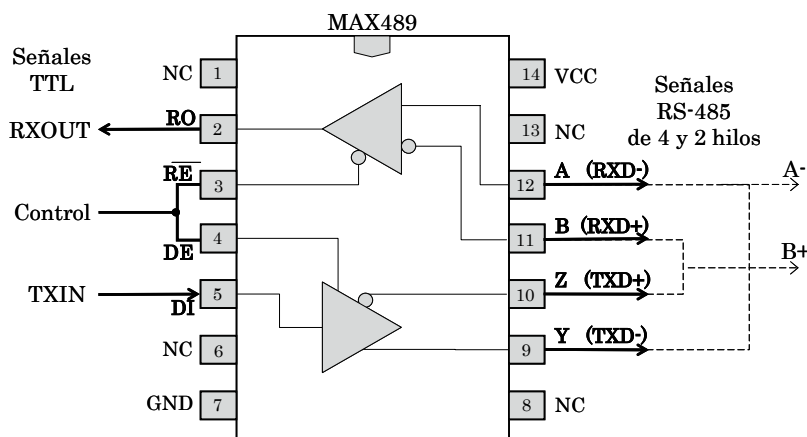


Figura 3.8. Diagrama de conexiones del CI MAX489 [26].

Tabla 3.1. Señales eléctricas del CI MAX489 de MAXIM [26].

No. de pin	Nombre	Descripción	Tipo
1, 8, 13	NC	Sin conexión	-
2	RO	Recepción de datos (TTL)	Salida
3	RE	Señal de habilitación del receptor (activo en bajo)	Entrada
4	DE	Señal de habilitación del transmisor (activo en alto)	Entrada
5	DI	Transmisión de datos (TTL)	Entrada
6, 7	GND	Señal de referencia común	-
9	Y	Señal no invertida para transmisión RS-485	Salida
10	Z	Señal invertida para transmisión RS-485	Salida
11	A	Señal no invertida para recepción RS-485	Entrada
12	B	Señal invertida para recepción RS-485	Entrada
14	VCC	Alimentación del CI	-



Figura 3.9. Terminales con tornillos empleadas para conexión del puerto RS-485.

En la *figura 3.9* se muestran los conectores TRT-02 empleados como salida del puerto RS-485, los cuales poseen terminales con tornillos para sujetar los cables que portan las señales de intercambio de datos: A (-), B (+) y C (común) definidas en el estándar RS-485 para comunicaciones sobre un bus serial de 2 hilos y TXD+, TXD-, RXD+ y RXD- para 4 hilos (refiérase a las *tablas 2.15* y *2.16*).

3.4.2. Radio-módem MaxStream de 900 MHz

El radio-módem constituye el puerto de comunicaciones inalámbricas y puede conectarse al CODKYZ por cualquiera de los puertos de comunicación alamburada. Permite transferir la información almacenada en el CODKYZ en forma inalámbrica. El dispositivo utilizado es de la firma MaxStream modelo 9XStream que opera con un baudrate de 9600 bps y ocupa una frecuencia portadora de 900MHz para el enlace inalámbrico.

De acuerdo con el diagrama de la *figura 3.10*, conceptualmente este dispositivo se divide en dos módulos:

- La tarjeta de interfaz: recibe los datos de comunicación con niveles de voltaje RS-232 y RS-485 a través de un conector DB9 hembra y los convierte a niveles TTL de voltaje que son los que ocupa el siguiente módulo (*figura 3.11a*).
- El radio-módem: recibe los datos de la tarjeta de interfaz y los modula empleando la técnica de espectro expandido (*spread spectrum*) [URL11].

Por omisión, el baudrate de interfaz serial es 9600 o 19200 baudios, pero se puede cambiar empleando comandos AT de módem (*AT commands*) [27]. Además como se observa en la *figura 3.11b*, este modelo posee un conector SMA para adaptarle una antena externa (*figura 3.11c*).

La antena que se conectó al radio-módem es una antena tipo Yagui modelo PC906N de la firma Cushcraft.

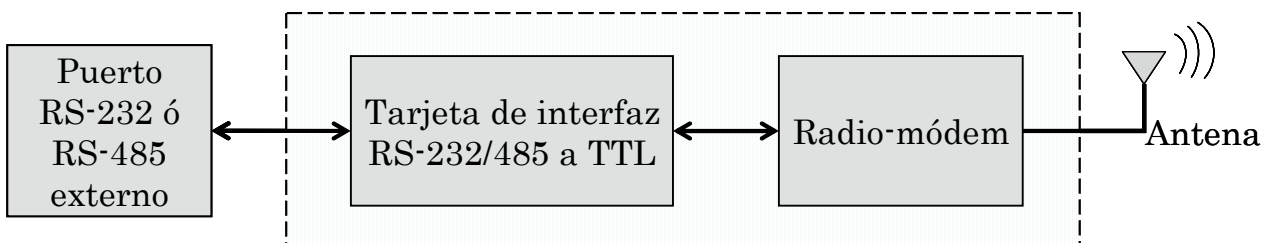


Figura 3.10. Diagrama a bloques del radio-módem MaxStream.

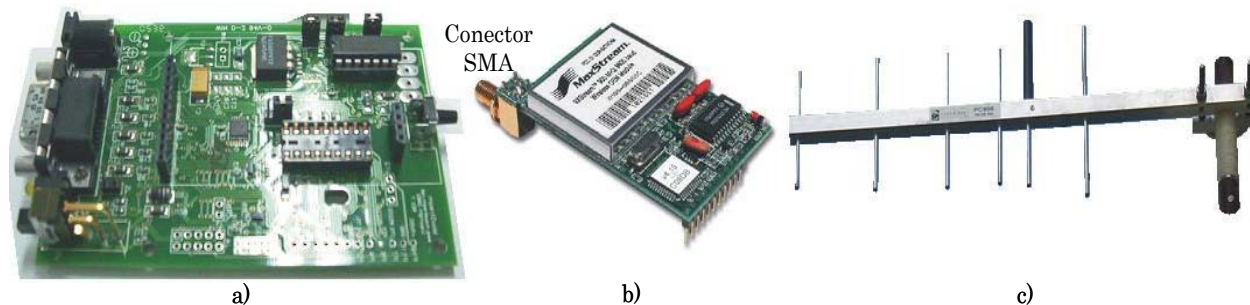


Figura 3.11. Módulo 9Xstream: a) tarjeta de interfaz, b) radio-módem, c) antena externa.

En las *tablas 3.2 y 3.3* se presentan las características principales del radio-módem y de la antena empleados para las pruebas de enlace inalámbrico del CODKYZ.

Tabla 3.2. Características principales de operación del radio-módem modelo 9XStream [27].

Característica	Tipo/Valor	Observaciones
Interfaz eléctrica	RS-232, RS-422/485	La configuración es manual a través de puentes.
Baudrate	2400, 4800, 9600, 19200, 38400, 57600 bps	La configuración se realiza a través de comandos AT.
Modulación	Espectro expandido	Ocupa la técnica de frecuencia escalonada [URL11].
Frecuencia de portadora	900 MHz	
Frecuencia modulante	9600 baudios	La razón de datos en la transmisión inalámbrica siempre es la misma sin importar el baudrate de operación en la interfaz serial.
Conector para antena	SMA	

Tabla 3.3. Características principales de la antena ocupada para enlace inalámbrico.

Característica	Tipo/Valor
Topología	Yagui
Número de elementos	6
Conector	SMA
Frecuencia de operación	896-940 MHz
Ganancia	6 dBd ¹¹
Ancho de haz a 3 dB, Plano E	55°
Ancho de haz a 3 dB, Plano H	65°
Longitud	62.9 cm
Peso	720 g
Resistencia al viento	200 Km/h

¹¹ Esta unidad se ocupa para hacer referencia a un dipolo de media onda.

3.5. Módulo digital

Todas las funciones y tareas que efectúa el microcontrolador AT90S2313 de ATMEL, conforman el módulo digital de este proyecto. En la *figura 3.12* se presenta el esquema a bloques de los datos de entrada, el procesamiento y los datos de salida del programa implementado, en ella pueden observarse también algunos de los recursos principales del microcontrolador que se emplean para realizar tales funciones.

Los datos que constituyen las entradas de este sistema son:

- Las señales pulsadas PKWh, PKVARh y PSINC de nivel TTL: provienen de la salida de la etapa de acoplamiento y representan los pulsos generados por la conmutación de los relevadores KYZ. Cuando estos datos de entrada son detectados por el microcontrolador, éste realiza una tarea específica para atenderlos.
- Las tramas de solicitud MODBUS: son enviadas por el dispositivo Maestro. Llegan al microcontrolador en niveles TTL y éste se encarga de procesarlas para ejecutar una función y generar una respuesta.

El procesamiento de los datos contiene tres tareas principales:

- Cuantificar los pulsos KYZ de entrada: realiza un conteo de los pulsos que ocurren y mide el tiempo transcurrido entre cada pulso.
- Almacenar temporalmente los datos en la RAM (*Random Access Memory*): hasta que se reciba un mensaje de petición que los solicite.
- Incluir los datos en las tramas de respuesta: de acuerdo a la codificación del protocolo de comunicaciones MODBUS RTU.

Las salidas del sistema son:

- La señal de control para el convertidor de nivel MAX489.
- Las tramas MODBUS que se generan como respuesta a los mensajes de solicitud enviados por un dispositivo Maestro.

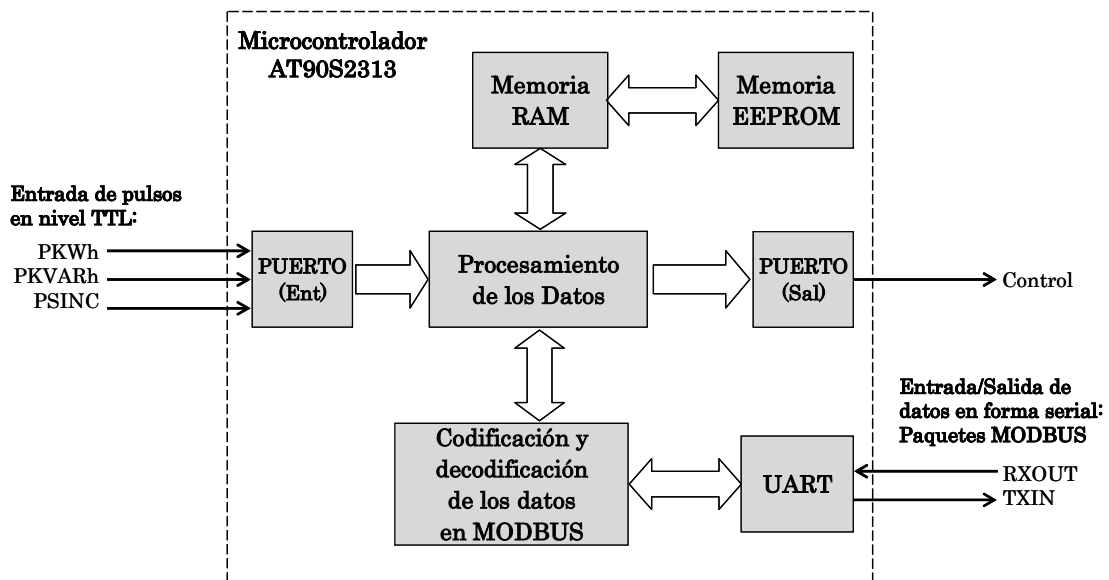


Figura 3.12. Diagrama funcional de la etapa digital implementada en el AT90S2313.

En las tramas se incluyen principalmente los datos codificados sobre la cuantificación de los pulsos KYZ, los datos de configuración o los datos de diagnóstico del dispositivo.

Para realizar todas las tareas que abarca la etapa digital, se ocupó la mayor parte de los recursos que posee el microcontrolador AT90S2313 (*apéndice A*).

Además de las descritas anteriormente, existen otras tareas que son necesarias para complementar la etapa digital; entre las más importantes están: el manejo de la memoria EEPROM (*Electrically Erasable Programmable Read Only Memory*) para almacenar datos de manera permanente, la implementación de un Reloj/Calendario para incluir información del tiempo en los datos que se reportan al nodo Maestro y la implementación del protocolo de comunicaciones MODBUS de acuerdo a lo que establece su norma (*capítulo 2*) [17]. A continuación se describe de manera detallada la implementación de las tareas que ejecuta el microcontrolador y los recursos que se ocupan para tal fin.

3.5.1. Detección de los pulsos KYZ

De acuerdo a lo descrito en el *inciso 1.4.1.1*, cuando se utiliza la configuración de 2 hilos, los pulsos KYZ se deben reconocer como un cambio de estado de los relevadores del medidor, ya sea de abierto a cerrado o de cerrado a abierto. En el caso del microcontrolador, esto representa cualquiera de los dos cambios de estado de los *pin*s de entrada, de alto a bajo o de bajo a alto. Sin embargo, lo importante es saber que el relevador KYZ ha conmutado.

Las especificaciones de los relevadores KYZ de estado sólido indican que no generan rebotes; sin embargo, éstos se previenen agregando un tiempo de retardo de 50 ms que, en la práctica, se encontró suficiente para evitar que se acepte un pulso como válido cuando en realidad son transiciones indeseables del relevador [10].

Los pulsos con nivel TTL de voltaje que llegan al microcontrolador, se *detectan* utilizando la técnica de *monitoreo* continuo del puerto, es decir, las entradas del puerto se exploran continuamente para verificar su estado y se emplean *banderas* que informan del último cambio presentado en cada entrada de pulsos (nivel alto o bajo).

El *procedimiento de detección* de los pulsos KYZ es muy similar para los tres casos: Al inicializar el dispositivo se revisa el estado actual del *pin* con la finalidad de configurar la *bandera de estado* y, en un ciclo repetitivo en el que se atienden diversas tareas del microcontrolador, se *monitorea* la señal de entrada del *pin*.

Cuando el valor de la bandera no coincide con el nivel actual de la señal de entrada, se espera un tiempo de retardo para verificar la validez del pulso. Si al término de este intervalo de tiempo, el nivel de la señal no coincide con la configuración de la *bandera de estado*, el pulso se acepta, en caso contrario, se descarta.

Para detectar los pulsos KYZ de entrada se utilizan tres *pin*s del puerto D del microcontrolador. La asignación de *pin*s se ilustra en la *figura 3.13*.

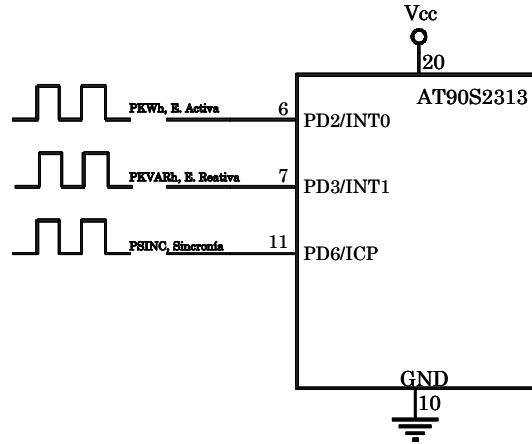


Figura 3.13. Lectura de los pulsos KYZ por medio de los pines del microcontrolador.

3.5.2. Cuantificación de los pulsos KYZ

La *cuantificación* de los pulsos KYZ se realiza después de haber verificado su validez: en un registro de 16 bits (formado por dos registros de 8 bits) se contabilizan los pulsos ocurridos y en otro registro se almacena la información del tiempo que permaneció en alto o en bajo la señal antes de originarse dicho pulso.

Lo anterior describe la forma en la cual se cuantifican los pulsos KYZ de energía activa y reactiva, cuyos registros de 16 bits pertenecen a la memoria RAM y se les denominará, de ahora en adelante, PulsosKWH y PulsosKVARH, a los que cuentan los pulsos KYZ ocurridos, y TiempoKWH y TiempoKVARH, a los que almacenan el tiempo entre cada pulso.

Para el pulso KYZ de sincronía, se usan dos registros de 16 bits (SINC1 y SINC2) en los que se almacena la información sobre la fecha y hora en que ocurrió el pulso (*figura 3.14*). Por esta razón, el sistema tiene implementando un Reloj/Calendario que debe configurarse al inicializar el dispositivo (consúltese el manual de usuario del CODKYZ en el *apéndice B*).

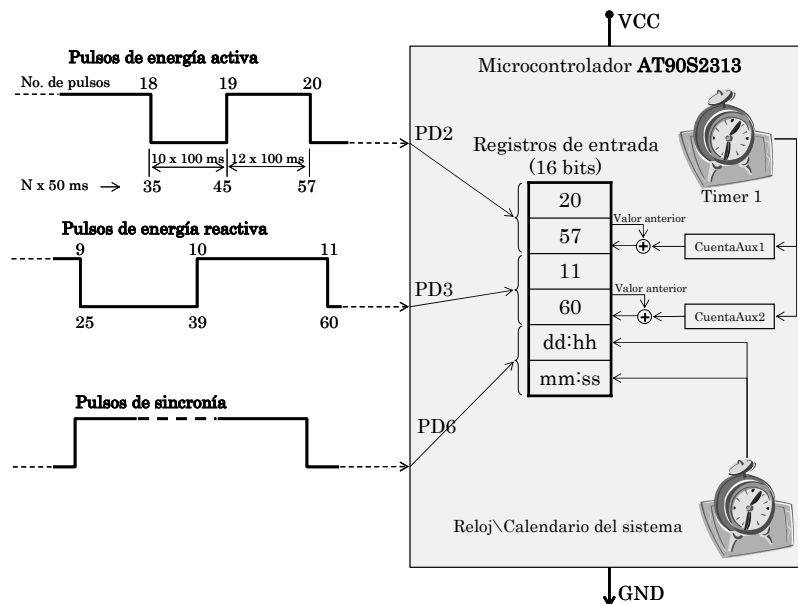


Figura 3.14. Cuantificación de los pulsos KYZ y su respaldo en memoria RAM.

Para medir el tiempo que permanece la señal en alto o en bajo entre cada pulso KYZ, se define una base de tiempo dependiente de la frecuencia máxima de conmutación que presenta el relevador de estado sólido. En el caso de los medidores de energía de la marca SquareD, la frecuencia máxima es de 10 pulsos por segundo [28]. Por ello la base de tiempo empleada en este caso en particular por el CODKYZ es configurable en 100 ó 1000 ms (1 s).

Considerando lo anterior, el *timer* 1 del microcontrolador se configura para que active su interrupción de sobreflujo cada vez que se ha completado esta base de tiempo. Se ocupan dos registros auxiliares (CuentaAux1 y CuentaAux2) para contabilizar el número de intervalos, de 100 ó 1000 ms, que ocurren entre cada transición generada por los pulsos KYZ.

Los registros PulsosKWH y PulsosKVARH se incrementan en 1 después de que ocurre un pulso KYZ de energía activa o reactiva, respectivamente. De igual modo, los registros TiempoKWH y TiempoKVARH, se suman con los registros CuentaAux1 y CuentaAux2, los cuales toman el valor 0 después de la suma.

De esta manera en los registros TiempoKWH o TiempoKVARH, se puede llegar a contar hasta un máximo de 65535 intervalos de tiempo que equivalen a más de 1 hora con 49 minutos cuando se usa la base de tiempo de 100 ms y a más de 18 horas con 12 minutos, si se usa la de 1000 ms.

En el caso de los registros SINC1 y SINC2, los datos se toman directamente de los registros de fecha y hora que ocupa el reloj del sistema.

Por consiguiente, en el momento en que se recibe un mensaje MODBUS solicitando la información de los pulsos, se inicia la codificación de los datos actuales para generar la respuesta con los datos actualizados. La *figura 3.15* ejemplifica el momento en que los pulsos KYZ de energía activa son solicitados y muestra qué valores son enviados como respuesta.

3.5.3. Codificación de los pulsos KYZ

La *codificación* de los datos consiste en construir una trama de respuesta de acuerdo a lo que establece el protocolo de comunicaciones MODBUS RTU.

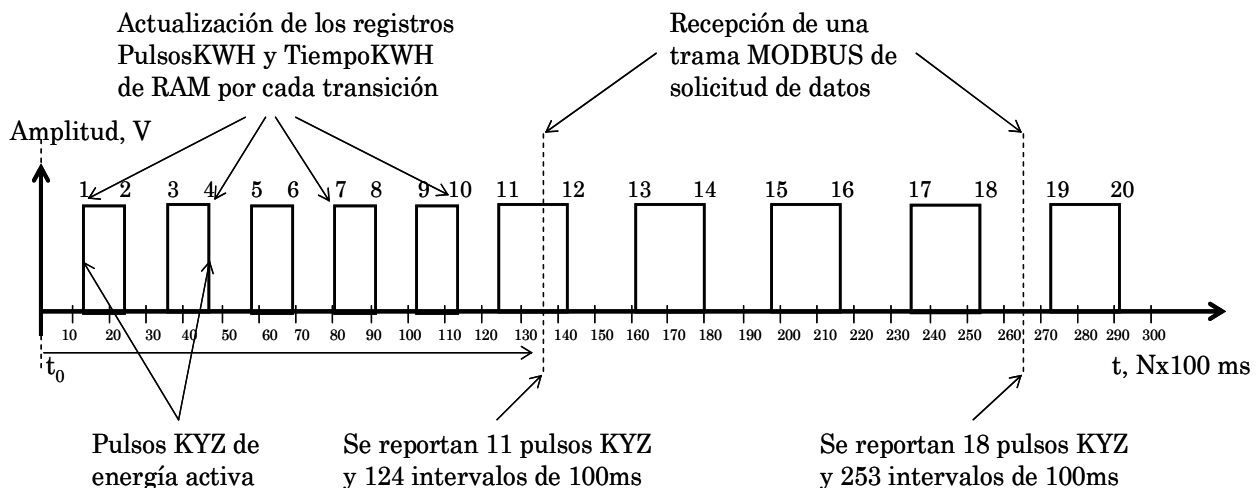


Figura 3.15. Representación de los pulsos KYZ y el momento en que se actualizan los registros.

La codificación RTU, que se emplea en este proyecto, coincide con la representación binaria de 8 bits de los registros del microcontrolador AT90S2313. De modo que para construir la trama de respuesta MODBUS que reporta la información de los pulsos KYZ, se añaden los registros correspondientes en el campo de datos. Lo anterior, y todo lo referente al soporte del protocolo de comunicaciones MODBUS, se explica con detalle en el *apartado 3.5.4*.

Cabe mencionar que todos los registros de información de los pulsos KYZ permanecen actualizados hasta que ocurre alguno de los siguientes casos: que el Maestro envíe una trama MODBUS que indique el reinicio de las cuentas, o que los registros se reinicien solos debido a un sobreflujo (debe tomarse en cuenta que son registros de 16 bits cuyo máximo valor es de 65535).

El Maestro debe evitar el sobreflujo de los registros solicitando los datos antes de que transcurra el tiempo indicado en el *apartado 3.5.2*, después es necesario enviar una orden para que el CODKYZ reinicie los registros de cuenta.

3.5.4. Implementación de MODBUS en el AT90S2313

En este trabajo de tesis se propone una manera de implementar el soporte del protocolo de comunicaciones MODBUS en el microcontrolador AT90S2313, para lo cual se requiere el uso de memorias EEPROM y RAM, de un temporizador y de la UART de este circuito integrado. A continuación se describen los aspectos más importantes de esta implementación, tomando en cuenta la aplicación que tiene este proyecto y lo que se describe en el *capítulo 2* sobre el protocolo MODBUS.

3.5.4.1. Uso de la memoria EEPROM

Cuando un dispositivo Esclavo se activa para ocuparse dentro de una red MODBUS sobre una línea serial, éste debe tener ciertos datos precargados que le permitan identificarse como un elemento de la red. Por ejemplo, la dirección del dispositivo y los datos de configuración de la UART (velocidad de transmisión, bits de datos, paridad y bits de fin). Generalmente estos datos y algunos más de configuración, se almacenan de manera permanente en una memoria EEPROM incluso cuando no está encendida.

De manera particular, el CODKYZ almacena sus *registros mantenidos* en la memoria EEPROM de 128 bytes (0x00-0x7F) que contiene el microcontrolador AT90S2313. El espacio en memoria designado para cada *registro mantenido* es de dos bytes aunque su valor no los requiera, la razón es que la norma del protocolo MODBUS especifica que todos los registros de datos (*mantenidos* o *de entrada*) deben tener esta longitud (16 bits). Los registros mantenidos de este dispositivo se muestran en la *tabla 3.4*.

Cuando el CODKYZ se inicializa, éste realiza una rutina de autoconfiguración cargando los *registros mantenidos* almacenados en la EEPROM. En caso de que por alguna razón no existan datos en la memoria, los registros de configuración del CODKYZ toman los valores por defecto que se muestran en la *tabla 3.5*.

Tabla 3.4. Registros mantenidos de configuración del CODKYZ.

Nombre	Descripción
AÑO-MES	Año y mes actuales
DÍA-HORA	Día y hora actuales
MINS-SEGS	Minutos y segundos actuales
DIRECCIÓN	Dirección de Esclavo del dispositivo (1-247)
BAUDRATE	Velocidad de transmisión (1200,2400,4800,9600,19200 bps)
INTERVALO	Valor de los intervalos de medición (100,1000 ms)
RETARDO	Tiempo programable de respuesta (0, 50 ó 100 ms)
EEPROM	Indica la existencia de datos en EEPROM (Sí=170, No= 102)

Tabla 3.5. Valores por defecto de los registros mantenidos de configuración del CODKYZ.

Registro	Valor (HEX)	Valor (DEC)	Descripción
AÑO-MES	0x0001	1	2000-ENERO-1 00:00:00 hrs.
DÍA-HORA	0x0100	16	
MINS-SEGS	0x0000	0	
DIRECCIÓN	0x0001	1	Esclavo 1
BAUDRATE	0x2580	9600	9600 baudios
INTERVALO	0x0064	1000	1000 ms
RETARDO	0x00	0	Retardo de respuesta 0
EEPROM	0x0066	102	EEPROM vacía

En el caso de los registros del RELOJ (año, mes, hora, etc.), los valores por defecto representan las 00:00:00 horas del día 1 de enero de 2000. Por lo que es recomendable que la primera trama enviada al CODKYZ, sea un mensaje de escritura de *registros mantenidos* que configure todos estos datos con la fecha y hora actuales. De este modo, el CODKYZ estará sincronizado con el Maestro. Para evitar conflictos con otros dispositivos, también se recomienda asignar la dirección de Esclavo que el CODKYZ tendrá dentro de la red.

Se definió además, una función MODBUS para que los *registros mantenidos* sean transferidos a la memoria EEPROM cuando el dispositivo Maestro lo solicite. Esta función se describe en el *párrafo 3.5.4.3.1*.

3.5.4.2. Uso de la memoria RAM

El microcontrolador AT90S2313 posee 32 registros de propósito general y una memoria RAM de 128 bytes (0x60-0xDF) para el almacenamiento temporal de datos. Se reservaron 32 bytes de la RAM para recibir las tramas MODBUS de los mensajes de solicitud enviadas por el Maestro; 16 para los *registros mantenidos*; 12 para los *registros de entrada*, es decir, para la información de los pulsos KYZ; 16 para los *registros contadores de eventos de comunicación* y 40 para construir los mensajes de respuesta MODBUS. La organización de estos datos se muestra en la *figura 3.16* que corresponde al mapa de memoria del CODKYZ.

En el bloque de memoria utilizado para recibir mensajes de solicitud, se almacenan consecutivamente los bytes que forman parte de la trama MODBUS de solicitud.

Dirección inicial	Uso del bloque de memoria	Espacio del bloque
0x60	Recepción de tramas MODBUS	32 bytes
0x80	Registros mantenidos	16 bytes
0x90	Registros de entrada	12 bytes
0x9C	Registros contadores de eventos de comunicación	16 bytes
0xAC	Sin uso	4 bytes
0xB0	Construcción de tramas MODBUS	40 bytes
0xDF		

Figura 3.16. Mapa de memoria del CODKYZ.

Los datos de la trama se empiezan a almacenar una vez que se ha recibido el primer byte después del delimitador de inicio ($t_{3.5}$), es decir, enseguida de que ha transcurrido un tiempo mayor o igual a tres veces y medio un tiempo de un byte o caracter y se finaliza con este mismo *tiempo de silencio* (*inciso 2.2.2.1*).

3.5.4.2.1. Memoria para recepción de tramas MODBUS

Para que una trama sea aceptada debe tener una longitud mínima de cuatro bytes: identificador (1 byte), función sin datos (1 byte) y el CRC -16 (2 bytes). En caso contrario, se considera un error de comunicación y el apuntador de la memoria retorna al inicio del bloque para reiniciar la construcción de otro paquete. Si la longitud de la trama excede los 32 bytes, ésta se descarta automáticamente, pues ninguno de los mensajes de solicitud destinados al CODKYZ debe ser mayor, cuando esto sucede, el apuntador de memoria se lleva al inicio del bloque.

Si el paquete se acepta, se procede a validarlo, lo cual se refiere a comprobar que el campo de CRC-16 recibido coincida con el que calcula el microcontrolador. Si estos registros son iguales, el paquete se considera válido; si no, se considera un evento de comunicación erróneo y el mensaje se rechaza sin enviar respuesta. El algoritmo para calcular el CRC-16 empleado para la detección de tramas erróneas se describe en el *inciso 2.2.3.1*, el diagrama de flujo se ilustra en la *figura 3.17*.

Antes de atender el mensaje, el paquete se somete a una verificación más: se examina el campo de dirección y se cerciora que sea igual a la dirección de Esclavo del CODKYZ. Una vez más, el mensaje se ignora si no está dirigido a este dispositivo, sin embargo, se actualiza el contador de eventos de comunicación correspondiente a los mensajes detectados en el bus.

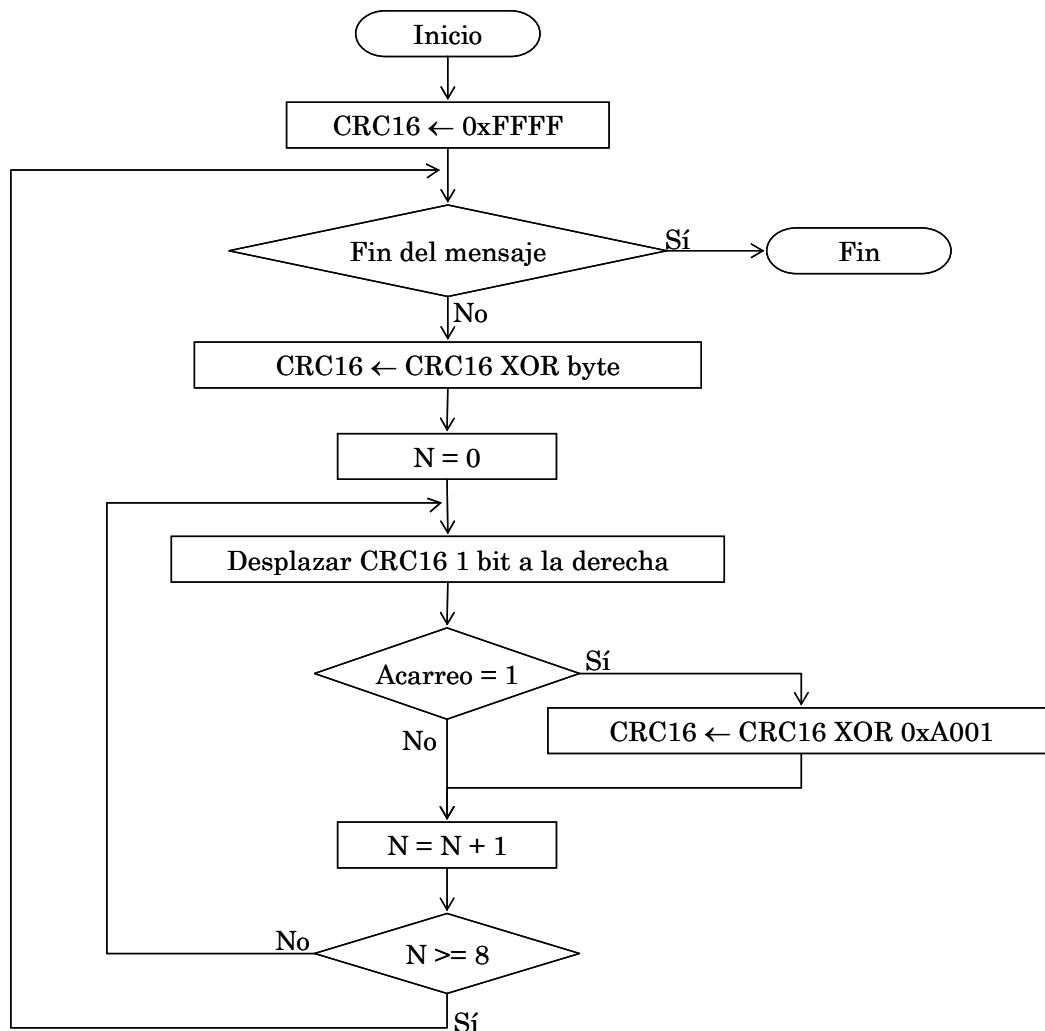


Figura 3.17. Diagrama de flujo del algoritmo para calcular el CRC-16 de una trama MODBUS.

En cambio, si las direcciones son idénticas, el paquete permanece temporalmente en memoria RAM para que posteriormente se atienda el mensaje que se ha recibido (*inciso 3.5.4.3*).

El espacio de 16 bytes de RAM empleado para almacenar temporalmente los *registros mantenidos* es igual al ocupado en la EEPROM para almacenarlos permanentemente. La diferencia radica en que los registros en RAM son accedidos de manera más rápida que los almacenados en EEPROM, además de que sus valores pueden ser modificados a través de la función 3 de MODBUS para configurar temporalmente el CODKYZ. Cuando éste se apaga, los *registros mantenidos* actuales se pierden si no se transfieren a la EEPROM.

3.5.4.2.2. Memoria para registros mantenidos y de entrada

En el bloque de memoria RAM designado para los *registros de entrada* se acumulan los datos relacionados con la información de los pulsos KYZ que se ha adquirido del medidor de energía. En la *tabla 3.6* se presentan los *registros de entrada* del CODKYZ. Cada vez que ocurre un pulso KYZ, los registros correspondientes son extraídos y actualizados con la información de cuantificación del pulso.

Tabla 3.6. Registros de entrada del CODKYZ.

Dirección (hex)	Nombre	Descripción
0008	KWH	Número de pulsos KYZ que reportan energía en KWH
0009	TiempoKWH	Número de intervalos de 100 ms después de la última consulta
000A	KVARH	Número de pulsos KYZ que reportan energía en KVARH
000B	TiempoKVAR	Número de intervalos de 100 ms después de la última consulta
000C	SINC1	Día y hora de ocurrencia del último pulso de sincronía
000D	SINC2	Minutos y segundos de ocurrencia del último pulso de sincronía

3.5.4.2.3. Memoria para registros contadores de eventos de comunicación

La *tabla 3.7* lista todos los registros que cuentan determinados eventos de comunicación ocurridos mientras el CODKYZ se mantiene activo. Se incluyen además los registros de diagnóstico y de estado del dispositivo.

Al presentarse un determinado evento de comunicación, se extrae de la memoria el registro correspondiente y se actualiza incrementando su valor en 1. Estos registros pueden ser consultados únicamente enviando un mensaje de solicitud con el código de función 0x08 y la subfunción relacionada al contador del evento de comunicación requerido (*tabla 2.9*).

3.5.4.2.4. Memoria para transmisión de tramas MODBUS

El mensaje de respuesta más largo, generado por el CODKYZ, es el que reporta los datos de identificación del dispositivo, cuya longitud es de 40 bytes. Considerando la longitud de este mensaje, se han reservado 40 bytes de RAM para construir estas tramas. Las tramas MODBUS de respuesta se construyen en este espacio de memoria de acuerdo a la función que se ha solicitado ejecutar. Los datos que las forman se toman de los registros existentes en la memoria y/o de las propias tramas de solicitud.

Tabla 3.7. Registros contadores de eventos de comunicación del CODKYZ.

No. De contador	Nombre	Descripción
--	DIAGNOS	Registro de diagnóstico
CON1	MENSBUS	Contador de mensajes detectados en el BUS del Esclavo
CON2	CRCERR	Contador de errores de CRC
CON3	RESPEXC	Contador de respuestas de excepción
CON4	MENRESP	Contador de mensajes ejecutados por el Esclavo
CON5	NORESP	Contador de solicitudes del Maestro sin respuestas
CON6	COMEXIT	Contador de comunicaciones exitosas
--	REGSTAT	Registro de estado del dispositivo

Cuando ya se ha terminado de construir una trama de respuesta, se calcula el CRC-16 y se coloca en el campo de comprobación de error localizado al final de la trama. Después, todos los bytes que conforman la trama se extraen uno por uno y se transmiten para que sean recibidos por el dispositivo Maestro de la red MODBUS.

3.5.4.3. Atención a un mensaje MODBUS

Una vez aceptada por completo una trama MODBUS y almacenada temporalmente en memoria RAM, se atiende el mensaje que ésta contiene. El proceso se divide en dos fases que pueden ser no consecutivas, pero que

forzosamente producen un mensaje de respuesta para el Maestro, bien puede ser una respuesta de excepción que se genera en la primera fase o uno exitoso en la segunda.

La primera fase consiste en validar la información contenida en los campos de función y de datos (subfunciones, direcciones, cantidad de registros por leer/escribir, valores de registros, etc.) de la trama MODBUS. Si la validación de cualquiera de los valores de estos campos resulta incorrecta, es decir, si los datos recibidos no coinciden con los que este dispositivo Esclavo espera, se produce un *error de excepción*, y por lo tanto, un *mensaje de excepción MODBUS*. Los errores de excepción que genera el CODKYZ se describen en el *párrafo 3.5.4.3.4*.

La segunda fase se lleva a cabo sólo si la primera no resultó en una excepción y se refiere a la ejecución de la función que el dispositivo Maestro indica en el mensaje MODBUS que ha enviado al Esclavo.

3.5.4.3.1. Validación de los campos de función y subfunción

La validación de los campos de función y subfunción consiste en corroborar que el número incluido en estos campos del mensaje MODBUS, corresponda con alguno de los códigos de las funciones y/o subfunciones que soporta el CODKYZ.

Si los códigos de función y/o subfunción recibidos no son parte de las tareas que puede ejecutar el CODKYZ, éste elabora un mensaje de excepción que reporta el código de error 01 en el campo de datos. En caso contrario, se continúa con el análisis de la trama recibida.

En la *tabla 3.8* se enumeran las funciones y en la *tabla 3.9*, las subfunciones MODBUS implementadas y soportadas en este dispositivo Esclavo y que han sido explicadas en el *apartado 2.2.4*.

Para tener un mejor control del codificador, se implementaron dos funciones MODBUS de tarea específica de acuerdo a la aplicación, sus códigos se encuentran en el rango de funciones definidas por el usuario.

El código de la primera función es 101 (0x65) y sirve para solicitar al CODKYZ que respalde todos los *registros mantenidos* en la memoria EEPROM del sistema o para que sean borrados de ella. De este modo, cuando el dispositivo se inicializa, estos registros toman sus valores por defecto. Si no se presenta algún error, el mensaje de respuesta a esta función es una réplica de la solicitud. Un ejemplo de las PDUs de una transacción MODBUS relacionada con esta función aparece descrito en la *tabla 3.10*.

El código de la segunda función implementada es 105 (0x69) y se emplea para ordenar al CODKYZ que reinicie los *registros de entrada* (información de los pulsos KYZ) con el valor 0 sin importar su contenido actual. Esta función sirve para evitar que los *registros de entrada* experimenten un sobreflujo. La respuesta del CODKYZ es idéntica a la solicitud si no hubo error al ejecutar esta función. Las PDUs de estas tramas se muestran *tabla 3.11* y consisten únicamente en el campo de función indicando la tarea que el CODKYZ debe realizar.

Tabla 3.8. Códigos de funciones MODBUS que soporta el CODKYZ.

DEC	HEX	Descripción de la función
3	03	Lectura del contenido de los registros mantenidos de configuración
4	04	Lectura del contenido de los registros de entrada
6	06	Escritura de un registro mantenido
8	08	Función de diagnóstico (ver las subfunciones de diagnóstico)
11	0B	Lectura de estado de operación y número de comunicaciones con éxito
16	10	Escritura de múltiples registros mantenidos
23	17	Escritura y lectura de múltiples registros mantenidos y de entrada
43	2B	Lectura de los datos de identificación del dispositivo

Tabla 3.9. Códigos de subfunciones MODBUS que soporta el CODKYZ.

DEC	HEX	Descripción de la subfunción
0	00	Solicitud de eco
1	01	Reinicio de comunicación y de contadores
2	02	Retornar el registro de diagnóstico
4	04	Activar el MSE
10	0A	Reinicio de contadores de eventos de comunicación
11	0B	Retornar el registro contador de mensajes en BUS
12	0C	Retornar el registro contador de errores de comunicación (CRC)
13	0D	Retornar el registro contador de repuestas de excepción
14	0E	Retornar el registro contador de mensajes hacia el Esclavo
15	0F	Retornar el registro contador de mensajes sin respuesta

Tabla 3.10. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x65.

Solicitud		Respuesta	
Nombre del campo	Valor (hex)	Nombre del campo	Valor (hex)
Función	65	Función	65
Dato que activa el resguardo (byte alto)	00	Resguardo activado (byte alto)	00
Dato que activa el resguardo (byte bajo)	AA	Resguardo activado (byte bajo)	AA

Tabla 3.11. Ejemplo de las PDUs de una transacción MODBUS ejecutando la función 0x69.

Solicitud		Respuesta	
Nombre del campo	Valor (hex)	Nombre del campo	Valor (hex)
Función	69	Función	69

3.5.4.3.2. Rango de direcciones válidas

En MODBUS, los *registros mantenidos* del dispositivo Esclavo se enumeran de manera lógica a partir de la dirección 40001 y los *registros de entrada*, a partir de la 30001. Los primeros son tanto de lectura como de escritura (L-E) y los segundos sólo de lectura (L). El dígito inicial de las direcciones se utiliza para distinguir unos de los otros, ya que en el campo de datos, se manejan considerado el registro inicial a partir de 0001. En realidad, cuando se quiere acceder a ellos ya sea para lectura o escritura, este número representa el registro 0x0000. Sin embargo, estas direcciones no representan físicamente la ubicación de los *registros mantenidos* en la memoria del dispositivo. La dirección real en RAM queda a elección del programador.

En las *tablas 3.12 y 3.13* se exhiben las distintas direcciones de los *registros mantenidos y de entrada* explicadas previamente.: lógicas, físicas, y las que se especifican en el campo de datos al enviar un mensaje.

Tabla 3.12. Direcciones lógicas, físicas y del campo de datos de los registros mantenidos.

Lógica	RAM (hex)	Campo de datos (hex)	L/E	Nombre
40001	80-81	0000	L-E	AÑO-MES
40002	82-83	0001	L-E	DÍA-HORA
40003	84-85	0002	L-E	MINS-SEGS
40004	86-87	0003	L-E	DIRECCIÓN
40005	88-89	0004	L-E	BAUDRATE
40007	8C-8D	0005	L-E	INTERVALO
40006	8A-8B	0006	L-E	RETARDO
40008	8E-8F	0007	L-E	EEPROM

Tabla 3.13. Direcciones lógicas, físicas y del campo de datos de los registros de entrada.

Lógica	RAM	Campo de Datos (hex)	L/E	Nombre
30001	90-91	0008	L	KWH
30002	92-93	0009	L	Tiempo1
30003	94-95	000A	L	KVARH
30004	96-97	000B	L	Tiempo2
30005	98-99	000C	L	SINC1
30006	9A-9B	000D	L	SINC2

Como se puede observar en las tablas anteriores, se eligió el rango de direcciones físicas 0x80-0x8F de la memoria RAM del microcontrolador para almacenar los *registros mantenidos* y 0x90-0x9B para los *registros de entrada*, los cuales conservarán su valor mientras el CODKYZ permanezca activo. Como los *registros mantenidos y de entrada* son de 16 bits, se usan dos bytes de memoria para su almacenamiento. Así, las direcciones 0x80 y 0x81 almacenan las partes alta y baja respectivamente, del *registro mantenido* 40001 (dirección 0x0000 en MODBUS) y las direcciones 0x90 y 0x91, el *registro de entrada* 30001 (0x0000 en MODBUS).

En la lectura y escritura de *registros mantenidos* se indica la dirección de inicio y la cantidad de registros que serán leídos o escritos. Si el rango de direcciones, comprendido por la dirección inicial y el número de registros, excede el límite de las direcciones válidas de los *registros mantenidos o de entrada*, se genera un mensaje de excepción con código de error 02.

3.5.4.3.3. Validación de los valores de registros

Para que un *registro mantenido* pueda ser modificado, el mensaje de solicitud de escritura debe portar datos afines al registro que desea escribir. Por ejemplo, si desea modificar el *registro mantenido BAUDRATE*, el campo de datos debe portar sólo valores de velocidad que el dispositivo soporta. Si se quiere modificar los registros de fecha y hora, los datos proporcionados deben coincidir con el formato de una fecha válida para el sistema, refiérase al apéndice B para conocer los valores aceptados para los *registros mantenidos* del CODKYZ.

La validación de los valores se realiza examinando la información del campo de datos, se identifican los registros que van a ser modificados y se comprueba que los valores proporcionados por el dispositivo Maestro se encuentren dentro de los

rangos de valores aceptados para cada *registro mantenido*. Si la verificación reporta algún dato no válido, se responde con un mensaje de excepción con código de error 3.

3.5.4.3.4. Códigos de excepción

Cuando el mensaje de solicitud falla en cualquiera de las validaciones antes descritas, se genera el mensaje de excepción correspondiente. De acuerdo a lo que se expone en el *inciso 2.2.2.6* sobre las respuestas de excepción, el código de función se suma con 0x80 y en el campo de datos se escribe el código del error que produjo la excepción.

En la *tabla 3.14* se muestran los tres códigos reportados por el CODKYZ cuando se produce cualquiera de los errores de validación explicados anteriormente.

Tabla 3.14. Códigos de excepción soportados por el CODKYZ.

DEC	HEX	Descripción del código
0	01	Se ha solicitado al dispositivo ejecutar una función que no soporta
1	02	Se ha intentado leer o escribir una dirección ilegal
2	03	El valor que se desea escribir en un registro no es válido

3.5.4.3.5. Ejecución de la función

Si la trama MODBUS no fue rechazada y no se generó una respuesta de excepción, se procede a ejecutar la función que el mensaje indica y se construye la trama que contiene la información solicitada o la confirmación de que ya se ha efectuado la operación requerida. Las tramas de respuesta se construyen de acuerdo a lo que describe en el *capítulo 2* respecto a las funciones del protocolo MODBUS.

3.5.5. Descripción del firmware del CODKYZ

En la *figura 3.18* se ilustra el diagrama de flujo de la función principal del programa implementado en el CODKYZ. El diagrama muestra el momento en que son invocadas las subrutinas (o funciones) de mayor importancia del programa, las cuales se han descrito a lo largo de este capítulo.

La primera función importante por describir es **Inicializa_Registros**, la cual pone en 0 todos los *registros de entrada* de la memoria RAM, pues contendrán la información de los pulsos KYZ que se van a *monitorear*.

La siguiente es **Configurar_dispositivo**, esta función toma en cuenta la existencia de datos de configuración en la EEPROM, primero extrae un registro de 16 bits que indica, con un valor predefinido, si hay o no datos en ella. Si existen datos, se invoca otra función encargada de extraer todos los datos de la memoria EEPROM a la RAM y con ellos se configura el dispositivo; si no, se toman los valores por defecto previamente establecidos (*tabla 3.5*).

Otra función importante es **Configura_PulsosKYZ**, pues es la que configura los *pin*s de entrada del microcontrolador con los que se detectan los pulsos KYZ de acuerdo a lo descrito en el *apartado 3.5.1*.

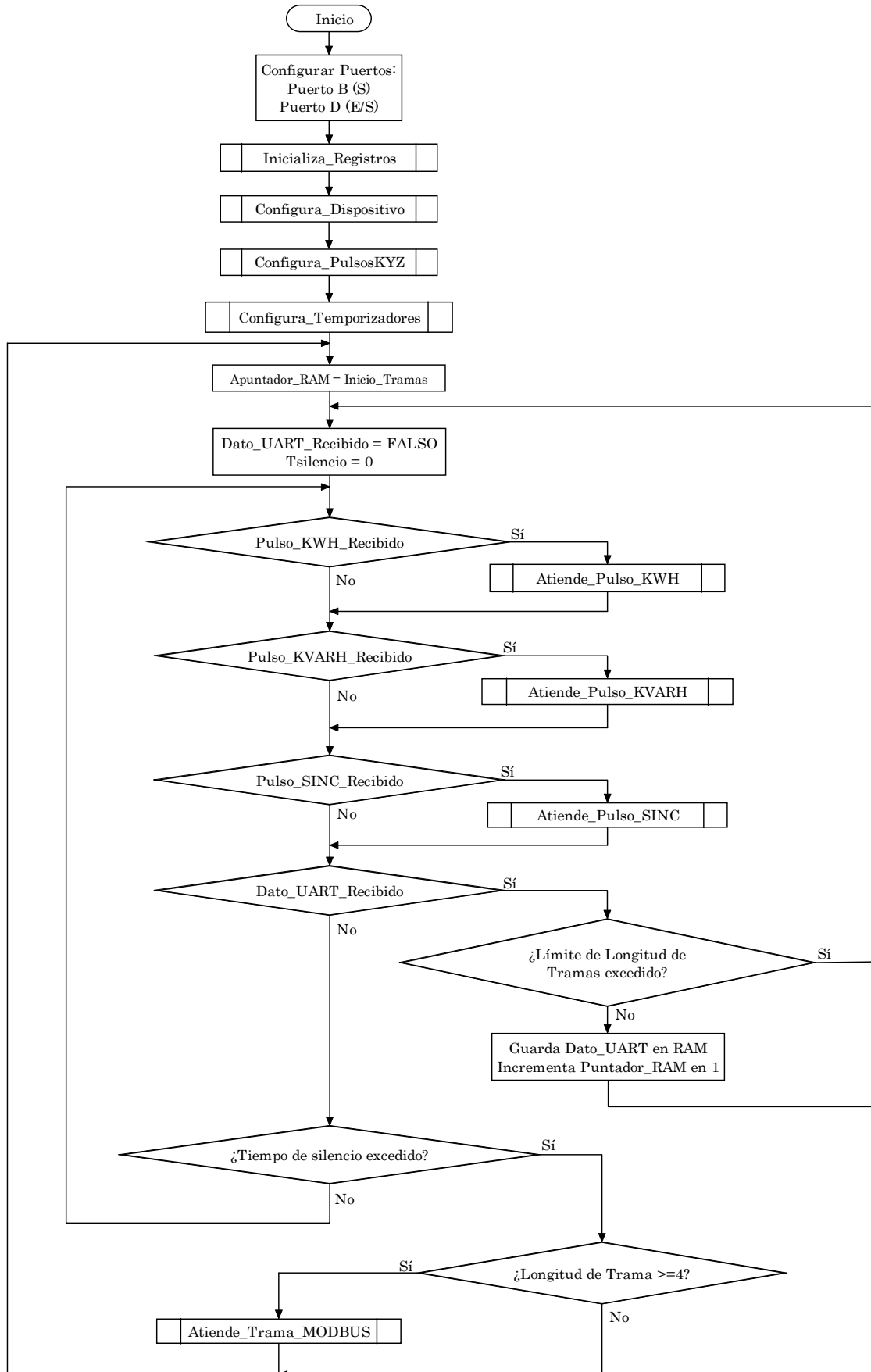


Figura 3.18. Diagrama de flujo de la función principal del programa implementado en el CODKYZ.

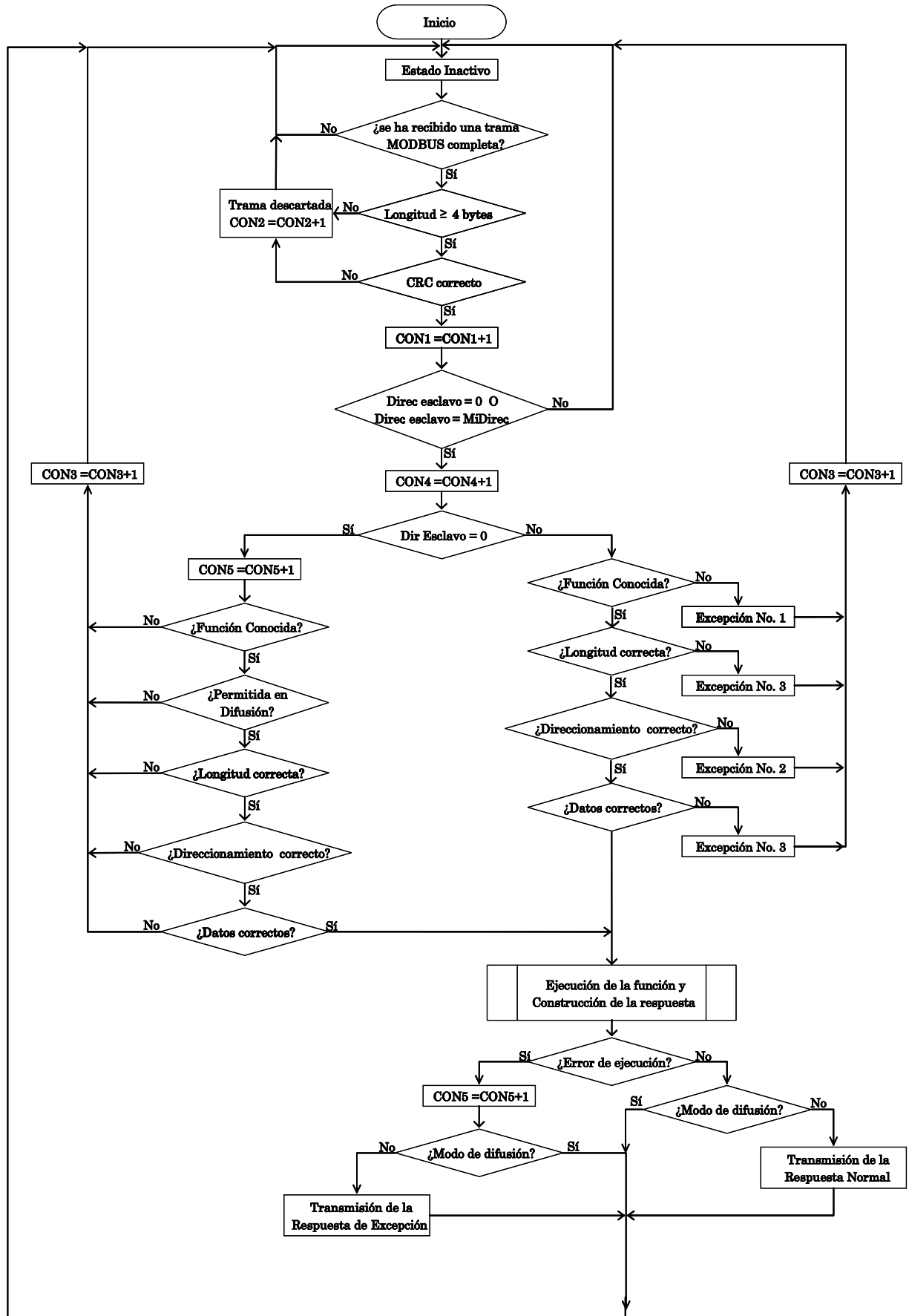


Figura 3.19. Diagrama de flujo de la función Atiende_Trama_MODBUS.

Con la función **Configura_Temporizadores** se configuran los *timer* 0 y 1 para que, a través de interrupciones de sobreflujo, se obtengan las bases de tiempo que se necesitan para controlar los eventos dependientes del tiempo, por ejemplo, el Reloj/Calendario y el tiempo que dura en bajo o en alto la señal de cada pulso KYZ.

Atiende_Pulso_KWH, **Atiende_Pulso_KVARH** y **Atiende_Pulso_SINC**, son subrutinas de tarea específica que se invocan a partir de la detección de los pulsos KYZ de energía activa, energía reactiva y sincronía, respectivamente (*apartados 3.5.1 y 3.5.2*).

La subrutina **Atiende_Trama_MODBUS** se ejecuta en el momento que se completa una posible trama de solicitud. Sin embargo, a partir del programa principal se implementan todas las tareas especificadas por la norma del protocolo para iniciar una transacción MODBUS. La ejecución de cada función MODBUS y la construcción de las respuestas se efectúan de acuerdo a lo descrito en el *capítulo 2* y en el *apartado 3.5.4*. En la *figura 3.19* se muestra el diagrama de flujo de esta función.

4. Desarrollo del software de pruebas y configuración del CODKYZ

El software de configuración y pruebas del CODKYZ, denominado Soft-KYZ, forma parte fundamental de este trabajo de tesis¹², ya que, además de ser una herramienta para verificar el funcionamiento correcto del CODKYZ, también funciona como un simulador de MODBUS para probar cualquier otro dispositivo que opere bajo este protocolo. En este capítulo se describe el proceso seguido durante su desarrollo.

4.1. Lenguaje de programación

El lenguaje de programación elegido para implementar el Soft-KYZ es **LabVIEW** versión **6.1** de *National Instruments* [URL3]. Las principales razones por las que se optó por este lenguaje son:

- Utiliza el paradigma de programación gráfica, el cual es un concepto para elaborar programas cuyo uso en México está aumentando [URL12].
- Es ampliamente usado en la industria para crear instrumentos virtuales orientados a la instrumentación y al *monitoreo*, automatización y control de procesos industriales [7, URL12].
- Ofrece facilidad para construir interfaces gráficas de usuario (GUI, *Graphical User Interfaces*) dirigidas al entorno industrial y científico [29].
- Se adapta a las necesidades de desarrollo del Soft-KYZ debido a su enfoque modular para la elaboración de programas.
- Cada módulo que se implementa en LabVIEW se puede ejecutar y probar de manera independiente del programa principal [30].
- En la UTM se cuenta con una licencia de la versión profesional de este lenguaje, la cual permite elaborar el archivo ejecutable de los programas que en él se realizan.

¹² Anexo al documento de tesis se incluye un disco compacto que contiene el código fuente y el archivo de ayuda de este programa.

4.2. Modelo de desarrollo

Para el desarrollo del Soft-KYZ se siguió un *modelo incremental*, es decir, se llevó a cabo a través de incrementos de programa reutilizables que sirvieron como base para construir las partes faltantes del software. En cada incremento se siguió la secuencia de pasos que establece este modelo del ciclo de vida del desarrollo de software: *definición y análisis de requerimientos, diseño, codificación y pruebas* del software de manera repetitiva hasta su culminación [31, 32]. Como puede apreciarse en la *figura 4.1*, las fases se realizaron cíclicamente hasta llegar al programa final.

De acuerdo con [31], sin importar el modelo o la metodología empleada, el software se crea aplicando tres etapas distintas (*figura 4.2*):

- Definición, ¿qué se va a hacer?
- Desarrollo, ¿cómo se va a hacer?
- Mantenimiento, ¿qué cambios tendrá?

Considerando el modelo incremental, la *definición y análisis de los requerimientos* constituyen la etapa de definición; el *diseño y codificación* del software conforman la etapa de desarrollo; y las *pruebas*, la de mantenimiento. Partiendo de esta clasificación, a continuación se describen las fases del desarrollo del Soft-KYZ.

4.3. Requerimientos del Soft-KYZ

Dada su naturaleza, el Soft-KYZ se clasifica como un software de *propósito específico*, pues prácticamente está enfocado a la configuración y a la elaboración de pruebas de un solo dispositivo, el CODKYZ [32]. Con base en esta consideración se elaboraron los requerimientos funcionales de su implementación.

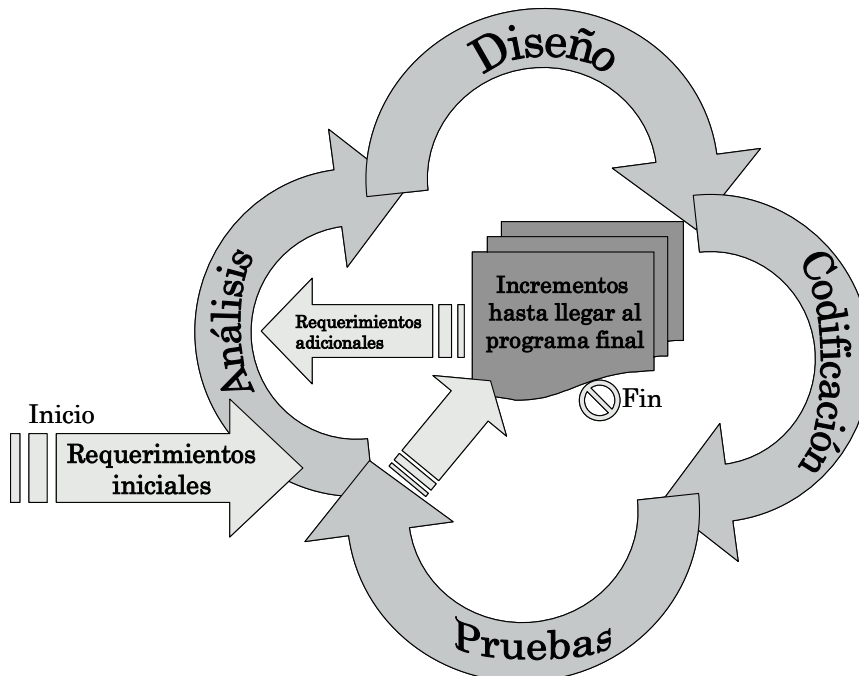


Figura 4.1. Representación de las fases del modelo incremental aplicado al desarrollo del Soft-KYZ.

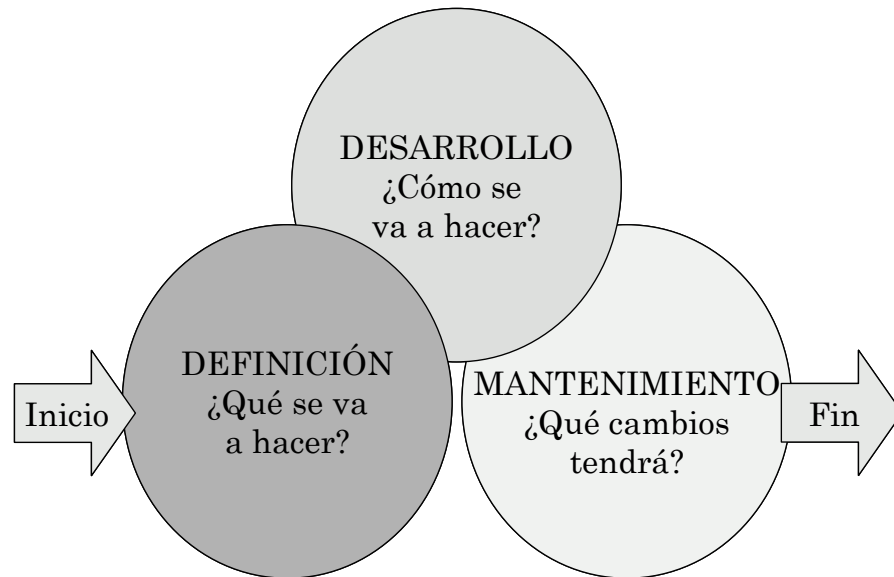


Figura 4.2. Fases del desarrollo del software.

4.3.1. Requerimiento general

Simular el comportamiento del nodo Maestro de una red MODBUS para configurar y realizar pruebas de funcionamiento del dispositivo Esclavo CODKYZ conectado a la red.

4.3.1.1. Requerimientos específicos:

A partir del requerimiento general se detallan los requerimientos específicos de funcionamiento del Soft-KYZ:

- Ejecutar un proceso de *monitoreo* que solicite los datos de cuantificación de los pulsos KYZ (*registros de entrada*) que el dispositivo Esclavo contiene.
- Calcular las variables de consumo eléctrico de la planta analizada con los datos de los pulsos KYZ y reportar sus valores a través de indicadores y gráficas en cada intervalo de consulta.
- Almacenar la información de consumo en un archivo conforme el *monitoreo* transcurra para que pueda ser abierto en Microsoft Excel en un análisis posterior.
- Llevar un registro de las transacciones MODBUS que el Soft-KYZ ha efectuado con el CODKYZ de manera exitosa o errónea.
- Ofrecer al usuario la opción de solicitar los datos de eventos de comunicación que el CODKYZ almacena (*registros de diagnóstico*).
- Ofrecer al usuario la opción de consultar y modificar los datos de configuración actual del CODKYZ (*registros mantenidos*).
- Contar con una sección en donde se construyan manualmente tramas MODBUS y se puedan efectuar transacciones con cualquier dispositivo Esclavo MODBUS.
- Presentar una GUI amigable para que el usuario pueda interactuar con el Soft-KYZ y el CODKYZ.
- Contar con un archivo de ayuda para orientar al usuario en su uso.

4.4. Análisis de los requerimientos

En la fase de análisis se interpretaron las condiciones de funcionamiento del sistema de acuerdo a los requerimientos presentados anteriormente. La labor principal en esta fase fue hacer un análisis modular del problema separando los requerimientos del programa en cuatro tareas principales. Estas tareas se atienden con funciones específicas ejecutadas por módulos, los cuales se ubican en tres bloques funcionales de acuerdo a lo que realizan dentro del programa. A continuación se dan los detalles de esta fase.

4.4.1. Análisis modular del Soft-KYZ

Nuevamente, considerando el requerimiento general del *apartado 4.3.1*, el **objetivo principal del Soft-KYZ** es: emplear la topología Maestro-Esclavo del protocolo de comunicaciones MODBUS para configurar el dispositivo CODKYZ y solicitar los registros actuales de información, diagnóstico y configuración que éste almacena para comprobar que opera adecuadamente dentro de una red MODBUS.

Para cumplir con el objetivo anterior y con los requerimientos específicos del *inciso 4.3.1.1*, el programa se divide en cuatro tareas principales:

- **Monitoreo de los pulsos KYZ:** se encarga de solicitar la información de los pulsos KYZ al CODKYZ y con ella calcula los valores de las variables de consumo eléctrico que registra la planta analizada para reportarlos al usuario mediante indicadores o gráficas.
- **Registro de las transacciones MODBUS:** reporta los *datos de las transacciones* MODBUS efectuadas entre el Soft-KYZ y el CODKYZ
- **Configuración de los dispositivos:** se utiliza para configurar los parámetros de operación tanto del Soft-KYZ como del CODKYZ.
- **Demostración del protocolo MODBUS:** permite al usuario entender el funcionamiento del protocolo MODBUS y la estructura de sus tramas.

Todas estas tareas pueden ser controladas por el usuario, pero además, el programa maneja un archivo de reporte que contiene la información de las variables eléctricas calculadas y las transacciones MODBUS efectuadas.

Para tener una mejor organización de la estructura modular del Soft-KYZ, los módulos que atienden las tareas del programa se ubican en cada uno de los bloques funcionales del Soft-KYZ de acuerdo con la función que realizan.

4.4.2. Bloques del Soft-KYZ

Los bloques *funcionales* que integran el Soft-KYZ son: *bloque de presentación*, *bloque de procesamiento* y *bloque de reporte* (figura 4.3).

4.4.2.1. Módulos del bloque de presentación

En el bloque de presentación se ubican los módulos que reciben órdenes directas del usuario. Estos módulos son parte de los *escenarios* o *casos de uso* del Soft-KYZ, es decir, los casos en los que el usuario activa directamente algún evento o procesamiento.

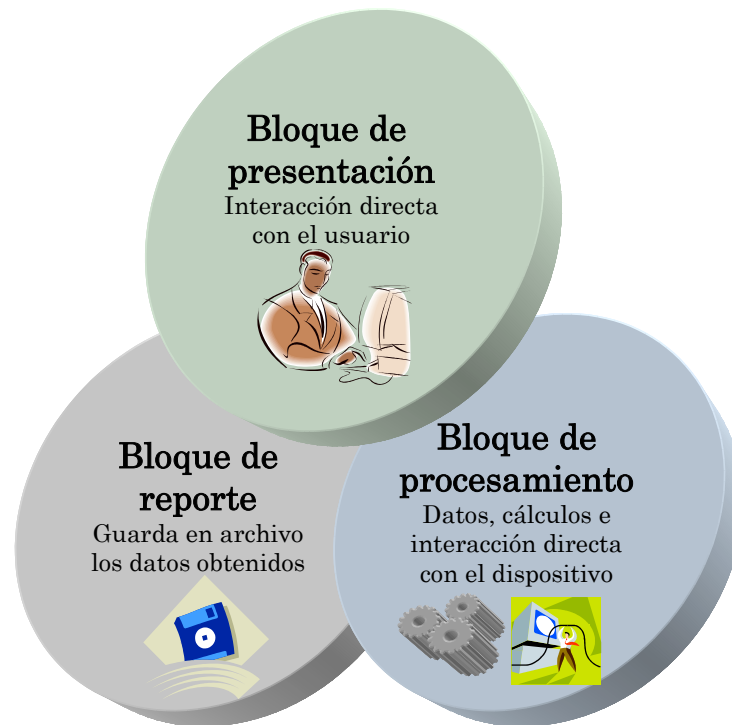


Figura 4.3. Bloques funcionales del Soft-KYZ.

4.4.2.2. Módulos del bloque de procesamiento

En el bloque de procesamiento se colocan los módulos que ejecutan las funciones referentes a la operación del sistema como manipulación de datos, cálculos, e intercambio de información con el dispositivo esclavo.

4.4.2.3. Módulos del bloque de reporte

En el bloque de reporte se incluyen los módulos que realizan tareas relacionadas con la manipulación del archivo de reporte.

En el *subcapítulo 4.6* se describen con mayor detalle las tareas principales del Soft-KYZ y los módulos que las atienden en cada bloque funcional.

4.5. Diseño del Soft-KYZ

La fase de diseño no sólo implica definir cómo se van a implementar los módulos analizados, sino también qué formato tendrán los datos que se manipularán, qué relación arquitectónica y funcional existirá entre los módulos en conjunto y qué interfaz de usuario se ajusta a esta relación. Después se hace el diseño de los módulos a nivel de componentes, es decir, módulo a módulo. En esta sección se explica cómo se llevó a cabo la fase de diseño del Soft-KYZ.

4.5.1. Diseño de los datos

Al analizar los módulos más importantes del Soft-KYZ se definieron cuáles serían los datos que cada uno debe manipular. Sin embargo en la fase de diseño de datos se concretó qué tipo de datos o qué estructuras de datos se ocuparían como entradas y salidas en cada módulo.

4.5.2. Diseño del archivo de reporte

La información del reporte generado durante el *monitoreo* de las variables de consumo eléctrico se almacena en un archivo de texto (codificación ASCII), cuya extensión por omisión es “.kyz”. En él aparece una tabla con las etiquetas de las variables eléctricas como encabezado y sus valores numéricos como datos. El formato de la tabla es matricial con separaciones por tabuladores, de modo que el archivo puede ser abierto con cualquier versión de Microsoft Excel.

4.5.3. Diseño de la arquitectura

El diseño de la arquitectura de software de un programa, se refiere al establecimiento de las relaciones que deben existir entre los elementos que lo conforman y sus propiedades. Existen distintos estilos arquitectónicos que definen estas relaciones. En el caso del Soft-KYZ se ha hecho una combinación entre dos arquitecturas: la primera de ellas es la arquitectura *programa principal-subprograma* y la segunda, la arquitectura *estratificada* [31].

La primera se basa en la descomposición del programa en una estructura jerárquica de control, en la cual existe un programa principal que invoca a un número de subprogramas y éstos, a su vez, pueden llamar a otros más. El Soft-KYZ se diseñó para seguir esta arquitectura clásica de programación, también llamada “divide y vencerás” [31].

La estructura estratificada se refiere a que cuenta con *bloques funcionales* que atienden diversas tareas a través de módulos o componentes del sistema. En la *figura 4.4* se hace una representación del diseño arquitectónico del Soft-KYZ, en ella se muestran las relaciones que se establecen entre los bloques funcionales y los módulos más importantes del programa destinados a atender las tareas principales del Soft-KYZ.

4.5.4. Diseño de la interfaz de usuario

En esta fase del diseño, se pensó en un ambiente gráfico en el que el usuario pueda interactuar fácilmente con el CODKYZ. En el diseño de la GUI se siguió el mismo paradigma que en el análisis de los requerimientos, separar las tareas que debe realizar el programa en diferentes secciones.

Con base en lo anterior, se seleccionó la opción de la GUI basada en páginas (o pestañas), la cual ofrece la facilidad de atender varias tareas usando una sola ventana de programa. Cada página de la ventana se destina a cada una de las tareas principales del Soft-KYZ mencionadas en el *apartado 4.4.1* y cuyos módulos de presentación aparecen ubicados en el bloque de presentación de la *figura 4.4*.

La desventaja de este diseño radicó en la codificación, pues en el caso de LabVIEW, si una GUI de este tipo contiene varios controles e indicadores en cada página, el número de variables que se manejan en el módulo principal puede provocar un código grande y poca modularidad inicial. Sin embargo, el compromiso vale la pena porque este estilo de GUI es entendible para la mayoría de los usuarios finales [29]. Al final de este capítulo, en la *figura 4.24*, se presenta parte de la GUI final del Soft-KYZ.

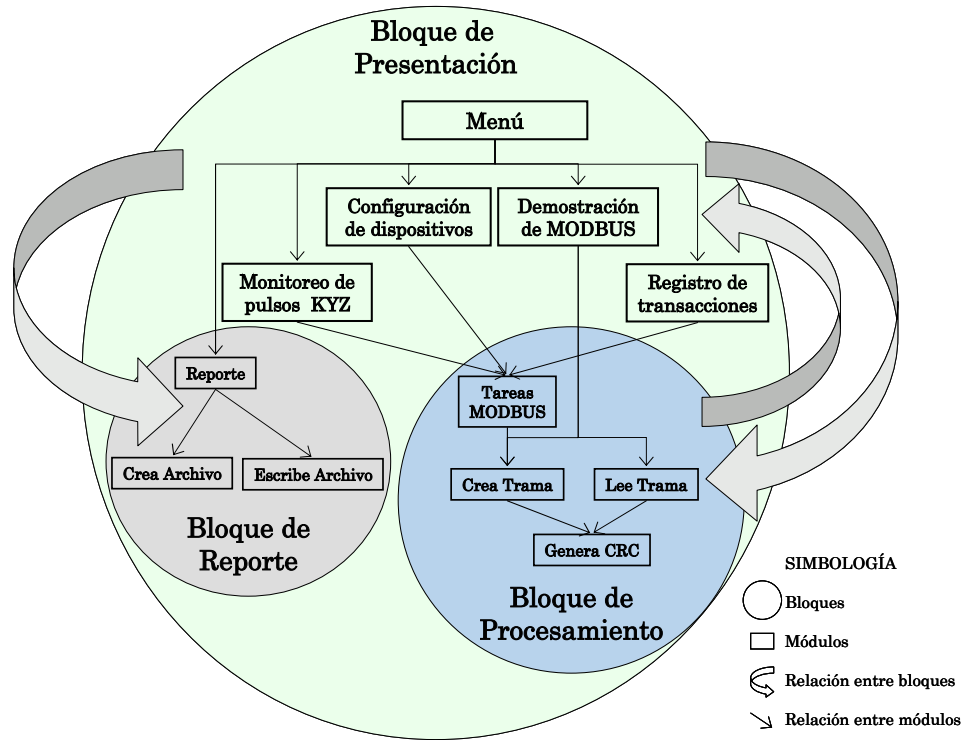


Figura 4.4. Diseño arquitectónico del Soft-KYZ.

4.5.5. Diseño de los componentes

Esta parte del diseño consiste en elaborar una representación gráfica o en texto del programa previa a su codificación. Para tal fin se emplean las técnicas de diseño procedimental que, en el caso de la programación estructurada, son: diagramas de flujo, diagramas Nassi-Schneiderman, tablas de decisión y pseudocódigos [31]. Para diseñar el Soft-KYZ se emplearon las técnicas de los diagramas de flujo y de pseudocódigo. En la fase de codificación se presentan y se describen los diagramas de flujo de los módulos más importantes de este programa.

4.6. Codificación del Soft-KYZ

LabVIEW es un lenguaje que, dentro del paradigma gráfico de programación, emplea la secuencia de flujo de datos para la ejecución de programas, es decir, los bloques de programa sólo se ejecutan cuando todos los datos de entrada que necesitan están disponibles. Aunado a esto, LabVIEW emplea las estructuras clásicas de control de flujo de datos: secuencial, condicional y repetitiva, por lo tanto, también es un lenguaje de programación estructurada.

Por las razones anteriores, el diseño del Soft-KYZ se hizo pensando en un control de flujo estructurado: el diseño arquitectónico cuenta con una estructura jerárquica de control basada en bloques (arquitectura estratificada) y los componentes, o módulos, se diseñaron usando principalmente diagramas de flujo para facilitar su codificación en LabVIEW.

En LabVIEW cada módulo implementado es llamado subVI y se representa por un icono descriptivo de su tarea, al cual se pueden conectar las señales de entrada y obtener de él sus salidas.

A continuación se describen los subVIs más relevantes del Soft-KYZ de acuerdo a la tarea que atienden. Para ello se emplean diagramas de flujo que describen de manera general la secuencia del programa. Los iconos representativos de cada módulo se insertan en la parte donde son invocados, después cada subVI se explica por separado a partir de su icono de entradas y salidas.

4.6.1. Descripción de la tarea Monitoreo de los pulsos KYZ

En la *figura 4.5* se presenta el diagrama de flujo que describe la tarea **Monitoreo de los pulsos KYZ** que el medidor de energía genera y que el CODKYZ codifica y reporta.

En la página *monitoreo* de la GUI, el usuario debe introducir los siguientes parámetros del *monitoreo* y de las transacciones MODBUS:

- La variable *Datos de solicitud*: se diseñó como una estructura numérica que contiene las variables *No. de puerto*, *Dirección de Esclavo* y *Tiempo de espera* porque son los datos importantes de las transacciones MODBUS que se realizan en cada consulta.
- *Duración del monitoreo*: incluye tanto las horas como los minutos.
- *Intervalo de muestreo*: establece cada cuánto se realiza una consulta.
- *Tiempo de archivo*: cada cuánto se actualizan las variables en el archivo.
- *Guardar*: qué variables se incluirán en el archivo de reporte del *monitoreo*.

Después de introducir los datos anteriores, la tarea inicia cuando el usuario activa el control *Iniciar monitoreo*, acto seguido se **Crea el Archivo** en el cual se almacenarán las variables eléctricas calculadas durante el *monitoreo*; después se entra a un ciclo repetitivo que dura el tiempo especificado por el usuario en el que se invoca a la subrutina principal de esta tarea, **Monitoreo**, la cual, a través de la subrutina **Tareas MODBUS**, solicita al CODKYZ los datos de los pulsos KYZ cada vez que se completa un *Intervalo de muestreo*. Con la subrutina **Escribe Respuesta** se extraen los datos de la trama de respuesta MODBUS y con ellos, la subrutina **Calcula Variables** obtiene los valores actuales de las variables de consumo eléctrico que se presentan al usuario y que se han de **Escribir en el Archivo** de reporte. El usuario puede *Detener* este proceso en el momento que lo decida.

Enseguida se explican los subVIs que intervienen en la tarea **Monitoreo de los pulsos KYZ** en el orden en el que aparecen en el diagrama de flujo.

4.6.1.1. Módulo Crea Archivo

En la *figura 4.6* se muestra el icono de entradas y salidas del módulo **Crea Archivo**. La única entrada de este módulo es el arreglo de variables lógicas *Guardar* que indica qué parámetros eléctricos se almacenarán en el archivo de reporte.

Este módulo crea el archivo de reporte del *monitoreo* con las características especificadas en el *apartado 4.5.2*. Al ejecutarse muestra un nombre y una ruta por defecto del archivo de reporte, los cuales pueden ser cambiados por el usuario.

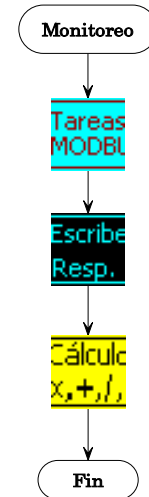
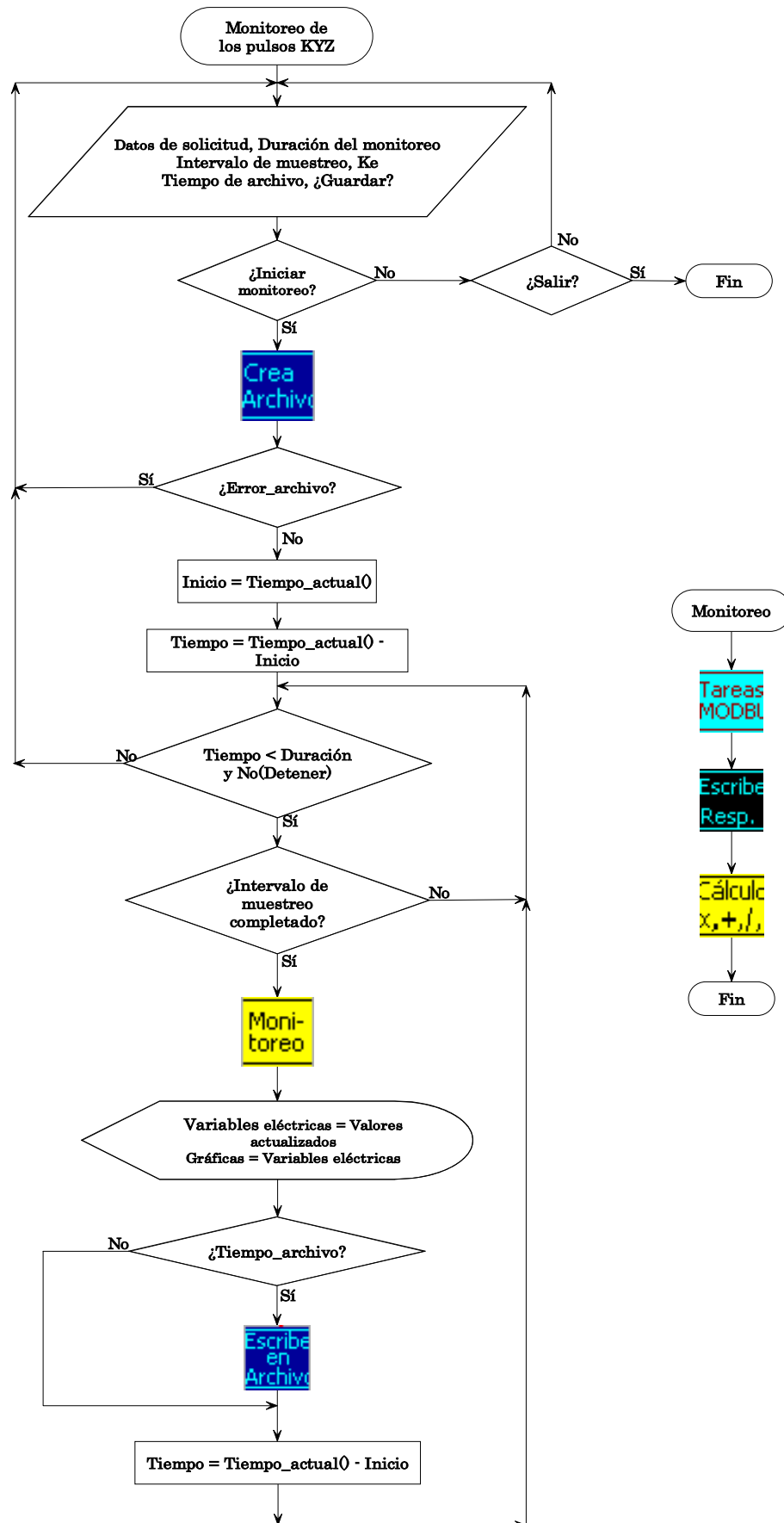


Figura 4.5. Diagrama de flujo de la tarea *Monitoreo* de los pulsos KYZ.



Figura 4.6. Icono de entradas y salidas del módulo Crea Archivo.vi.

Al finalizar, este módulo devuelve:

- Un parámetro de *Referencia* para manipular el archivo.
- Una variable lógica que indica si ocurrió algún *Error* al crear el archivo.
- Una estructura que proporciona información sobre el posible *error* ocurrido (*Error out: Estatus, Código y Fuente de error*) [33].

4.6.1.2. Módulo Monitoreo

En la *figura 4.7* se presenta el icono de entradas y salidas del módulo **Monitoreo**. Este es el módulo principal de la tarea **Monitoreo de los pulsos KYZ** porque recibe las entradas proporcionadas directamente por el usuario:

- La estructura *Datos de solicitud* que contiene la información para realizar las transacciones MODBUS.
- El valor de la *Constante de energía (Ke)* del medidor de la planta analizada.
- La estructura *Tiempo archivo* que indica cada cuánto tiempo se debe guardar el archivo de reporte en disco.

Monitoreo utiliza los *Datos de solicitud* para invocar a la subrutina **Tareas MODBUS**, la cual se encarga de efectuar las transacciones que solicitan los datos de los pulsos KYZ. Una vez que se recibe la trama de respuesta, ejecuta la subrutina **Escribe Respuesta** con la que descifra la trama recibida y obtiene los registros de los datos solicitados. Con esta información y con la *Ke* recibida, invoca la subrutina **Calcula Variables** que efectúa las operaciones matemáticas para obtener los *Valores actualizados* de las variables de consumo eléctrico. Las salidas del módulo **Monitoreo** son:

- Los *Datos* de las variables eléctricas en un arreglo numérico para que sea escrito directamente en el archivo de reporte.
- Una estructura con la *Información* obtenida en la transacción MODBUS: Datos solicitados, función y código de error en caso de excepción.
- Una estructura numérica con los *Valores actualizados* de los parámetros eléctricos calculados: energías activa, reactiva y aparente, FP, demanda medida, demanda máxima y factor de carga.
- Una estructura numérica con los *Valores instantáneos* que se usan para graficar el FP y las potencias activa, reactiva y aparente.
- Una estructura de variables lógicas con información sobre los posibles *Errores* que pudo haber tenido la transacción MODBUS: *Tiempo de espera* expirado, trama recibida con *CRC erróneo* o *Mensaje de excepción* recibido.
- Una variable lógica que indica si es momento de *Escribir* en el archivo de reporte, lo cual se determina por la información de la estructura *Tiempo archivo* recibida como entrada y que indica qué opción de tiempo eligió el usuario para escribir sobre el archivo.

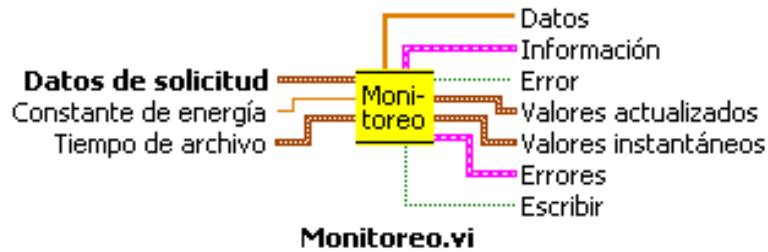


Figura 4.7. Icono de entradas y salidas del módulo *Monitoreo.vi*.

4.6.1.3. Módulo Tareas MODBUS

El icono de la *figura 4.8* muestra los datos de entrada y salida del módulo que construye las tramas de solicitud de las **Tareas MODBUS** más comunes del CODKYZ, también envía las tramas construidas y espera la respuesta del Esclavo por un tiempo especificado por el usuario.

En la *tabla 4.1* se listan las tareas MODBUS más comunes que el Soft-KYZ solicita ejecutar al CODKYZ. Se indica también qué entradas se ocupan para construir cada trama de solicitud.

- *Valores* es un arreglo numérico que contiene los valores de los registros que se desean leer o escribir.
- *Tarea* indica el número de la tarea MODBUS que se va a ejecutar.
- *Dirección* especifica la dirección inicial del bloque de registros mantenidos que se van a leer o escribir.
- *Valor* es el número que se espera recibir en una solicitud de ECO o el valor del único registro que se quiere modificar, *Cantidad* es el número de registros que se van a leer o escribir.

Tabla 4.1. Tareas MODBUS más comunes entre el Soft-KYZ y el CODKYZ.

Tarea	Descripción	Entradas necesarias
0	Leer registros de configuración (<i>mantenidos</i>)	-
1	Leer registros de los pulsos KYZ (<i>entrada</i>)	-
2	Leer datos de identificación del CODKYZ	-
3	Solicitar ECO de la trama enviada	Valor
4	Escribir un único registro de configuración	Dirección, Valor
5	Escribir múltiples registros de configuración	Dirección, Cantidad, Valores
6	Almacenar registros de configuración en la EEPROM	-
7	Reiniciar los registros de los pulsos KYZ	-
8	Reiniciar contadores de eventos de comunicación	-
9	Activar MSE del CODKYZ	-
10	Desactiva MSE del CODKYZ	-

Los datos de salidas son:

- La estructura con la *Información* obtenida en la transacción MODBUS.
- El *Mensaje generado* como solicitud MODBUS.
- El *Mensaje recibido* como respuesta MODBUS.
- La estructura con información sobre posibles *Errores* ocurridos en la transacción.

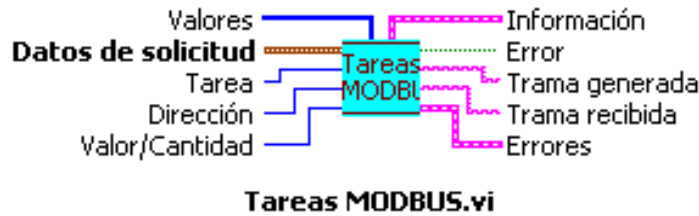


Figura 4.8. Icono de entradas y salidas del módulo Tareas MODBUS.vi.

4.6.1.4. Módulo Escribe Respuesta

Después de haberse efectuado una transacción MODBUS, el módulo **Escribe Respuesta**, cuyo icono de entradas y salidas se muestra en la *figura 4.9*, se encarga de devolver los datos que se han solicitado al CODKYZ. Sus entradas son:

- Un arreglo numérico con los registros que conforman el campo de *Datos de la trama* de respuesta MODBUS.
- La estructura *Datos* que contiene la información de todos los campos de la trama de solicitud.

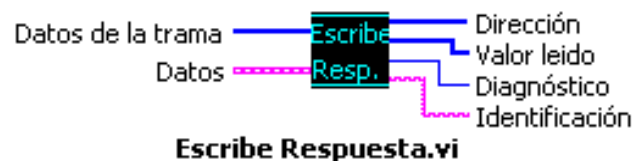


Figura 4.9. Icono de entradas y salidas del módulo Escribe Respuesta.vi.

De acuerdo con la solicitud que se haya realizado, el módulo devuelve:

- El vector *Dirección* contiene las direcciones de los registros solicitados.
- El *Valor* de todos los registros solicitados
- El valor numérico del campo de *Diagnóstico*.
- Una cadena con los datos de *Identificación* del CODKYZ.

4.6.1.5. Módulo Calcula Variables

En el icono de la *figura 4.10* puede apreciarse que el módulo **Calcula Variables** recibe como entradas los *Datos de los pulsos KYZ* y la *Constante de energía (Ke)*, con las cuales realiza los cálculos de los parámetros de consumo eléctrico que el módulo reporta como salida y que son:

- Una estructura numérica con los *Valores actualizados* de las variables eléctricas calculadas y que son las que el usuario elige almacenar en el archivo de reporte: energías activa, reactiva y aparente, FP, demanda medida, demanda máxima y factor de carga.
- Una estructura numérica con los *Valores instantáneos* que se usan para graficar el FP y las potencias activa, reactiva y aparente.

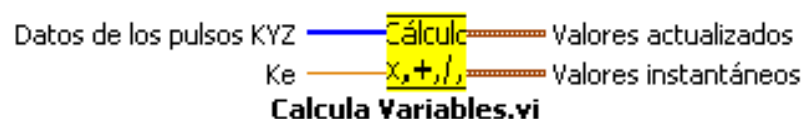


Figura 4.10. Icono de entradas y salidas del módulo Calcula Variables.vi.



Figura 4.11. Icono de entradas y salidas del módulo Escribe Archivo.vi.

4.6.1.6. Módulo Escribe Archivo

El icono de entradas y salidas del módulo **Escribe Archivo** es presentado en la figura 4.11. Este módulo recibe como entradas:

- Una variable lógica que indica si se debe *Escribir* en el archivo de reporte.
- El vector numérico *Datos* de las variables de consumo eléctrico.
- Una estructura de variables lógicas llamada *Guardar*.
- El parámetro de *Referencia* del archivo de reporte.
- Una estructura con datos de algún *Error* antes de manipular el archivo.

Si la variable *Escribir* es verdadera, este módulo escribe el vector *Datos* en el archivo de reporte. *Guardar* indica qué valores del vector serán escritos.

Las salidas del módulo son las variables *Referencia* y *Error* que se recibieron como entradas.

4.6.2. Descripción de la tarea Registro de las transacciones MODBUS

Tanto el Soft-KYZ como el CODKYZ poseen sus propios registros que cuentan eventos de comunicación MODBUS. Para distinguir entre uno y otro, a los registros del Soft-KYZ se les denominará *Datos de transacciones* y a los del CODKYZ, *Contadores de eventos de comunicación*, como lo establece la norma del protocolo MODBUS para un Esclavo. En la tabla 4.2 se listan los *Datos de las transacciones* y en la tabla 3.5 (capítulo 3), los *Contadores de eventos de comunicación*.

De acuerdo con lo ilustrado en el diagrama de flujo de la figura 3.18 (capítulo 3), los *Contadores de eventos de comunicación* se actualizan antes de que el CODKYZ responda; en cambio, los *Datos de las transacciones* se actualizan automáticamente empleando la estructura de variables lógicas *Errores* que se genera después de que una transacción MODBUS finaliza. La figura 4.12 muestra el diagrama de flujo que describe cómo se obtienen los registros contadores actualizados de ambos dispositivos para que sean presentados al usuario.

Tabla 4.2. Datos de las transacciones MODBUS que registra el Soft-KYZ.

Número	Descripción del dato
1	Consultas realizadas
2	Total de respuestas recibidas
3	Respuestas correctas
4	Respuestas de excepción
5	Respuestas con CRC erróneo
6	Consultas sin respuesta
7	Transacciones válidas (%)

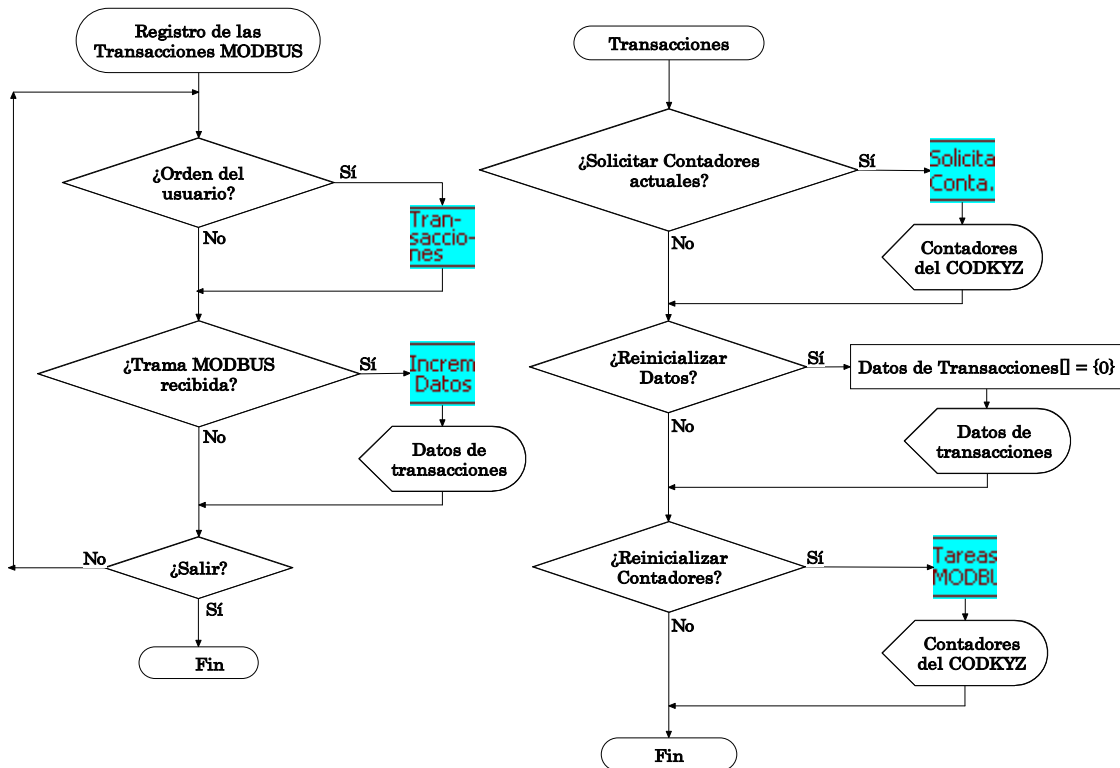


Figura 4.12. Diagrama de flujo de la tarea Registro de las transacciones MODBUS.

En el diagrama de flujo anterior puede observarse que los *Datos de las transacciones* se actualizan automáticamente y se presentan al usuario una vez que se ha recibido una respuesta MODBUS. Esto se realiza siempre que el Soft-KYZ efectúa una transacción y la subrutina **Incrementa Datos** es la encargada de hacerlo.

Transacciones es el módulo principal de esta tarea y se hace cargo de la ejecución de órdenes directas del usuario dadas desde la GUI. Por ejemplo, para que los *Contadores de eventos de comunicación* se presenten actualizados, se deben solicitar al CODKYZ, por lo que el usuario debe dar la orden para hacerlo. De igual manera, la reinicialización de los registros contadores de ambos dispositivos es ordenada por el usuario.

4.6.2.1. Módulo Transacciones

El icono de la *figura 4.13* muestra los datos de entrada y salida del módulo **Transacciones**. Como puede observarse en la figura, las entradas de este módulo son:

- Los *Datos de solicitud* para efectuar una transacción MODBUS.
- Tres entradas lógicas que indican cuál es la orden del usuario que se debe ejecutar: *Reinicializar datos* de las transacciones MODBUS, *Reinicializar contadores* de eventos de comunicación o *Solicitar contadores* actuales.

Además de ejecutar estas órdenes, el módulo lleva el conteo de los *Datos de las transacciones* MODBUS efectuadas entre el CODKYZ y el Soft-KYZ. , también se encarga de solicitar al CODKYZ sus registros *Contadores de eventos de comunicación* y de reinicializar todos estos registros si el usuario lo ordena. La orden *Reinicializar contadores* es ejecutada por la subrutina **Tareas MODBUS**, explicada en el *inciso 4.6.1.3*.

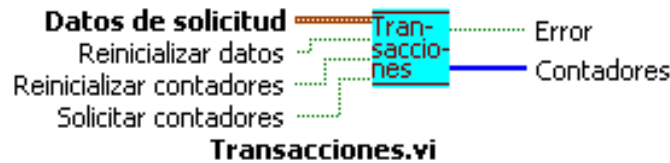


Figura 4.13. Icono de entradas y salidas del módulo Transacciones.vi.

Las salidas del módulo **Transacciones** son:

- La variable lógica *Error* que indica si la orden solicitada no pudo ser ejecutada con éxito.
- Un vector numérico con los valores de los *Contadores de eventos de comunicación* del CODKYZ si han sido solicitados.

4.6.2.2. Módulo Incrementa Datos

En la *figura 4.14* se muestra el icono de entradas y salidas del módulo **Incrementa Datos**. Este módulo es el que actualiza el valor de los *Datos de las transacciones* después de haberse efectuado una transacción MODBUS o de que haya ocurrido un error al intentar efectuarla.

Como entradas recibe:

- El vector *contadores* con los valores anteriores de los *Datos de las transacciones*.
- La estructura de variables lógicas *Errores* que se genera después de que una transacción MODBUS finaliza.

El módulo revisa la estructura *Errores* para verificar si ocurrió uno de los posibles errores de transacción: *Tiempo de espera* expirado, trama recibida con *CRC erróneo* o si se ha recibido un *Mensaje de excepción*. Si no hubo error, la transacción se considera válida, en caso contrario se revisa cuál es el error que se produjo y de ese modo se incrementan los contadores de los *Datos de las transacciones*.

El módulo devuelve el mismo vector de entrada, pero con el *Valor* de los *Datos de las transacciones* actualizado.



Figura 4.14. Icono de entradas y salidas del módulo Incrementa Datos.vi.

4.6.2.3. Módulo Solicita Contadores

El módulo **Solicita Contadores** se ejecuta como respuesta a una orden del usuario, en la *figura 4.15* se muestra su icono de entradas y salidas. La única entrada que necesita para efectuar las transacciones MODBUS de diagnóstico es la estructura *Datos de solicitud*.

El módulo se encarga de solicitar los *Contadores de eventos comunicación* uno por uno como lo establece el protocolo, por lo tanto, construye una trama MODBUS de solicitud por cada contador. Sus salidas son idénticas al módulo **Transacciones**.



Figura 4.15. Icono de entradas y salidas del módulo Solicita Contadores.vi.

4.6.3. Descripción de la tarea Configuración de los dispositivos

En el caso del software, lo que se configura es el puerto serial de la computadora, manualmente, con datos proporcionados por el usuario; o de manera automática, detectando la velocidad del CODKYZ en un puerto especificado por el usuario. Para configurar el CODKYZ es necesario solicitar, modificar y almacenar en EEPROM sus registros de configuración (*registros mantenidos*).

Con esta tarea también puede ponerse al CODKYZ en el *modo sólo escuchar* para deshabilitarlo de la red MODBUS. El diagrama de flujo de la figura 4.16 describe el proceso mediante el cual se configura tanto el software de pruebas como el CODKYZ.

Configuración es el módulo principal de la tarea **Configuración de los dispositivos** y se encarga de efectuar las operaciones de configuración del Soft-KYZ y del CODKYZ a través de la interacción del usuario, quien proporciona los parámetros que el módulo necesita para llevar a cabo la *Tarea* de configuración elegida y después se le presentan los resultados.

A continuación se describe el módulo **Configuración** y los otros módulos importantes que atienden esta tarea.

4.6.3.1. Módulo Configuración

La figura 4.17 muestra el icono de entradas y salidas del módulo encargado de realizar las **Configuración** del Soft-KYZ y del CODKYZ.

Las entradas de este módulo son:

- Los *Datos de solicitud* para efectuar una transacción MODBUS.
- Los *Datos de configuración del puerto serie* (si se desea configurar).
- Los *Valores de los registros* que se van a modificar.
- El número de *Tarea* de configuración que se va a realizar.

De acuerdo con la *Tarea* de configuración elegida, el módulo ejecuta una subrutina específica para realizar la operación correspondiente y los valores que necesita los obtiene de los parámetros proporcionados por el usuario. Por ejemplo, si el usuario ha elegido configurar el puerto automáticamente, el módulo invoca a la subrutina **Detecta CODKYZ** y le cede su parámetro de entrada *Datos de solicitud* para verificar si el CODKYZ está conectado al puerto especificado.

Como puede apreciarse en el diagrama de flujo de la figura 4.16, la mayoría de opciones de configuración son atendidas por la subrutina **Tareas MODBUS** explicada en el inciso 4.6.1.3, sólo que aquí el parámetro *Tarea* cambia de acuerdo con la opción de configuración elegida por el usuario (tabla 4.1).

Las salidas del módulo **Configuración** son:

- La variable lógica *Error* que indica si la orden solicitada no pudo ser ejecutada con éxito.
- Una estructura que contiene los datos de *Identificación* del CODKYZ que se devuelve después de su detección.
- Una estructura numérica que contiene los *Valores actuales* de todos los parámetros de configuración actual del CODKYZ si éstos han sido solicitados.

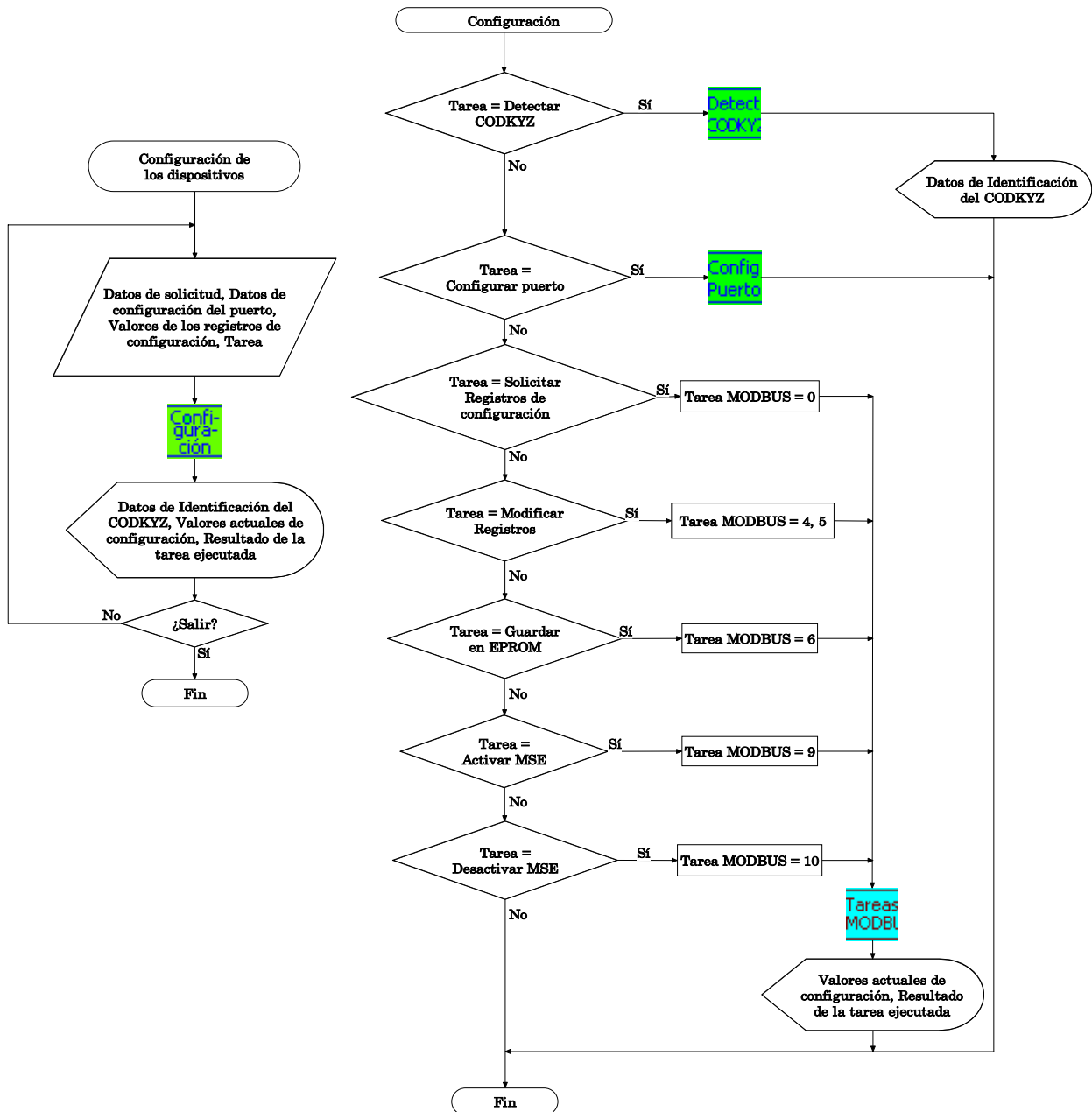


Figura 4.16. Diagrama de flujo de la tarea Configuración de los dispositivos.



Figura 4.17. Icono de entradas y salidas del módulo Tareas de Configuración.vi.

4.6.3.2. Módulo Detecta CODKYZ

Este es el módulo encargado de verificar si el CODKYZ está conectado a un puerto especificado por el usuario, su icono de entradas y salidas se muestran en la figura 4.18.

La única entrada que el módulo necesita es la variable *Datos de solicitud*, ya que este módulo envía tramas MODBUS a diferentes velocidades solicitando un eco del dato enviado. El proceso finaliza cuando el CODKYZ responde exitosamente a una trama o cuando ya se han probado todas las velocidades soportadas por el CODKYZ y no se recibió ninguna respuesta. Las salidas de este módulo son:

- La variable lógica *Error* que indica si la detección se llevó a cabo exitosamente o no.
- Una estructura que contiene los datos de *Identificación* del CODKYZ si la detección fue exitosa.



Figura 4.18. Icono de entradas y salidas del módulo Detecta CODKYZ.vi.

4.6.3.3. Módulo Configura Puerto

En la figura 4.19 se muestra el icono de entradas y salidas del módulo que *Configura el Puerto* serie de la PC. La estructura de entrada *Parámetros de configuración* contiene los datos: *Número de puerto*, *Velocidad*, *Bits de datos*, *Paridad* y *Bits de fin* con los que se inicializa el puerto serie.

La única salida del módulo es la variable lógica *Error* que indica si la inicialización del puerto se llevó a cabo exitosamente o no.



Figura 4.19. Icono de entradas y salidas del módulo Configura Puerto.vi.

4.6.4. Descripción de la tarea Demostración del protocolo MODBUS

Para construir una red MODBUS, como mínimo deben existir un dispositivo que funja como Maestro y otro como Esclavo. Con esta tarea, se hace una demostración del funcionamiento del protocolo MODBUS empleando el Soft-KYZ como Maestro de la red y al CODKYZ como Esclavo.

La demostración abarca desde la construcción de mensajes de solicitud, hasta la recepción exitosa de mensajes de respuesta o la visualización e interpretación de errores en las transacciones efectuadas.

La *figura 4.20* muestra el diagrama de flujo de la tarea **Demostración del protocolo MODBUS**. Con esta tarea se ilustra el funcionamiento del protocolo de comunicaciones MODBUS (*capítulo 2*) en interacción directa con el usuario desde la GUI del Soft-KYZ.

MODBUS es el módulo principal de esta tarea y antes de que se ejecute, es necesario que el usuario proporcione los valores de los campos del mensaje de solicitud por construir. Todos los campos pertenecientes al mensaje MODBUS se introducen en una estructura de datos que sirve como entrada al módulo. Una vez hecho lo anterior, el módulo **MODBUS**, explicado a continuación, se encarga de lo subsiguiente.

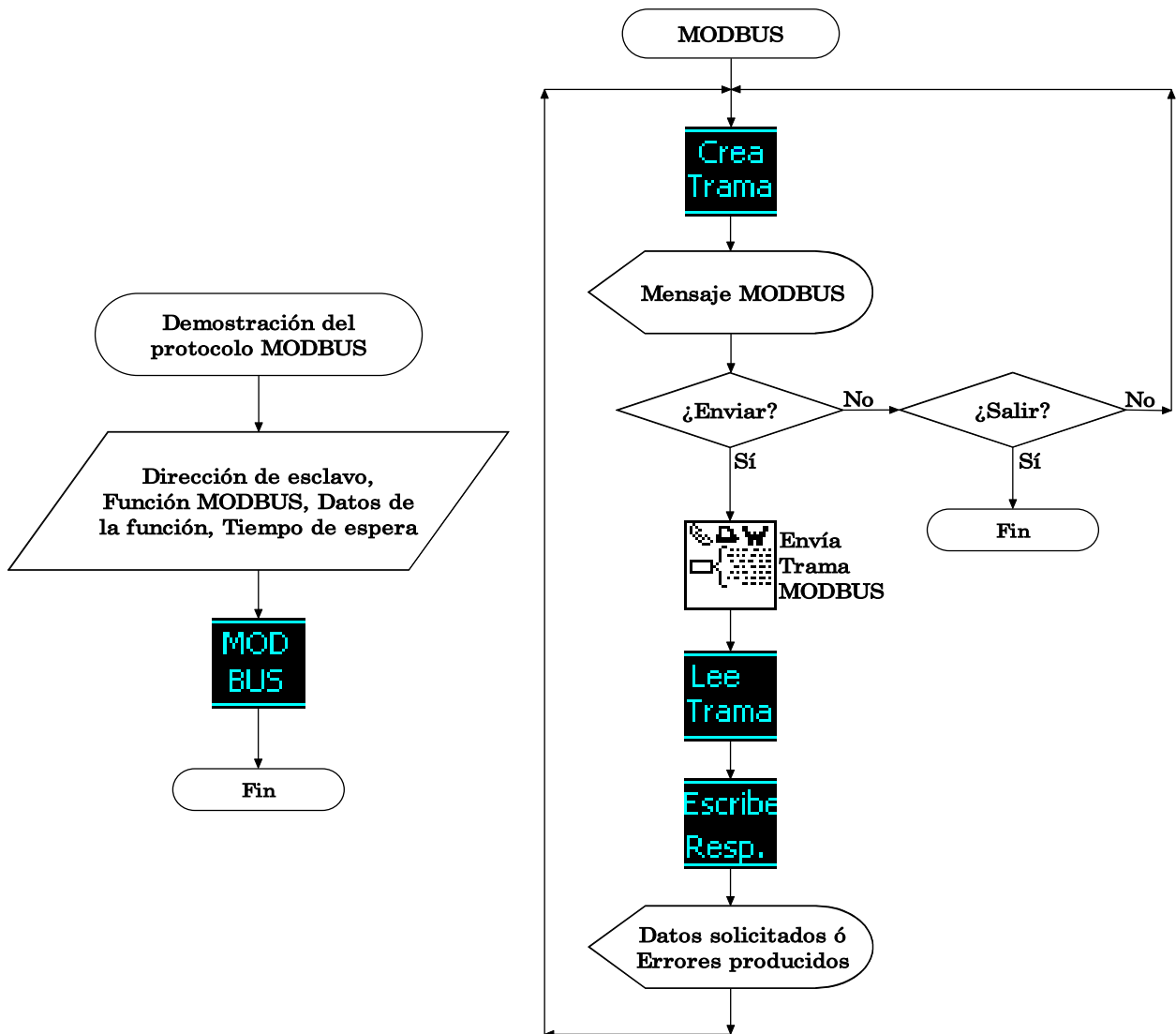


Figura 4.20. Diagrama de flujo de la tarea Demostración del protocolo MODBUS.

4.6.4.1. Módulo MODBUS

La *figura 4.21* muestra el icono de entradas y salidas del módulo **MODBUS**, el cual recibe como datos de entrada:

- El número de *Puerto* con el que se establece la comunicación serial.
- La estructura *Datos* que contiene los campos del mensaje.
- El *Tiempo de espera* de un mensaje de respuesta
- Una variable lógica que indica si el paquete construido se debe *Enviar* o no.

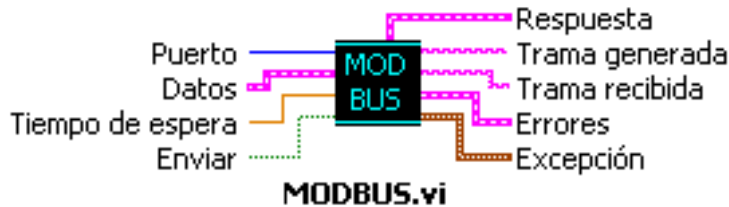


Figura 4.21. Icono de entradas y salidas del módulo MODBUS.vi.

El proceso es el siguiente: con los campos proporcionados, la subrutina **Crea Trama MODBUS** se encarga de construir un paquete numérico parcial, el cual se ocupa para calcular el CRC-16 del mensaje de solicitud MODBUS final y que es el que se presenta al usuario. Este mensaje MODBUS se envía al CODKYZ sólo si la entrada lógica *Enviar* así lo indica, de lo contrario permanece en pantalla hasta que el usuario decide enviarlo o cambiarlo modificando el valor de sus campos.

Cuando el usuario ordena enviar el mensaje MODBUS por el puerto serie, después de hacerlo, el programa invoca a la subrutina **Lee Trama** para esperar la respuesta hasta que expira el *Tiempo de espera*. Si antes de ese tiempo se recibe un mensaje de respuesta, se descifra y se verifica que sea correcto. De ser así, el módulo **Escribe Respuesta** devuelve los datos solicitados en una estructura de *Respuesta* tomando en cuenta los campos del mensaje de solicitud inicial, en caso contrario se determina la causa del error y se reporta al usuario.

Las salidas proporcionadas por el módulo **MODBUS** son:

- La estructura *Respuesta* que contiene los datos solicitados por el usuario.
- La *Trama generada* como mensaje de solicitud.
- La *Trama recibida* como mensaje de respuesta.
- La estructura de variables lógicas que indican los posibles *Errores* en la transacción MODBUS.

4.6.5. Estructura jerárquica de control del Soft-KYZ

En la *figura 4.22* se muestra la estructura jerárquica de control final del SoftKYZ construida con los iconos de entrada y salida de los subVIS explicados anteriormente. Esta estructura se basa en el diagrama de *diseño arquitectónico* de la *figura 4.4*, el cual se propuso como modelo inicial en la fase de diseño para desarrollar el Soft-KYZ.

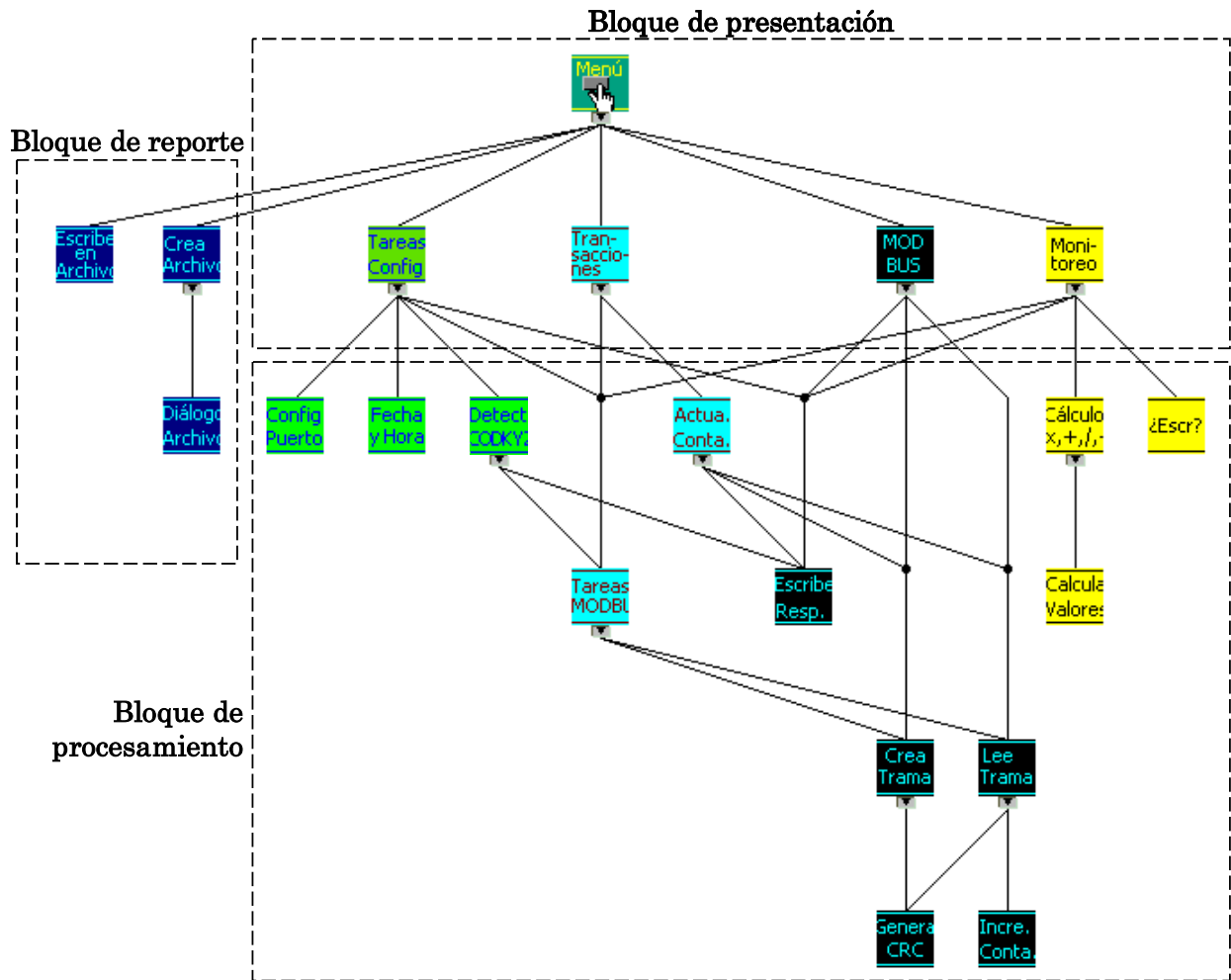


Figura 4.22. Estructura jerárquica de control del Soft-KYZ.

4.7. Pruebas del Soft-KYZ

LabVIEW ofrece la ventaja de ejecutar y probar las subrutinas independientemente de que sean invocadas o no por el programa principal. Con este enfoque modular de programación, las pruebas del Soft-KYZ se realizaron empleando las técnicas de caja blanca y caja negra siguiendo una integración ascendente de sus módulos en búsqueda de errores de ejecución. Además, se hicieron pruebas de caja negra al Soft-KYZ como producto final, verificando su funcionamiento y la usabilidad de su GUI.

4.7.1. Pruebas de ejecución e integración de módulos

La integración ascendente se refiere a realizar las pruebas del software empezando con los últimos módulos de la estructura jerárquica de control hasta llegar al primer nivel [31]. Para ello es necesario emplear la técnica de caja blanca, en la cual se revisa la ejecución del módulo paso a paso observando el flujo de los datos con la finalidad de encontrar errores y analizar el comportamiento del módulo en cuestión.

Cuando un módulo contiene otro módulo, se asciende de nivel y con ello, se puede aplicar la técnica de caja negra, es decir, mientras un módulo se analiza con la técnica de caja blanca, se verifica que cuando los datos de entrada de un módulo subordinado son correctos, dicho módulo proporcione las salidas que se esperan o que ejecute su tarea de manera adecuada.

Estas pruebas se realizaron con todos los módulos del sistema respetando niveles de jerarquía hasta finalizar. En la *figura 4.23* se muestra un ejemplo de las pruebas de caja blanca y caja negra de manera simultánea en las subrutinas *Crea Trama MODBUS* y *Genera CRC* utilizando las herramientas *highlight* y *probe* de LabVIEW [33].

4.7.2. Pruebas de funcionamiento

Estas pruebas se realizaron directamente empleando el CODKYZ como elemento del sistema. Se comprobó que todas las tareas implementadas funcionaran correctamente y proporcionaran los resultados adecuados, para ello se aplicó la técnica de caja negra tanto al Soft-KYZ como al CODKYZ, tratando a ambos como un producto terminal que proporciona las salidas esperadas siguiendo las órdenes del usuario. En el *capítulo 5* se describen las pruebas hechas a todo el sistema en conjunto.

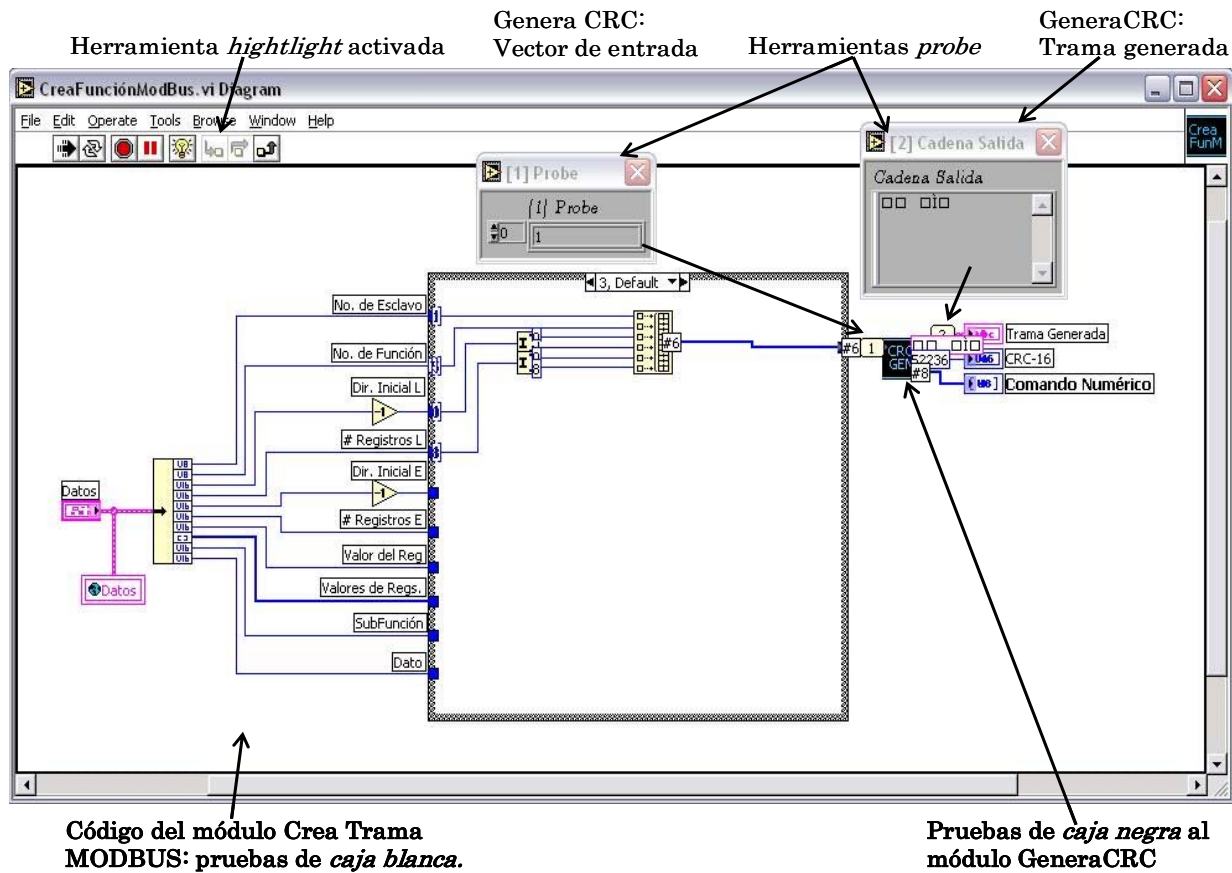


Figura 4.23. Pruebas de caja blanca y caja negra ejecutadas en dos módulos del Soft-KYZ.

4.7.3. Pruebas de usabilidad

Aunque el uso del Soft-KYZ está dirigido a personas relacionadas con el estudio de la electricidad, para las pruebas de usabilidad se solicitó la ayuda de 5 estudiantes de Ingeniería en Electrónica y 5 de Ingeniería en Computación para que evaluaran su funcionamiento. Además, se contó con las revisiones del M. en C. Mario A. Moreno Rocha y del Dr. Christian Sturm, ambos especialistas en usabilidad de software [34, URL13], quienes dieron el visto bueno y algunas recomendaciones para mejorar la usabilidad de la GUI del Soft-KYZ presentada en la *figura 4.24*.



Figura 4.24. Interfaz gráfica de usuario del Soft-KYZ.

5. Pruebas y resultados

En este capítulo se describen las pruebas realizadas al sistema completo, el cual está constituido por el codificador de pulsos KYZ (CODKYZ) y su software de configuración y pruebas (Soft-KYZ).

De acuerdo a su desarrollo, las pruebas del sistema se dividieron en 2 fases principales:

- *Pruebas de laboratorio:* se hicieron en un ambiente de bajo ruido para probar todas las características de comunicación y registro de pulsos del CODKYZ. Esto permitió identificar y corregir algunos problemas de funcionamiento antes de instalar el CODKYZ en campo. Los pulsos KYZ que producen los medidores electrónicos se emularon con un sistema electrónico que se construyó empleando 3 relevadores de CD a la salida, los cuales conmutaban a un tiempo fijo preestablecido. En estas pruebas, la conexión entre la PC y el CODKYZ se hizo de manera *directa* a través del puerto RS-232 con un cable de 2 m de longitud; y *a distancia*, usando el puerto RS-485 del CODKYZ y un convertidor RS-232 a RS-485 para la PC con un carrete de 250 m de cable de *par trenzado* entre ellos.
- *Pruebas de campo:* se instaló el medidor de variables eléctricas PM650 en la subestación eléctrica de la UTM y su único relevador KYZ se conectó al CODKYZ para registrar pulsos reales de energía activa. Los pulsos de energía reactiva y sincronía se tomaron del sistema emulador. Las comunicaciones MODBUS entre el CODKYZ (instalado en campo) y el Soft-KYZ (instalado en una PC de laboratorio) se establecieron de manera *inalámbrica* empleando un par de radio-módems de 900MHz de la firma MAXSTREAM y dos antenas externas.

En todos los casos, el Soft-KYZ fungió como el nodo Maestro con el que se configuró y se *monitoreó* el Esclavo CODKYZ usando exclusivamente mensajes del protocolo de comunicaciones MODBUS. A continuación se explica el procedimiento para efectuar las pruebas citadas anteriormente; posteriormente se proporcionan los detalles de las instalaciones hechas en laboratorio y en campo; y finalmente se presentan los resultados más importantes que se obtuvieron en cada una de ellas.

5.1. Procedimiento de pruebas

Las pruebas del sistema completo se efectuaron en el siguiente orden desde la GUI del Soft-KYZ:

- Detección y configuración del CODKYZ.
- Comprobación de que el CODKYZ registra los pulsos KYZ de forma correcta.
- *Monitoreo* de los pulsos KYZ generando un archivo de reporte de las variables eléctricas calculadas para crear gráficas en Microsoft Excel y comprobar que coinciden con los valores esperados.
- Revisión de los *Datos de las transacciones* MODBUS efectuadas por el Soft-KYZ para conocer y registrar el porcentaje de transacciones válidas. Estos datos también se incluyen en el archivo de reporte al finalizar el *monitoreo*.
- Documentación de resultados.

El diagrama de flujo del procedimiento anterior se ilustra en la *figura 5.1*.

Para el caso de los problemas que se mencionan en el diagrama de flujo, en el apéndice B se menciona cómo identificar y resolver los problemas más comunes que se presentan a la hora de trabajar con el hardware operado desde el software de pruebas.

Los pasos del procedimiento anterior se siguieron tanto para las pruebas de laboratorio como para las de campo. En seguida se dan los detalles de cada una de ellas.

5.2. Pruebas de laboratorio

Debido a que el CODKYZ está destinado a trabajar en un ambiente con intenso ruido electromagnético, uno de los objetivos de las pruebas de laboratorio fue comprobar que este dispositivo funciona correctamente en un ambiente con condiciones normales de interferencia electromagnética (bajo ruido). Esto daría la pauta para saber si el campo electromagnético afecta el funcionamiento de las comunicaciones con MODBUS, si es que se presentaba algún problema al trabajar bajo un ambiente ruidoso.

Asimismo se verificó que, de acuerdo a la razón de pulsos programada en el sistema emulador de relevadores KYZ, el software de pruebas reportara los valores correctos de las variables de consumo eléctrico.

La comunicación serial entre el hardware y el software se hicieron primero usando el puerto RS-232 y después, el puerto RS-485. La conexión del sistema y el procedimiento de las pruebas se explican a continuación.

5.2.1. Instalación del sistema para las pruebas de laboratorio

En el diagrama a bloques de la *figura 5.2* se ilustra la conexión de los componentes del sistema para realizar las pruebas de laboratorio. Las *figuras 5.3 y 5.4* muestran el aspecto de la instalación usando los dos puertos seriales.

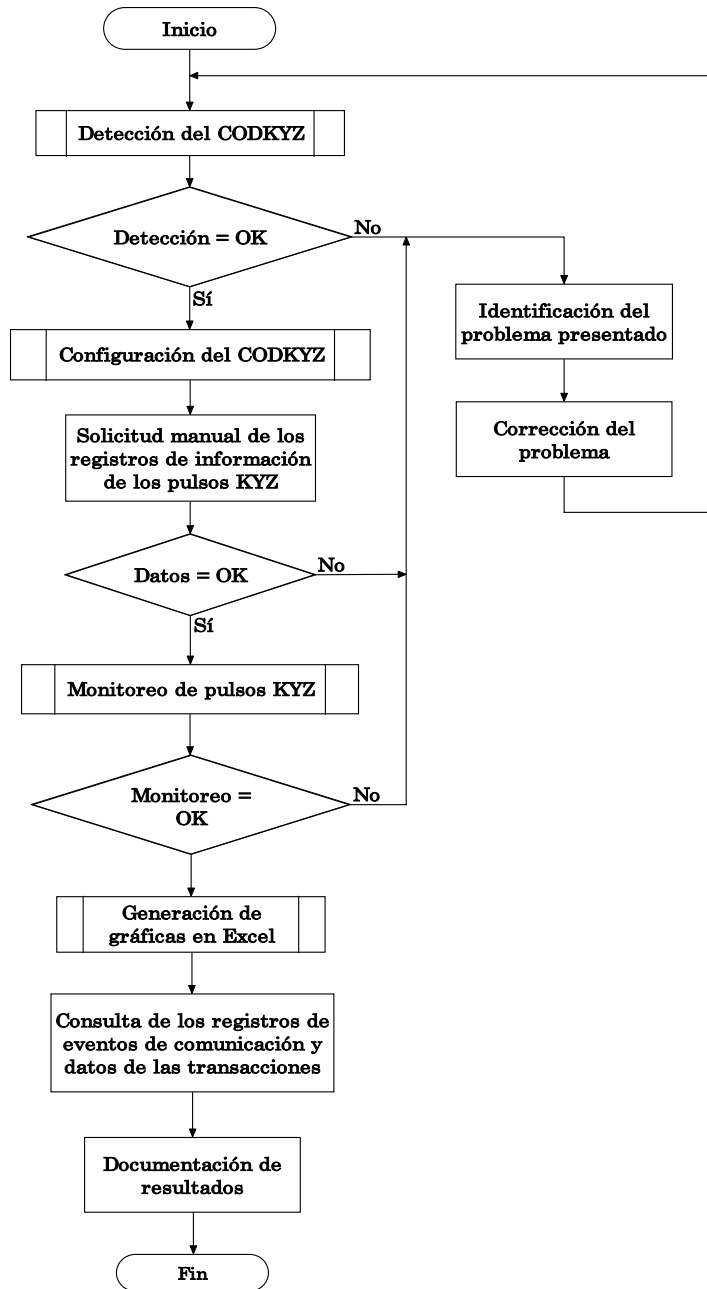


Figura 5.1. Diagrama de flujo del procedimiento de pruebas.

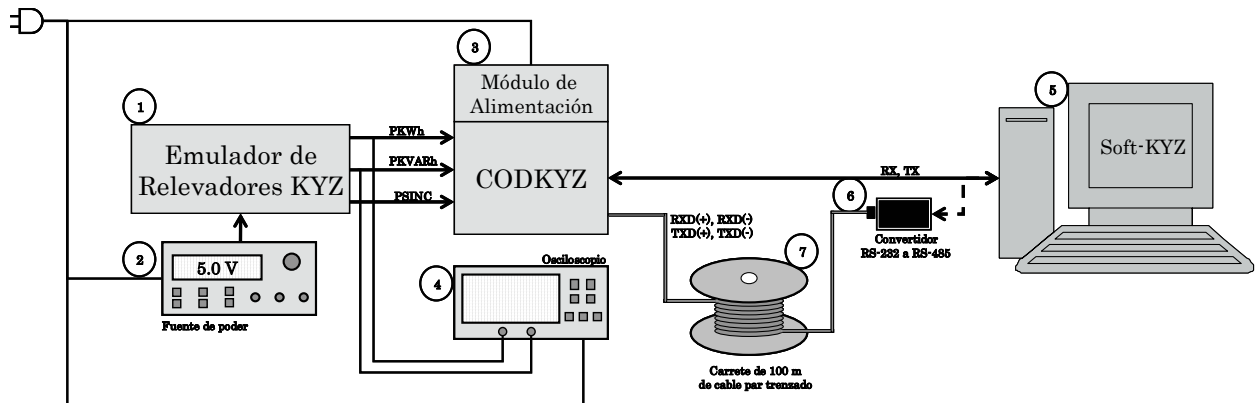


Figura 5.2. Diagrama de instalación del sistema para las pruebas de laboratorio.



Figura 5.3. Instalación del sistema para las pruebas de laboratorio usando el puerto RS-232.

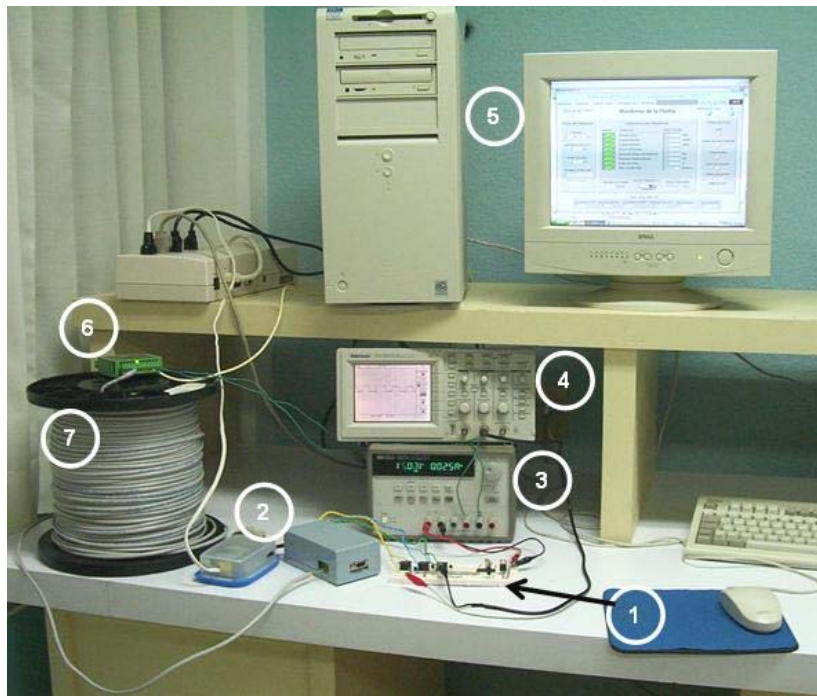


Figura 5.4. Instalación del sistema para las pruebas de laboratorio usando el puerto RS-485.

4.1.1.1. Elementos de las pruebas de laboratorio

Considerando las figuras anteriores, los elementos que integraron el sistema de pruebas de laboratorio son:

1. Emulador de relevadores KYZ que generaba pulsos en tiempos constantes.
2. El CODKYZ con su módulo de alimentación por separado.
3. Una fuente de poder de laboratorio para alimentar el emulador.

4. Un osciloscopio con el que se verificó de manera visual el momento en que ocurrían los pulsos KYZ emulados y su comportamiento (rebotes).
5. El Soft-KYZ que fungió como el Maestro de la red MODBUS.
6. Un convertidor RS-232 a RS-485 para la PC que se usó en las pruebas con *conexión alamburada a distancia*.
7. Un carrete de 250 metros de cable par trenzado para simular la conexión a distancia.

5.2.1.1.1. Emulador de relevadores KYZ

Debido a que en el laboratorio no se puede instalar un medidor de energía que genere los pulsos proporcionales al consumo de energía de una planta, fue necesario diseñar un circuito que los emulara. A este circuito se le llamó *emulador de pulsos KYZ* y se construyó con un microcontrolador AT90S2313 y 3 relevadores de CD. La *figura 5.5a* ilustra su diagrama esquemático en el que sólo aparece la conexión del relevador que emula los pulsos KYZ de energía activa. En *figura 5.5b* muestra el emulador con los tres relevadores que necesita el sistema.

El emulador, se construyó con las siguientes características:

- Se programó un microcontrolador para que generara las señales de activación/desactivación (PEA, PER y PEAp de la *figura 5.5a*) cada tiempo específico para cada relevador y, de esa manera, se emularan los pulsos KYZ correspondientes. Los tiempos de conmutación de cada relevador se programaron de la siguiente manera:

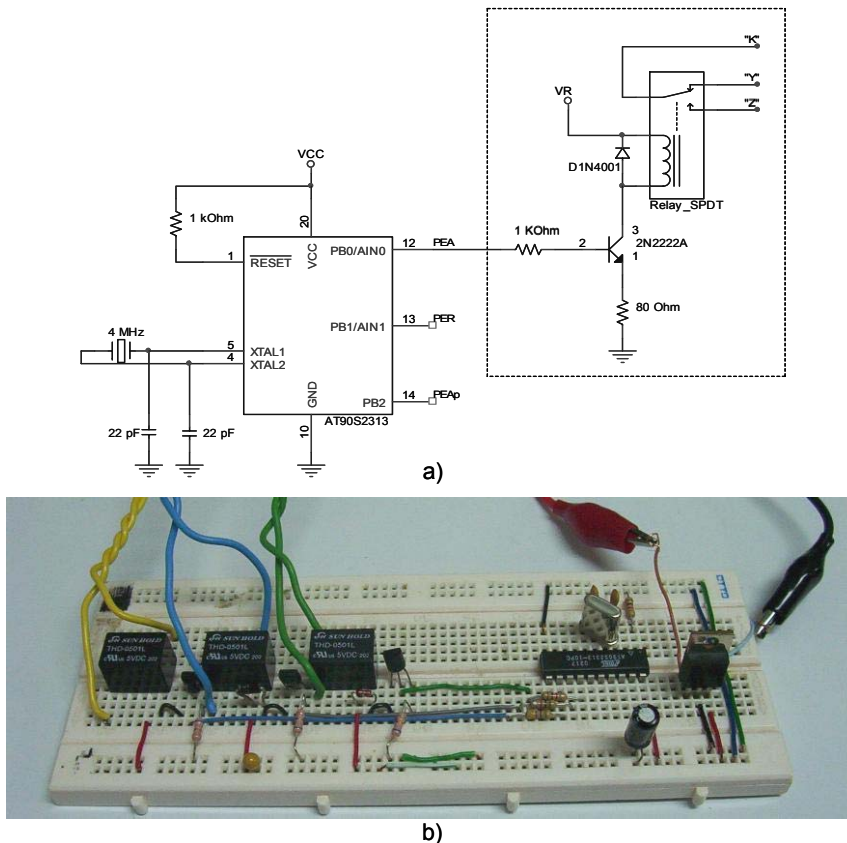


Figura 5.5. a) Diagrama esquemático y b) fotografía del emulador de relevadores KYZ.

- Pulsos KYZ de energía activa (PKWh): cada 3 segundos.
- Pulsos KYZ de energía reactiva (PKVARh): cada 9.8 segundos.
- Pulsos KYZ de sincronía (PSINC): cada 5 minutos (valor fijo de acuerdo al *intervalo de demanda* de los medidores de energía de CFE).
- La etapa de activación/desactivación de los relevadores se hizo con transistores operando en modo corte/saturación y se usó un diodo como supresor de picos de voltaje en cada relevador.

5.2.2. Resultados de las pruebas de laboratorio

En este apartado se presentan los resultados finales que se obtuvieron al seguir los pasos del procedimiento de pruebas sobre el sistema completo en el laboratorio y más adelante, en el *apartado 5.2.3*, se mencionan los principales problemas que se tuvieron mientras se realizaban estas pruebas.

5.2.2.1. Resultados empleando el puerto serial RS-232

Los resultados mostrados a continuación, corresponden a las pruebas que se hicieron usando una conexión directa entre el CODKYZ y el Soft-KYZ por puerto serie RS-232 siguiendo los pasos del procedimiento presentado en el *subcapítulo 5.1*.

5.2.2.1.1. Detección y configuración del CODKYZ

La *figura 5.6a* muestra el momento en que el Soft-KYZ envía el mensaje “*Detección realizada con éxito*” informando al usuario que el CODKYZ ya ha sido detectado. La *figura 5.6b* presenta la página *Configuración* del Soft-KYZ con los datos de identificación y configuración reportados por el CODKYZ después de haberse ejecutado su detección.

5.2.2.1.2. Comprobación del registro de los Pulsos KYZ

Para comprobar que el CODKYZ registra correctamente los pulsos KYZ, se hizo funcionar el emulador de relevadores durante un tiempo determinado (330 segundos) y después se consultaron los registros de entrada del CODKYZ para verificar que coincidieran con los valores esperados. El intervalo de medición usado para cuantificar los pulsos KYZ en esta prueba fue de 1 segundo.

La *figura 5.7* presenta la página *MODBUS* con el resultado obtenido después de una transacción manual solicitando los *registros de entrada* (datos de cuantificación de los pulsos KYZ) del CODKYZ. La *tabla 5.1* contiene los cálculos y los valores de los pulsos esperados en estas pruebas.

Tabla 5.1. Valores de registros de pulsos KYZ esperados y los reportados por el CODKYZ.

Pulso	Valor ideal esperado	Valor reportado	Diferencia	% Error
PKWh	$(330 \text{ s}) / (3 \text{ s/pulso}) = 110 \text{ pulsos}$	109 pulsos	1 pulso	0.91
T-PKWh	$110 \text{ pulsos} \times 3 \text{ s/pulso} = 330 \text{ s}$	228 s	2 s	0.61
PKVARh	$(330 \text{ s}) / (9.8 \text{ s/pulso}) = 33.67 \text{ pulsos}$	33 pulsos	0 pulsos	0
T-PKVARh	$33.67 \text{ pulsos} \times 9.8 \text{ s/pulso} = 323.4$	324 s	1 s	0.31

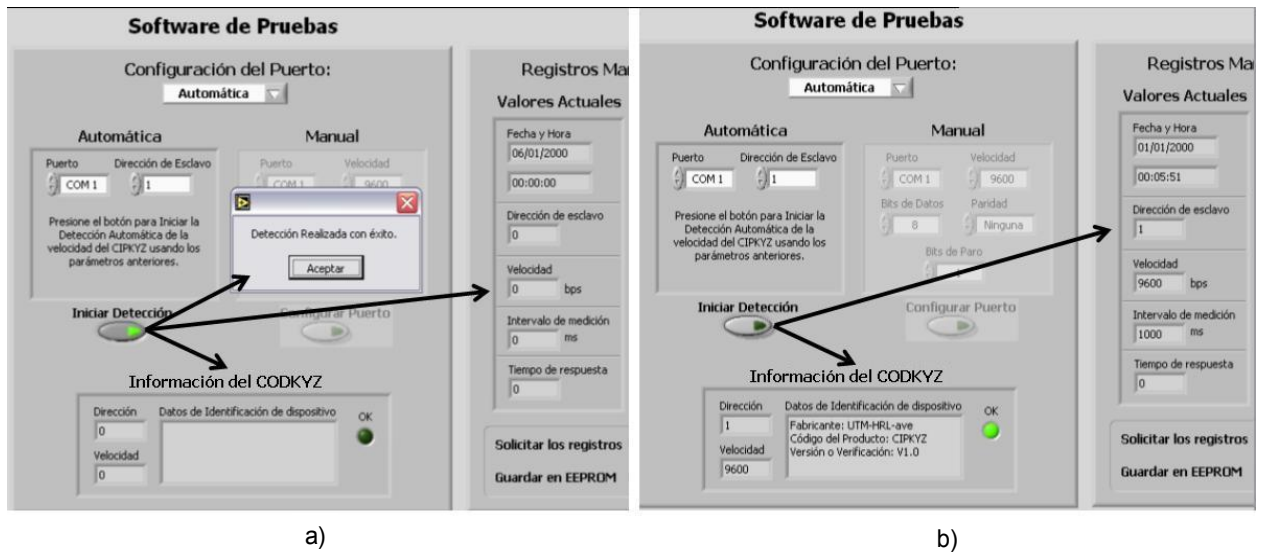


Figura 5.6. a) Detección y b) reporte de datos de identificación del CODKYZ.

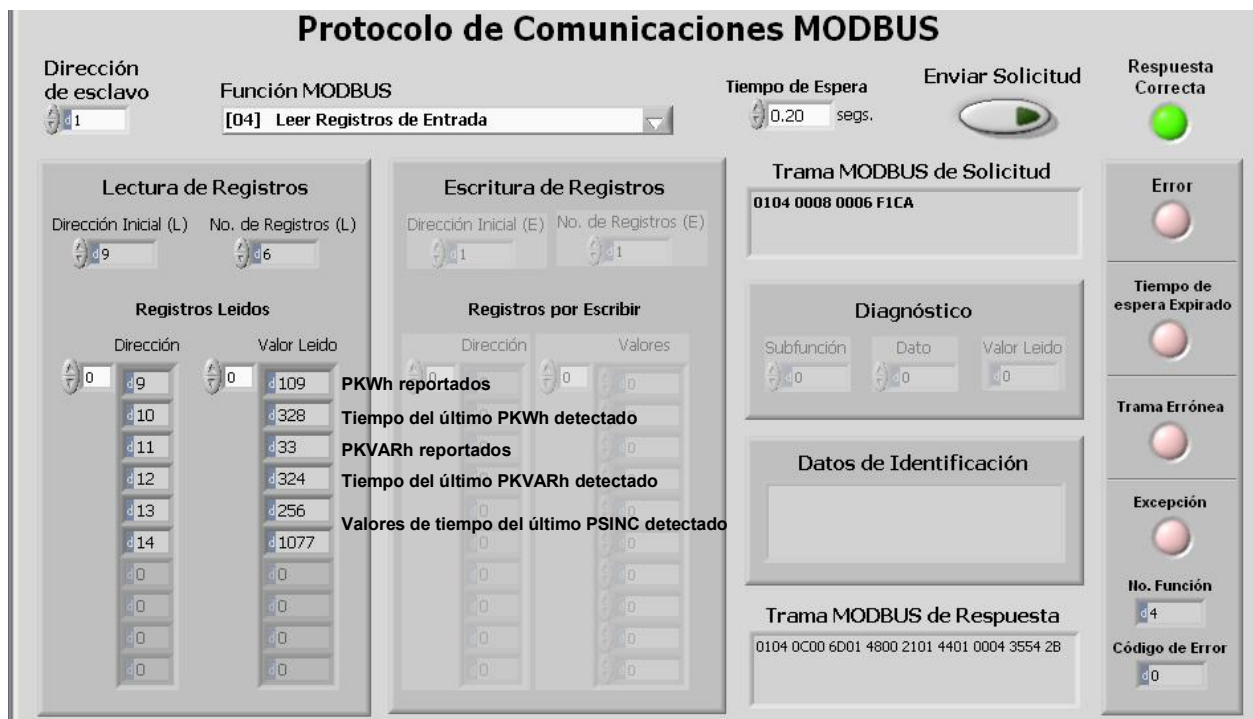


Figura 5.7. Datos de cuantificación de los pulsos KYZ reportados por el CODKYZ.

Cabe aclarar que el emulador y el CODKYZ no fueron encendidos en el mismo instante y que las diferencias presentadas entre los valores esperados y los valores reportados por el CODKYZ, son pequeñas. Por ejemplo, analizando la figura 5.7 puede apreciarse que el pulso 110 de energía activa que se esperaba como resultado ideal, ocurriría en el segundo 331 (3 segundos después del valor 328 reportado).

Por lo tanto, tomando en cuenta que las razones constantes de pulsos programadas en el emulador no eran exactas (3 segundos en el caso de los PKWh) y que el tiempo que el CODKYZ permaneció detectando estos pulsos no fueron exactamente 330 segundos, se concluye que la cantidad y el tiempo de cuantificación de pulsos KYZ que el CODKYZ reporta en estas pruebas son los correctos.

5.2.2.1.3. Monitoreo de pulsos KYZ

Esta prueba se hizo con la finalidad de comprobar que las gráficas de las variables de consumo eléctrico, obtenidas como resultado del *monitoreo*, coincidieran con los valores esperados de acuerdo con la razón de pulsos producida por cada relevador del emulador.

En este caso, la constante de energía K_e del medidor se consideró con un valor de 0.1 KWh/pulso y el intervalo de medición de los pulsos, con un valor de 100 ms.

Puesto que la razón de pulsos KYZ del circuito emulador es constante (1 pulso cada 3 segundos), para estas pruebas se esperaba que las gráficas resultantes de potencia fueran líneas rectas de valor constante (pendiente 0) y que las gráficas de energía fueran unas líneas rectas con pendiente diferente de 0 porque el consumo de energía se acumula e incrementa también de manera constante. Empleando las ecuaciones del *capítulo 1*, en la *tabla 5.2* se calculan y se listan los valores instantáneos que se esperaban para las potencias eléctricas durante todo el *monitoreo* y los valores totales para las energías al final del *monitoreo*.

En la *figura 5.8* se muestra la página *Monitoreo* del Soft-KYZ exhibiendo los datos del *monitoreo* y los resultados finales de esta prueba. Los datos del *monitoreo* son: *Duración* (1 hora y 0 minutos), *Intervalo de muestreo* (10 segundos) y *Tiempo de espera* (1.5 segundos). Los resultados de los parámetros más importantes en esta prueba son: Energía Activa, 120.3 KWh; Energía Reactiva, 36.8 KVARh; y Energía Aparente, 125.98 KVAh, los cuales son valores muy cercanos a los esperados idealmente. La comparación de estos valores se presenta en la *tabla 5.2*.

En la *figura 5.9* se presentan las gráficas de potencias instantáneas reportadas por el Soft-KYZ durante el *monitoreo*. Los valores que aparecen en la figura son: 118.67 KW de potencia activa, 36.36 KVAR de potencia reactiva y 124.12 KVA de potencia aparente. Estos valores se comparan con los esperados en la *tabla 5.2*.

Datos del Monitoreo		Parámetros por Monitorear	
Duración: 1 hrs. 0 mins.		¿Guardar? <input checked="" type="radio"/> Sí	Parámetros
Intervalo de Muestreo: 10.0 segs.		<input checked="" type="radio"/> Sí	Valores Actuales
Tiempo de Espera: 1.50 segs.		<input checked="" type="radio"/> Sí	120.30 KWH
Actualizar Archivo cada: Intervalo de Muestreo		<input checked="" type="radio"/> Sí	36.80 KVARH
Indicar el Tiempo: 5.00 mins.		<input checked="" type="radio"/> Sí	125.98 KVAH
		<input checked="" type="radio"/> Sí	0.95
		<input checked="" type="radio"/> Sí	119.60 KW
		<input checked="" type="radio"/> Sí	119.60 KW
		<input checked="" type="radio"/> Sí	100.59 %
		<input checked="" type="radio"/> Sí	22:42:32 hh:mm:ss
		Constante de Energía: 0.100 KWh/Pulso	Iniciar Monitoreo <input checked="" type="radio"/>
			Tiempo Transcurrido 01:00:00.00 segs.

Figura 5.8. Datos del *monitoreo* y resultados finales de los parámetros de consumo eléctrico.

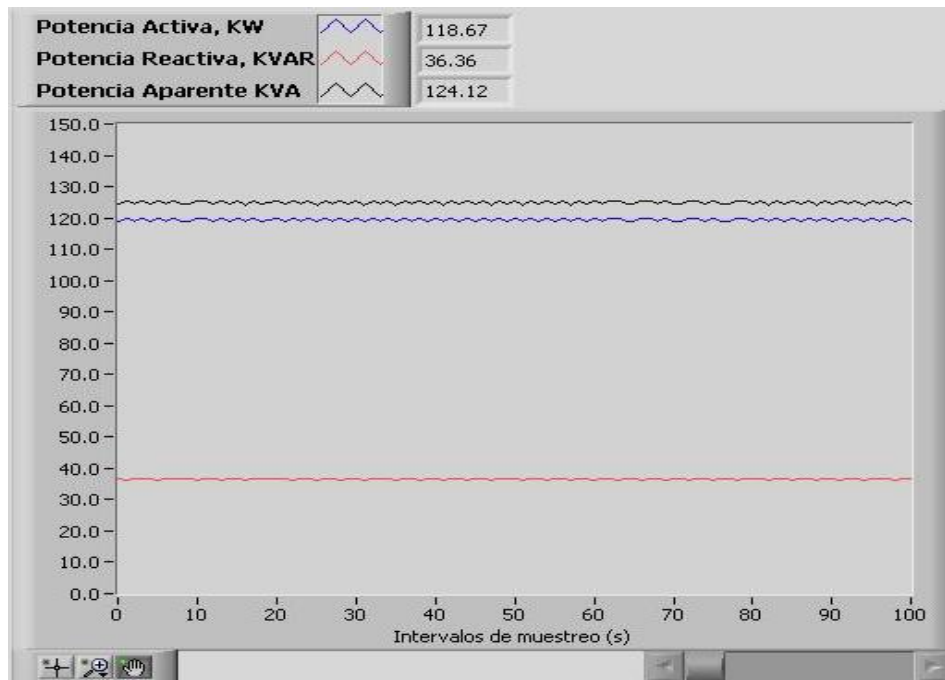


Figura 5.9. Curvas de potencia reportadas por el Soft-KYZ durante el *monitoreo*.

En la figura 5.9 puede observarse que las gráficas de potencias no son completamente líneas rectas, existen pequeños cambios de valor debido a que no siempre se obtuvieron exactamente los mismos valores de cuantificación de pulsos KYZ en todas las consultas. La razón es que las consultas no siempre se efectuaron exactamente en el tiempo marcado por el *intervalo de muestreo* y por una diferencia de un solo intervalo de medición (100 ms) entre una consulta y otra, se generaron estos cambios de valor constante (idealmente 120 KW).

Tabla 5.2. Cálculos y valores esperados del *monitoreo* hecho en las pruebas de laboratorio.

Parámetro	Valor ideal esperado	Valor reportado	% Error
Potencia Activa Instantánea	$P = \left(\frac{0.1KWh}{pulso}\right)\left(\frac{1pulso}{3s}\right)\left(\frac{3600s}{1h}\right) = 120KW$	118.67 KW	1.11
E. Activa	$Ea = (120KW)(1h) = 120KWh$	120.3 KWh	0.25
Potencia Reactiva Instantánea	$Q = \left(\frac{0.1KVARh}{pulso}\right)\left(\frac{1pulso}{9.8s}\right)\left(\frac{3600s}{1h}\right) = 36.7KVAR$	36.8 KVARh	0.93
E. Reactiva	$Er = (36.7KVAR)(1h) = 36.7KVARh$	36.8 KVARh	0.27
Potencia Aparente Instantánea	$S = \sqrt{(36.7KVAR)^2 + (120KW)^2} = 125.48KVA$	124.12 KVA	1.08
E. Aparente	$Eap = \sqrt{(36.7KVARh)^2 + (120KWh)^2} = 125.48KVAh$	125.98 KVAh	0.40

En la figura 5.10 se muestran las curvas completas de las potencias instantáneas hechas en Microsoft Excel con los datos finales del archivo de reporte. En estas gráficas también pueden observarse los pequeños cambios del valor constante esperado idealmente.

En la figura 5.11 se muestran las rectas que describen el consumo constante de las energías activa, reactiva y aparente que se obtienen por la conmutación constante de los relevadores del emulador.

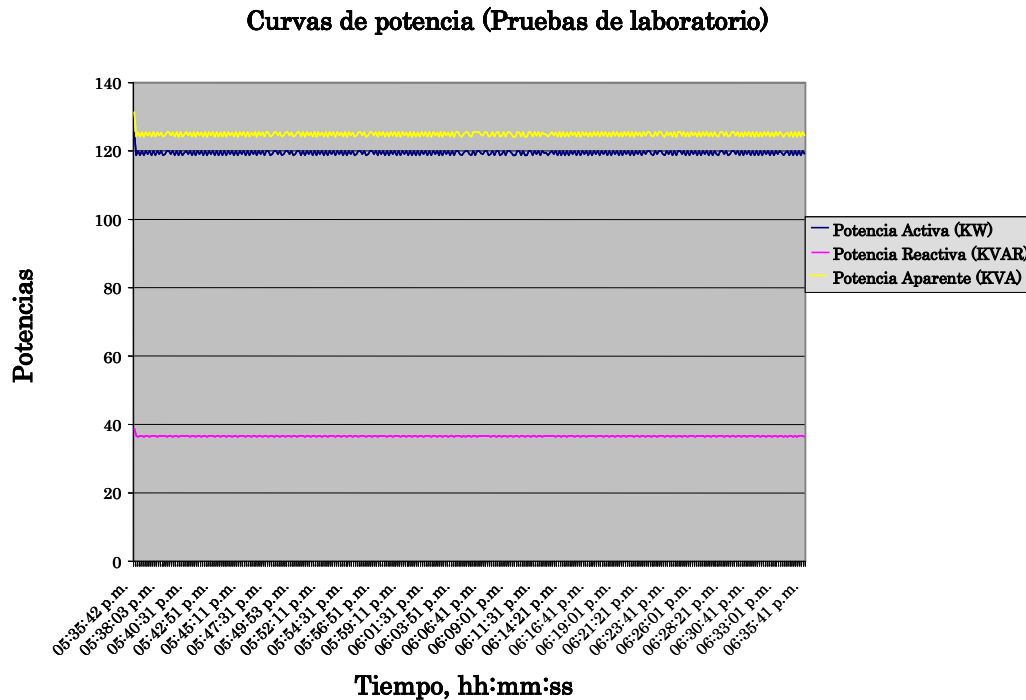


Figura 5.10. Curvas de potencia del *monitoreo* completo dibujadas en Microsoft Excel.

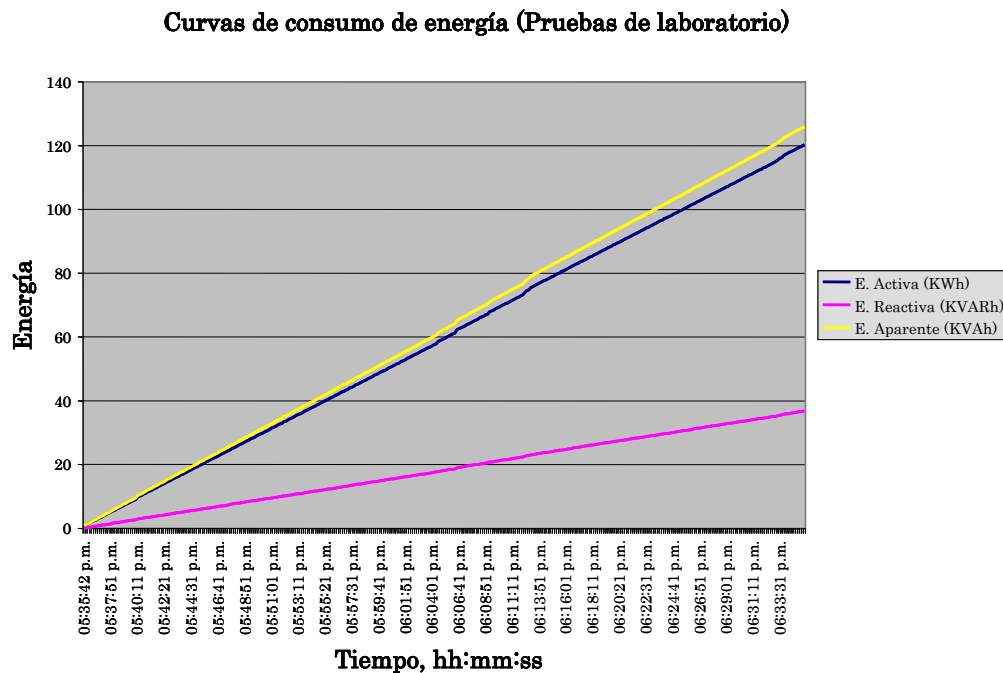


Figura 5.11. Curvas de consumo de energía *monitoreo* completo dibujadas en Microsoft Excel.

5.2.2.1.4. Datos de las transacciones MODBUS hechas con el puerto RS-232

En la figura 5.12 se presentan los valores finales de los *Datos de las transacciones* MODBUS efectuadas durante el *monitoreo*. El porcentaje de transacciones válidas de este *monitoreo*, que duró 1 hora y que realizó 363 consultas en intervalos de 10 segundos, es del 97%, el cual es un valor aceptable para estas condiciones de operación.

Software de Pruebas

Datos de transacciones MODBUS	Valor
Consultas Realizadas	363
Total de Respuestas Recibidas	357
Respuestas Correctas	351
Respuestas de Excepción	0
Respuestas con CRC erróneo	6
Consultas sin Respuesta	6
Transacciones Válidas (%)	97

Reinicializar Registros

Figura 5.12. Datos finales de las transacciones hechas con el puerto RS-232.

5.2.2.2. Resultados empleando el puerto serial RS-485

Usando el puerto RS-485, los resultados que se obtuvieron en los pasos del 1 al 3 del procedimiento de pruebas (*subcapítulo 5.1*) son muy similares a los que se obtuvieron usando el puerto RS-232. Por esta razón, ahora sólo se reportan los *Datos de las Transacciones MODBUS* como resultado de las pruebas hechas con el puerto RS-485 (paso 4 del procedimiento de pruebas).

5.2.2.2.1. Datos de las transacciones MODBUS hechas con el puerto RS-485

En la *figura 5.13* se ilustran los valores finales de los *Datos de las transacciones MODBUS* efectuadas durante el *monitoreo* empleando el puerto RS-485, con el cual se puede observar que el 98% de las transacciones son válidas, superando incluso el porcentaje reportado por las transacciones hechas con el puerto RS-232. Cabe recalcar que para realizar estas pruebas se empleó un carrete de cable par trenzado de 250 m de longitud (100 veces mayor al cable empleado en RS-232).

Adicionalmente a esta figura, en la *tabla 5.3* se hace una comparación de los resultados obtenidos con el estándar RS-485 de 4 y 2 hilos a las 5 diferentes velocidades de operación soportadas por el CODKYZ (1200, 2400, 4800, 9600 y 19200 baudios). La tabla reporta el porcentaje de transacciones válidas de un total de 350 (100%) hechas con el Soft-KYZ para cada velocidad.

En esta tabla puede observarse que tanto la cantidad de errores de CRC como las consultas sin respuesta son menores en RS-485 de 2 hilos y que la velocidad de operación que presentó menos errores fue 9600 baudios.

Es importante mencionar que para estas pruebas se ocupó un cable de par trenzado para conexiones de redes Ethernet con las siguientes características: cable UTP de 4 pares, calibre 24 AWG, categoría 5e. Las características de este cable ayudaron a mejorar la calidad de transmisión de los datos [19].

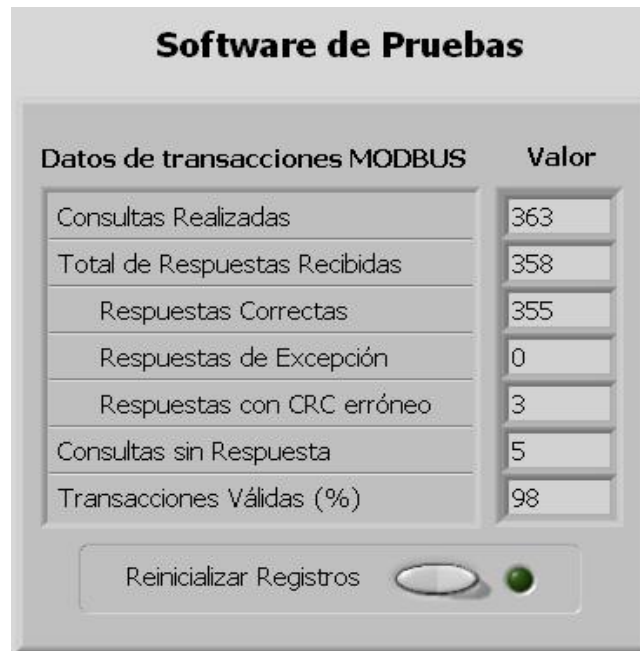


Figura 5.13. Datos finales de las transacciones hechas con el puerto RS-485.

Tabla 5.3. Comparación de la eficiencia del estándar RS-485 a 4 y 2 hilos.

350 cons.	Resultados con 4 hilos				Resultados con 2 hilos			
Velocidad	Válidas	CRC	Sin resp.	% Válidas	Válidas	CRC	Sin resp.	% Válidas
1200	332	4	14	95	338	6	6	97
2400	334	9	7	95	341	5	4	97
4800	341	7	2	97	344	1	5	98
9600	343	1	6	98	345	3	2	98
19200	339	7	4	97	342	6	2	98

5.2.3. Problemas presentados en las pruebas de laboratorio

En un inicio, el microcontrolador del CODKYZ se había programado para que detectara los pulsos KYZ usando tres de sus interrupciones que son activadas por eventos en un *pin* (flancos de subida y caída en este caso).

Sin embargo, con estas pruebas se comprobó que hacer esto provoca conflictos entre las interrupciones del AT90S2313, principalmente por los rebotes generados cuando los relevadores KYZ conmutan, ya que si se usan retardos para evadir los rebotes en este modo de operación, muchas veces estos retardos provocan que no se generen las interrupciones que deben ser activadas por las otras entradas de pulsos.

Lo anterior se debe a que sólo se genera la interrupción de más alta prioridad en el microcontrolador y, mientras se usa un retardo para validar el primer pulso, cuando ocurre un flanco en otra entrada la interrupción no se activa y el pulso no se detecta. El problema se resolvió usando la técnica de *monitoreo* de *pin*s o “*polling*”, descrita en el apartado 3.5.1.

Otro problema que se presentó en estas pruebas de laboratorio es que, después de un tiempo de operación (aproximadamente 30 minutos) los relevadores de CD empleados para generar los PKWh y PKVARh se quedaban “pegados”, es

decir, ya no conmutaban a pesar de que las señales de activación/desactivación del microcontrolador se estaban generando correctamente.

La causa de este problema era que, después de este tiempo de operación, las bobinas (inductores) de los relevadores almacenaban energía e impedían la variación de corriente cada 3 segundos en el caso de los PKWh, o cada 9 segundos en el caso de los PKVARh. La solución fue conectar un diodo de propósito general (1N4001) en polarización inversa al voltaje de alimentación para descargar al inductor cuando no era energizado por la fuente (*figura 5.5a*).

5.3. Pruebas de campo

Las pruebas de campo consistieron en ejemplificar el modo real de operación para el cual fue diseñado el sistema que consiste en lo siguiente: el CODKYZ se debe conectar al medidor de energía en la subestación eléctrica de la planta que se va a analizar y, desde una sala de control, una Estación Maestra se encarga de configurarlo, darle órdenes y de solicitarle sus datos de información, al igual que lo hace con los otros dispositivos Esclavos pertenecientes a la red MODBUS establecida.

Las pruebas de campo que se realizaron en esta ocasión, sólo incluyen al CODKYZ como único Esclavo de la red MODBUS, instalado en campo, y al Soft-KYZ como Maestro, instalado en una PC de laboratorio.

5.3.1. Instalación del sistema para pruebas de campo

El diagrama de la *figura 5.14* ilustra la instalación total del sistema para efectuar las pruebas de campo. Dicha instalación se divide en dos: *instalación de campo* e *instalación en sala de control*, ambas tienen en común el uso de los radio-módems de 900 MHz, para establecer la conexión inalámbrica entre los dos nodos de la red MODBUS: el CODKYZ y el Soft-KYZ.

5.3.1.1. Elementos de las pruebas de campo

Los elementos que integran el sistema de pruebas de campo son:

1. La acometida trifásica principal de la UTM.
2. El medidor de energía PM650, el cual sólo ofrece un relevador KYZ de salida que se programó para generar pulsos KYZ reales de energía activa.
3. Interruptor trifásico a través del cual se conectan las tres fases de voltaje (V_a , V_b y V_c) al PM650.
4. Tres transformadores de corriente de relación 600:5 A, a través de los cuales se conectan las tres fases de corriente (I_a , I_b e I_c) al PM650.
5. Emulador de relevadores KYZ para los PKVARh y PSINC.
6. El CODKYZ como único Esclavo de la red.
7. Un par de radio-módems modelo 9XStream de 900MHz para la *conexión inalámbrica*.
8. El Soft-KYZ como Maestro de la red.

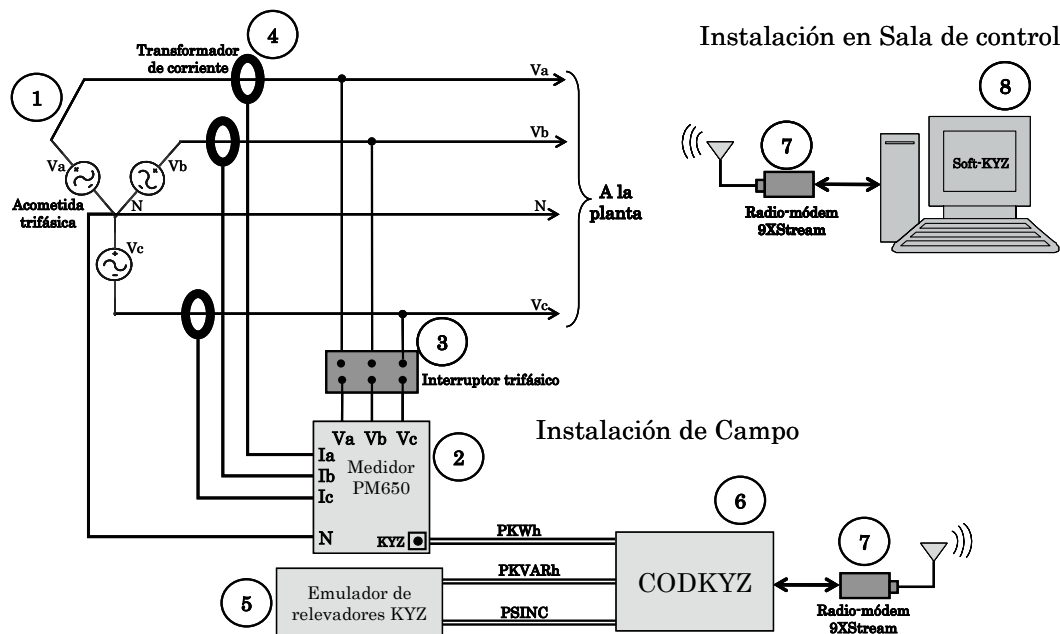


Figura 5.14. Diagrama de instalación del sistema para las pruebas de campo.

5.3.1.2. Instalación de campo

En la *instalación de campo*, el medidor PM650 se conectó a la acometida trifásica principal de la subestación eléctrica de la UTM (*figura 1.1*) para que registrara su consumo eléctrico [35, 36]. El PM650 sólo tiene un relevador KYZ, el cual se programó para generar los PKWh reales que son las entradas principales del CODKYZ en estas pruebas. La constante de energía de este medidor se configuró con un valor de 0.1 KWh/pulso, es decir, su relevador KYZ generaba un pulso cada vez que la planta analizada (UTM) completaba un consumo de 0.1 KWh de energía activa. En la *figura 5.15* se muestra el aspecto de la instalación de campo y en la *figura 5.16*, la ubicación de la antena empleada para establecer el enlace inalámbrico.



Figura 5.15. Aspecto de la instalación del sistema de pruebas de campo.



Figura 5.16. Ubicación de la antena en la instalación de campo.

En esta instalación también se empleó el emulador de relevadores KYZ descrito en las pruebas de laboratorio, sólo que aquí se ocupan únicamente dos relevadores de CD, uno para emular los PKVARh y el otro para emular los PSINC.

La *figura 5.17* muestra la conexión de los componentes centrales de la instalación de campo: el relevador KYZ del PM650 y los dos relevadores del emulador se conectan a las entradas de pulsos del CODKYZ; y el radio-módem 9Xstream, al puerto RS-232.

La *instalación en sala de control* se hizo en el laboratorio del Instituto de Electrónica y Computación de la UTM (IEC-UTM) con el Soft-KYZ funcionando como Maestro de la red MODBUS. En la *figura 5.18* puede observarse la configuración real del sistema: el radio-módem se conectó a la PC a través del puerto RS-232 y su antena se colocó en la parte exterior del edificio.

Es importante mencionar que después de haber realizado las instalaciones explicadas anteriormente de acuerdo al diagrama de la *figura 5.14*, no se presentó ningún problema a la hora de efectuar las pruebas de campo del sistema completo. He aquí la importancia de haber realizado previamente las pruebas de laboratorio.

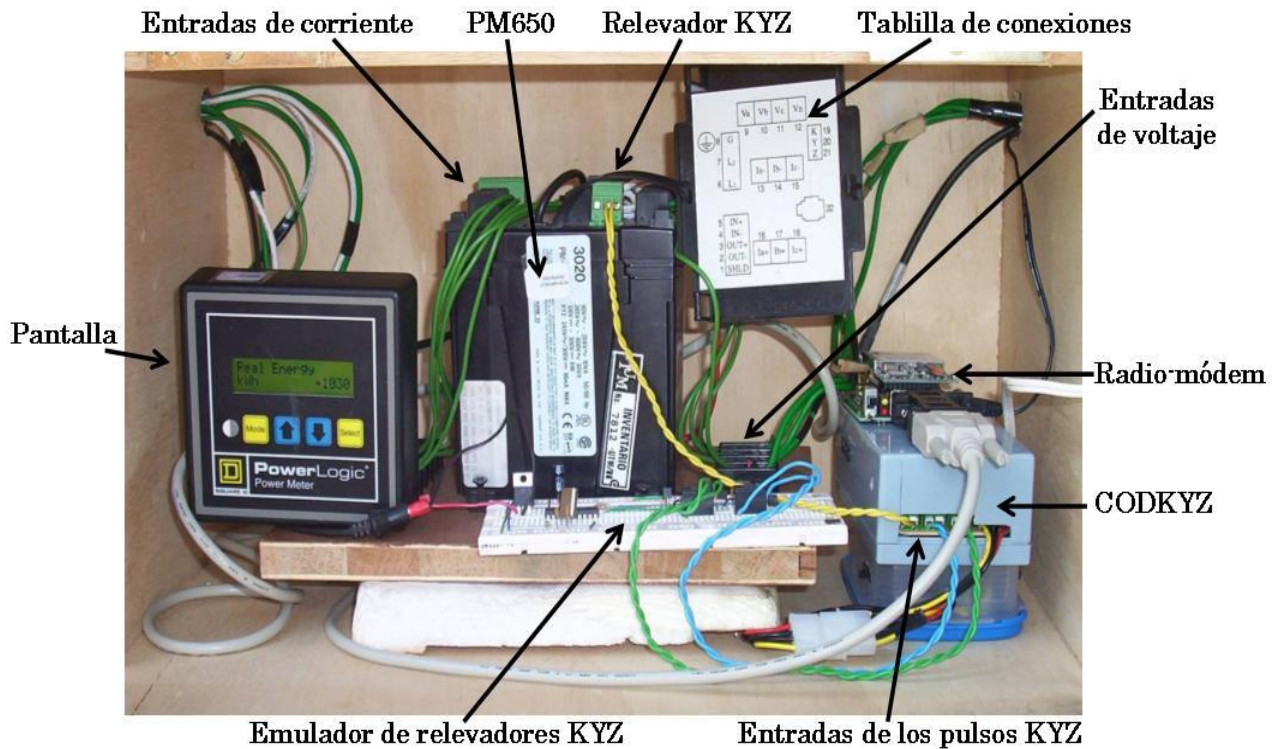


Figura 5.17. Conexión de los componentes centrales: PM650, CODKYZ y radio-módem.

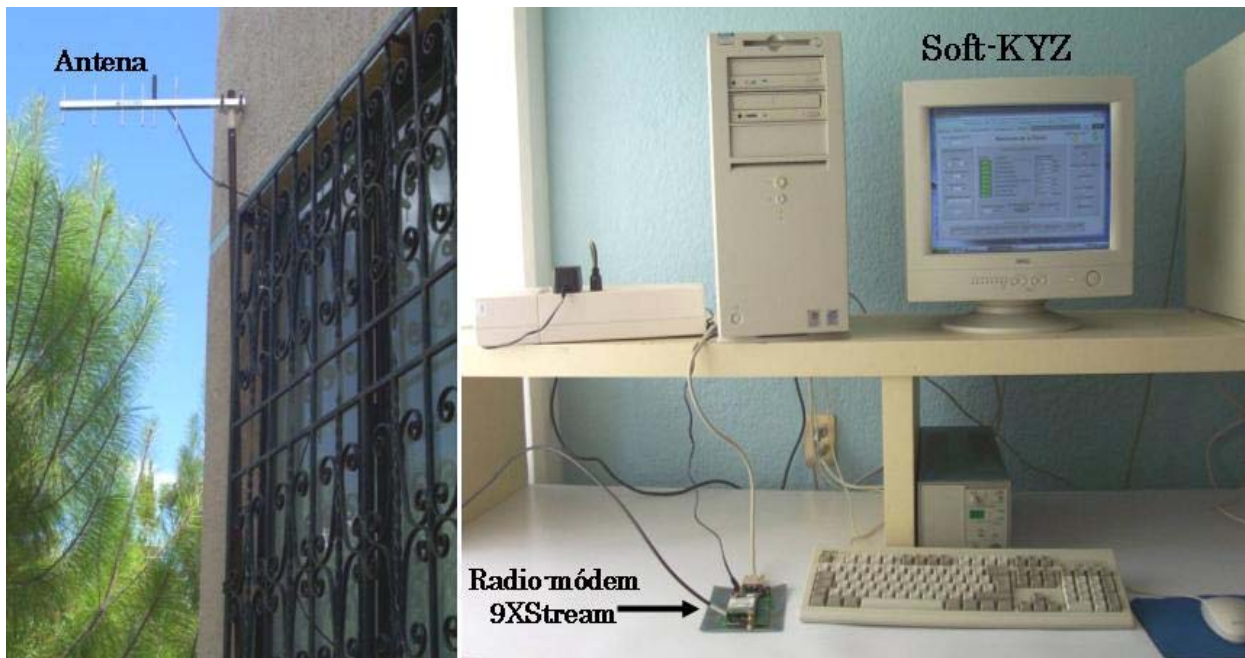


Figura 5.18. Aspecto real de la instalación hecha en sala de control.

5.3.2. Resultados de las pruebas de campo

En los resultados de las pruebas de campo que se presentan a continuación, se omiten los de los pasos 1 y 2 del procedimiento presentado en el *subcapítulo 5.1*, la razón es que ahora se da por hecho de que esos pasos ya han sido comprobados previamente antes de efectuar los pasos 3 y 4.

5.3.2.1. *Monitoreo* de pulsos KYZ

En la *figura 5.19* se muestran los datos del *monitoreo* realizado para la prueba de campo. Los valores de los datos son: *Duración*, 120 horas y 0 minutos (5 días); *Intervalo de muestreo*, 150 segundos (cada 2.5 minutos); y *Tiempo de espera*, 1.5 segundos.

Las gráficas de potencia que reporta el Soft-KYZ en su página *Gráficas* (*figura 5.20*), corresponden sólo a una parte del *monitoreo* de las variables del consumo eléctrico registrado por la UTM entre el 5 y 9 de agosto de 2004. Las gráficas completas de potencia activa, energía activa, demanda medida y demanda máxima de este consumo se presentan en las *figuras 5.21, 5.22 y 5.23*, respectivamente. Estas gráficas se hicieron en Microsoft Excel a partir del archivo de reporte, el cual se muestra editado en la *figura 5.24* presentando sólo el encabezado, la parte donde se registra la demanda máxima y el final del archivo.

Más adelante se presenta una tabla de comparación entre los valores de estos parámetros obtenidos con el Soft-KYZ y los valores que registró el medidor PM650 al final del *monitoreo*.

Es importante mencionar que las demás variables, como el FP y las potencias reactiva y aparente no se presentan en estas últimas gráficas porque serían erróneas. La razón es que, en las pruebas de campo, el PM650 sólo generó pulsos KYZ reales de energía activa y los de energía reactiva se emularon.

Para ocupar los PKWh, PKVARh Y PSINC del medidor de energía de CFE instalado para facturación de la UTM, se requiere de una tarjeta adicional con salidas a relevadores y el permiso correspondiente de CFE para hacer uso de su equipo de medición. Desafortunadamente en estos momentos no se cuenta con tales condiciones de trabajo.



Datos del Monitoreo

Duración:
120 hrs. 0 mins.

Intervalo de Muestreo:
150.0 segs.

Tiempo de Espera:
1.50 segs.

Actualizar Archivo cada:
Intervalo de Muestreo

Indicar el Tiempo:
5.00 mins.

Figura 5.19. Datos del *monitoreo* hecho para las pruebas de campo.

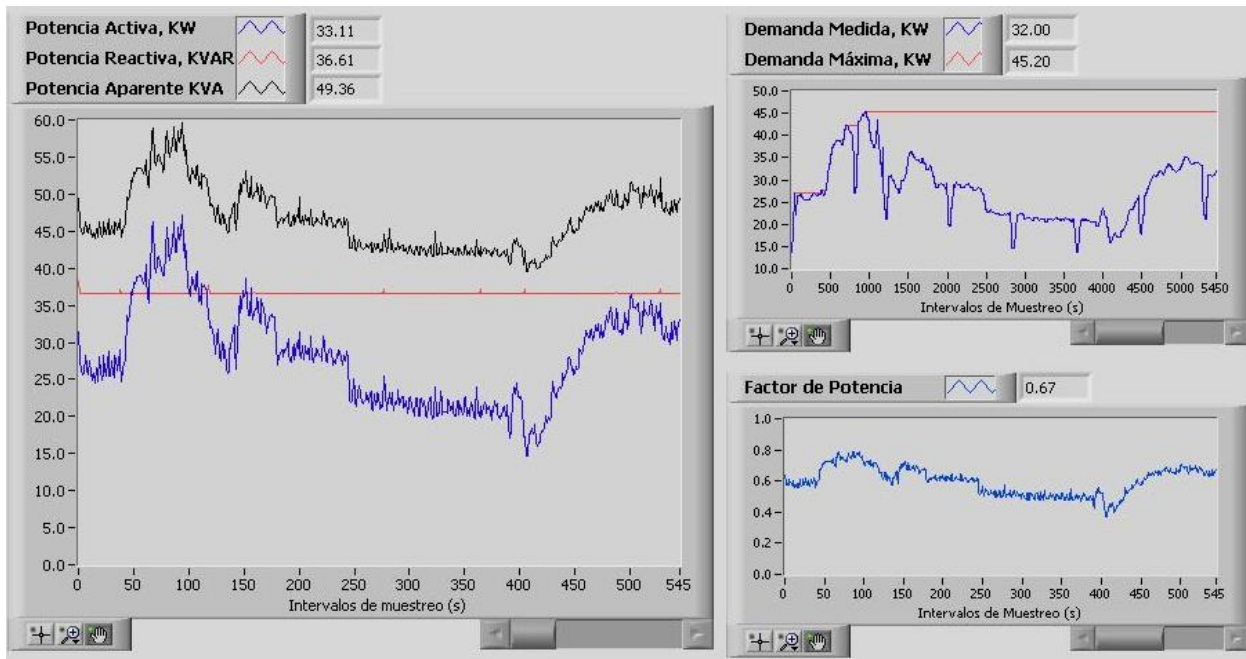


Figura 5.20. Gráficas reportadas por el Soft-KYZ durante el monitoreo.

Curva de potencia activa registrada por la UTM del 05 al 09 de agosto de 2004

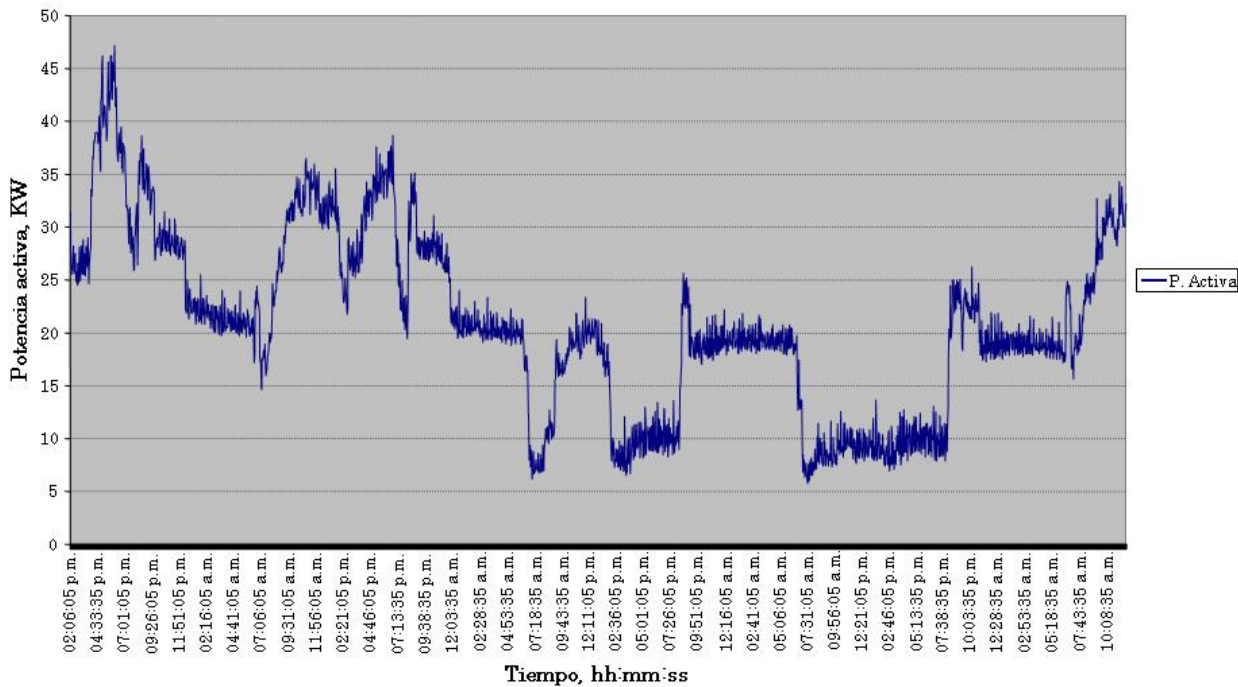


Figura 5.21. Curva de potencia activa hecha en Microsoft Excel a partir del archivo de reporte.

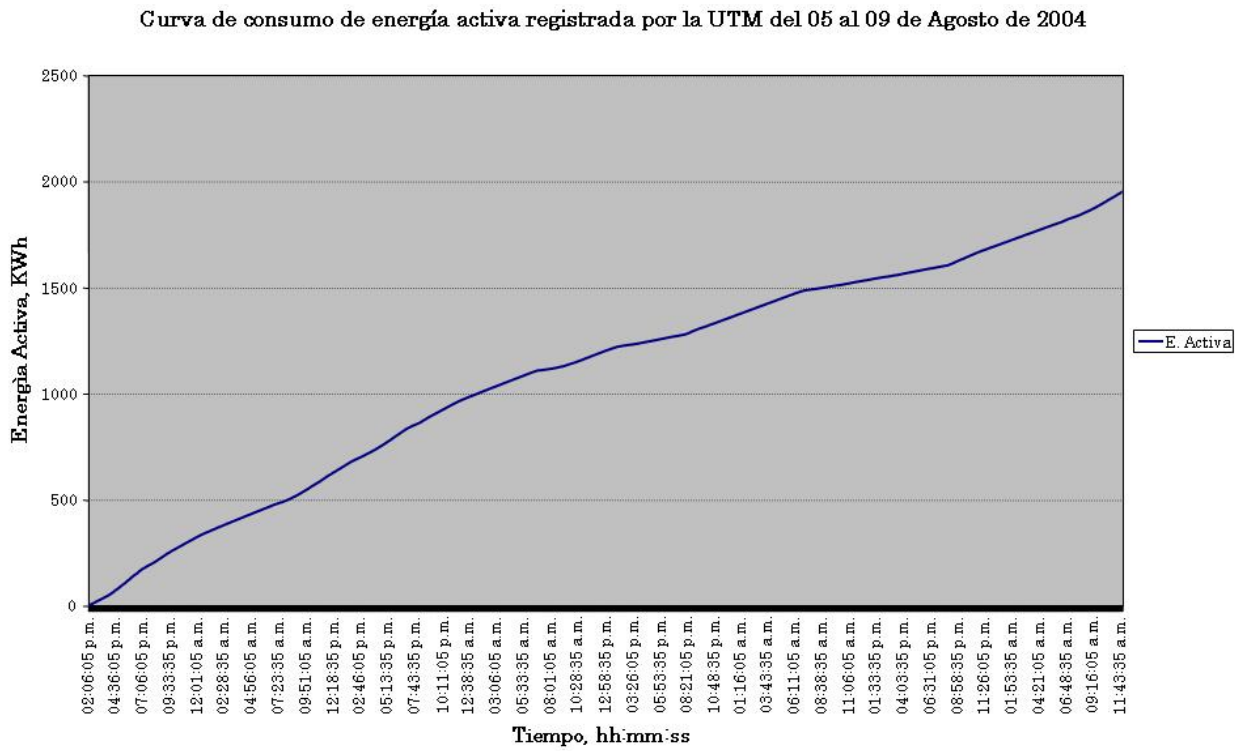


Figura 5.22. Curva de consumo de energía activa realizada en Microsoft Excel.

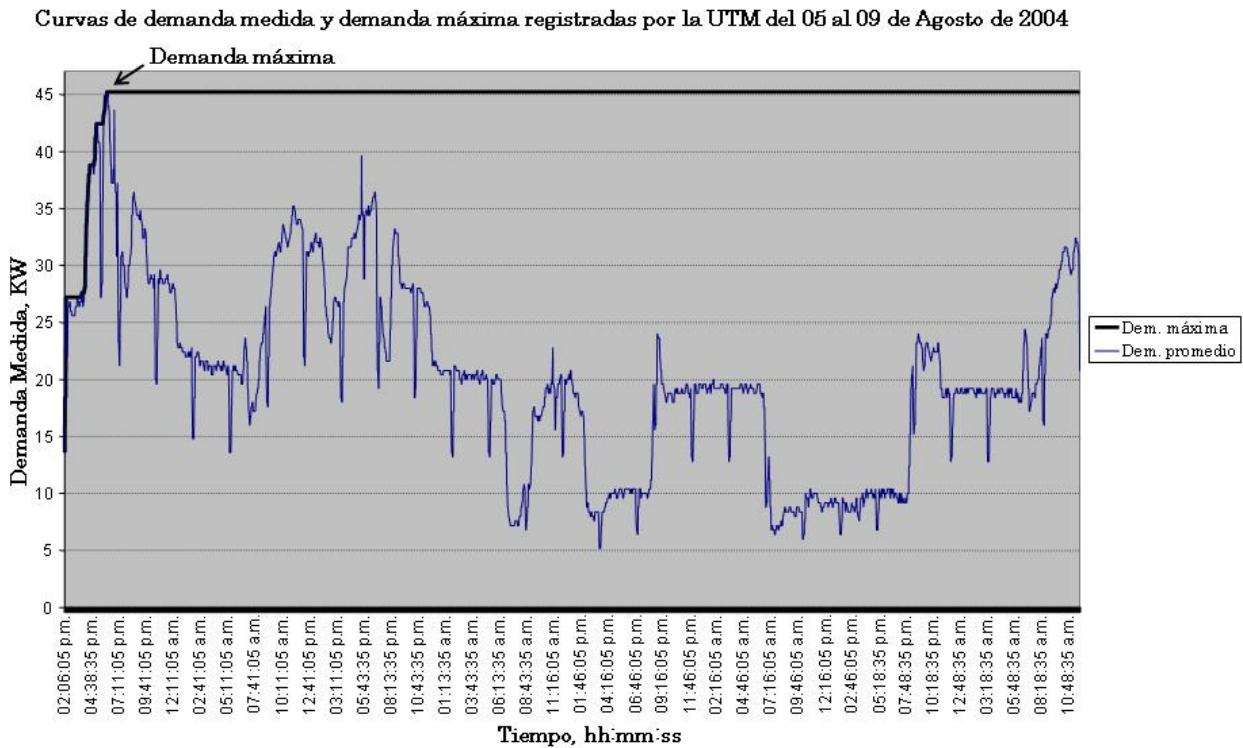


Figura 5.23. Curvas de demanda medida y demanda máxima realizada en Microsoft Excel.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Sistema de Monitoreo de Pulsos KYZ para el Cálculo de Variables de Consumo Eléctrico											
2	Fecha de Inicio de Monitoreo: 05/08/2004											
3	Hora (H:M:S)	P. Activa	P. Reactiva	P. Aparente	E. Activa	E. Reactiva	E. Aparente	FP	Dem. Máxima	Dem. Promedio	F. Carga (%)	SINC
4	02:06:05 p.m.	31.39	38.21	49.45	3.4	4.2	5.4	0.63	13.6	13.6	Inf	548
5	02:08:35 p.m.	27.89	36.63	46.04	4.6	5.7	7.32	0.63	18.4	18.4	599.98	1802
6	02:11:05 p.m.	26.03	36.61	44.92	5.7	7.2	9.18	0.62	18.4	18.4	599.98	1802
7	02:13:35 p.m.	26.4	36.61	45.14	6.8	8.8	11.13	0.61	27.2	27.2	200	2859
8	02:16:05 p.m.	25.55	36.61	44.64	7.8	10.3	12.93	0.6	27.2	27.2	200	2859
9	02:18:35 p.m.	25.88	36.63	44.85	8.9	11.8	14.79	0.6	27.2	22	157.06	4110
10	02:21:05 p.m.	28.23	36.61	46.23	10.1	13.3	16.71	0.6	27.2	22	157.06	4110
11	02:23:35 p.m.	27.12	36.61	45.57	11.2	14.9	18.65	0.6	27.2	26.4	141.18	5167
12	02:26:05 p.m.	26.38	36.61	45.12	12.3	16.4	20.51	0.6	27.2	26.4	141.18	5167
13	02:28:35 p.m.	25.7	36.61	44.73	13.4	17.9	22.37	0.6	27.2	26.4	131.37	6420
14	02:31:05 p.m.	27.52	36.63	45.82	14.5	19.4	24.23	0.6	27.2	26.4	131.37	6420
15	02:33:35 p.m.	25.86	36.61	44.83	15.6	21	26.17	0.6	27.2	26.8	125.13	7475
Momento en que se registró la demanda máxima												
95	05:56:05 p.m.	43.84	36.61	57.12	132.4	144.7	197.27	0.67	44.8	44.8	76.82	7719
96	05:58:35 p.m.	43.78	36.61	57.07	134.2	146.2	199.61	0.67	44.8	44.4	77.3	8968
97	06:01:05 p.m.	44.73	36.63	57.82	136.1	147.7	202.03	0.67	44.8	44.4	77.3	8968
98	06:03:35 p.m.	47.17	36.61	59.71	138	149.3	204.52	0.67	45.2	45.2	77.13	10023
99	06:06:05 p.m.	43.2	36.61	56.62	139.8	150.8	206.86	0.68	45.2	45.2	77.13	10023
100	06:08:35 p.m.	41.38	36.61	55.25	141.5	152.3	209.13	0.68	45.2	44	77.46	11274
2240	11:28:35 a.m.	31.92	36.61	48.57	1940.9	3423.8	3995.18	0.49	45.2	32.4	45.99	15108
2241	11:31:05 a.m.	33.82	36.63	49.86	1942.3	3425.3	3997.23	0.49	45.2	32.4	45.99	15108
2242	11:33:35 a.m.	32.45	36.61	48.92	1943.6	3426.8	3999.21	0.49	45.2	32	46.01	803
2243	11:36:05 a.m.	31.68	36.61	48.41	1945	3428.4	4001.34	0.49	45.2	32	46.01	803
2244	11:38:35 a.m.	30.21	36.61	47.46	1946.2	3429.9	4003.26	0.49	45.2	32	46.03	2056
2245	11:41:05 a.m.	30.02	36.63	47.36	1947.5	3431.4	4005.24	0.49	45.2	32	46.03	2056
2246	11:43:35 a.m.	30.21	36.61	47.46	1948.7	3432.9	4007.17	0.49	45.2	31.2	46.05	3112
2247	11:46:05 a.m.	30	36.61	47.33	1950	3434.5	4009.23	0.49	45.2	31.2	46.05	3112
2248	11:48:35 a.m.	31.2	36.61	48.1	1951.3	3436	4011.21	0.49	45.2	30.8	46.07	4364
2249	11:51:05 a.m.	31.45	37.76	49.14	1952.6	3437.5	4013.2	0.49	45.2	25.6	46.08	0
2250	11:53:35 a.m.	32.23	36.63	48.79	1953.9	3439	4015.18	0.49	45.2	20.8	46.09	5422
2251	Fecha de Finalización: 09/08/2004											
2252												
2253	Datos de las Transacciones:											
2254	Total de consultas realizadas: 2375											
2255	Total de respuestas recibidas: 2344											
2256	Respuestas correctas: 2312											
2257	Respuestas de excepción: 0											
2258	Respuestas con CRC erróneo: 32											
2259	Consultas sin respuesta: 31											
2260	Transacciones válidas (%): 97											

Figura 5.24. Encabezado, registro de demanda máxima y parte final del archivo de reporte.

En la figura 5.25 se presenta la página *Monitoreo* con los valores finales de las variables de consumo obtenidos al terminar el *monitoreo*. Estos resultados se comprobaron con los valores registrados por el medidor PM650 (figura 5.26) en la tabla 5.4. Cabe aclarar que el PM650 sólo reporta valores enteros.



Figura 5.25. Resultados finales del *monitoreo* reportados por el Soft-KYZ.



Figura 5.26. Resultados finales del monitoreo reportados por el medidor PM650.

Tabla 5.4. Comparación de los resultados del monitoreo reportados por el Soft-KYZ y el PM650.

Variable	PM650	Soft-KYZ	Diferencia	% Error
Energía activa total (KWh)	1951	1951.3	0.3	0.01
Demanda máxima (KW)	45	45.2	0.2	0.44
Demanda rolada actual (KW)	31	30.8	0.2	0.65

5.3.2.2. Datos de las transacciones MODBUS hechas con los radio-módems

En la figura 5.27 se presentan los valores finales de los Datos de las transacciones MODBUS efectuadas durante el monitoreo. El porcentaje de transacciones válidas de este monitoreo, que duró exactamente 4 días, 2 horas, 57 minutos y 30 segundos y que realizó en total 2375 consultas en intervalos de 150 segundos, es del 97%, el cual es un valor bastante aceptable para estas condiciones de operación, considerando la época lluviosa en la que se realizaron estas pruebas de enlace inalámbrico.

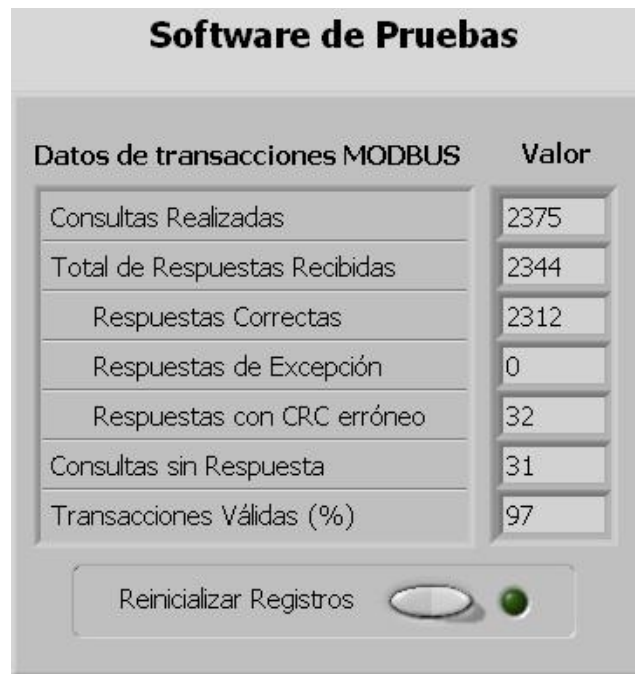


Figura 5.27. Datos finales de las transacciones hechas con el enlace inalámbrico.

Conclusiones

En el presente trabajo de tesis se construyó un dispositivo y un software con los que se comprueba que usando las salidas pulsadas de un medidor electrónico de energía, se pueden obtener todas las variables globales del consumo eléctrico que registra una planta. Esto con la finalidad de estudiar su comportamiento e implementar medidas de prevención, control y corrección que ayuden a optimizar el uso de la electricidad. Tras la finalización de este sistema y considerando las diferentes fases de su desarrollo, a continuación se presentan las conclusiones del trabajo:

- El CODKEYZ es un producto terminal a nivel de circuito impreso con chasis listo para instalarse en campo y ser incluido dentro de una red que emplee el protocolo MODBUS para la comunicación entre sus elementos.
- Considerando el total de funciones y especificaciones MODBUS que se implementaron en el CODKEYZ, se hace una estimación de que se cumplió en un 80% con los requisitos establecidos por la norma oficial de este protocolo para un dispositivo Esclavo sobre enlace serial.
- Respecto a la codificación del firmware del CODKEYZ, el lenguaje ensamblador ofreció la ventaja de optimizar la memoria de código y tener un mejor control de la RAM del microcontrolador AT90S2313, ya que para programar esta misma aplicación en lenguaje C se necesita un microcontrolador con más memoria de código y más espacio en RAM, pero a la vez, más grande, más costoso y con recursos que serían desperdiciados porque no son necesarios en esta aplicación.
- El porcentaje de uso que se le dio a los recursos empleados del microcontrolador AT90S2313 son: FLASH: 1014 de 1024 palabras de 16 bits (99%), RAM: 120 de 128 bytes (90%), EEPROM: 16 de 128 bytes 9%, Timers: 2 de 2 (100%), Interrupciones: 7 de 11 (70%), UART 1 de 1, Puertos: 7 de 15 líneas de entrada/salida (*pines*) 50%.
- Las primeras pruebas que se realizaron al CODKEYZ bajo el protocolo MODBUS, consistieron en comprobar que respondiera correctamente a un conjunto limitado de mensajes generados con la versión de prueba del

software ModScan32, disponible en la dirección WEB <http://www.wintech.com/html/demos.htm>. El desarrollo del Soft-KYZ como herramienta de prueba, trajo la ventaja significativa de comprobar, en forma práctica, que el CODKYZ cumple satisfactoriamente con todas las tareas para las que fue implementado y que opera de manera correcta bajo el protocolo de comunicaciones MODBUS. El Soft-KYZ ofrece además, una herramienta didáctica para probar cualquier dispositivo que soporte este protocolo.

- Los radio-módems de la firma MaxStream, ofrecieron la ventaja de contar con un sistema de comunicación inalámbrica con sólo configurar su velocidad de operación, instalar su antena y conectarlo al puerto RS-232 del CODKYZ y de la PC. Estos radio-módems son parte de la infraestructura con la que se cuenta en la UTM; sin embargo, el CODKYZ puede emplear cualquier otro que posea una interfaz de comunicación RS-232 o RS-485.
- El CODKYZ cuenta con un puerto RS-485 para conectarse con varios dispositivos Esclavos que trabajen en campo y puedan ser consultados remotamente por el Maestro de la red [3]. Además, en la práctica se obtuvieron mejores resultados con este estándar que con el RS-232, a pesar de que el cable de éste último era de 2 m y el de RS-485 era de 250 m.
- Las ventajas de haber usado LabVIEW como lenguaje de programación para implementar el Soft-KYZ son: se pudieron hacer pruebas de ejecución a sus módulos independientemente del programa principal; se pudo construir una GUI con enfoque industrial entendible, amigable y fácil de usar; y se pudo construir el archivo ejecutable del Soft-KYZ.

A continuación se listan los principales problemas que se presentaron durante el desarrollo de este trabajo de tesis:

- Los rebotes que los relevadores de CD generaban en las pruebas hechas en laboratorio provocaron continuos cambios en el diseño planteado inicialmente para detectar los pulsos KYZ. Finalmente, se cambió la técnica de utilizar las interrupciones activadas por eventos en *pin*s, por la técnica de *monitoreo* continuo de *pin*s “polling” (consulte los *capítulos 3 y 5*).
- Inicialmente no se contaba con la infraestructura para realizar pruebas experimentales del estándar RS-485, esto retrasó la culminación de esta parte del hardware. Fue necesario construir un convertidor RS-232 a RS-485 para pruebas iniciales y posteriormente se adquirió uno comercial. Con ambos se obtuvieron resultados favorables (*tabla 5.3*).
- Respecto al software, el principal problema fue la nula experiencia inicial en el paradigma de programación gráfica y del funcionamiento de LabVIEW. Afortunadamente este problema disminuyó al avanzar el proyecto tras haber estudiado la bibliografía del tema [7, 29, 30, 33] y después de haber seguido un modelo de desarrollo de software y aplicar las técnicas que la literatura recomienda [31, 32].

Trabajos futuros

Actualmente se tiene contemplado tomar este trabajo de tesis como base para iniciar la construcción de una red inalámbrica que opere bajo MODBUS y que se dedique a controlar la demanda máxima y a corregir automáticamente el FP de una planta. Para ello es necesario realizar los siguientes proyectos:

- Diseñar y construir un dispositivo Esclavo MODBUS que opere como controlador de cargas eléctricas, es decir, que se encargue de conectar y desconectar cargas eléctricas cuyo funcionamiento pueda ser detenido durante ciertos periodos de tiempo, cuando el Maestro lo indique.
- Diseñar y construir un dispositivo Esclavo MODBUS que opere como controlador de los bancos de capacitores usados para corregir automáticamente el FP de una planta.
- Desarrollar el software de la Estación Maestra MODBUS que se encargará de configurar y dirigir las tareas de los Esclavos pertenecientes a la red: el CODKYZ, el controlador de cargas y el controlador de bancos de capacitores. En este caso se puede retomar el código del Soft-KYZ para agregarle las características adicionales u ocuparlo como herramienta de apoyo.
- Adicionalmente, se puede hacer un estudio de las tarifas eléctricas que establece la CFE para consumidores que ocupan medidores electrónicos de energía y agregarle al Soft-KYZ una sección que, de acuerdo al tipo de tarifa que posea la planta analizada, devuelva el costo que representa el consumo eléctrico *monitoreado*.
- Una mejora al hardware es implementar la capacidad de reportar a la Estación Maestra si la fuente de alimentación de los pulsos KYZ ha dejado de operar o que no ha recibido pulsos KYZ después de mucho tiempo, esto implicaría aumento de código de programa en el firmware y, por lo tanto, es necesario emplear un microcontrolador con mayor memoria de código.

Referencias

- [1] Torres, S.: “Automatización: Buses de Campo”. *Electroindustria*, Santiago de Chile, año 4, n° 38, Octubre de 2003.
- [2] Kaschel, H. y Pintol, E.: “Análisis del estado del arte de los buses de campo aplicados al control de procesos industriales”. Facultad de Ingeniería, Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile, Santiago de Chile, 2001.
- [3] Poza, F., Domínguez M., Mariño, P., Machado F. y Vázquez F.: “Sistema de Iniciación y Aprendizaje de Redes de Comunicaciones Industriales”. Departamento de Tecnología Electrónica, E.T.S.I. Industriales, Universidad de Vigo, España.
- [4] Mariño, P., Domínguez, M., Poza, F. y Benigno, J.: “Buses de Campo (I): Características y Aplicaciones”. *Automática e Instrumentación*, España, n° 314, Enero de 2001, pp.69-79.
- [5] Cobo, R.: “Buses de Campo”. *Electroindustria*, México, año 4, n° 31, Marzo de 2003.
- [6] Wright, R.: “La Inalámbrica Acelera el Objetivo Justo a Tiempo”. *InTech México Automatización*, México, año 2, n° 1, Enero-Marzo de 2003, pp.34-36.
- [7] Lázaro, A.: *LabVIEW 6i: Programación Gráfica para el Control de Instrumentación*. Paraninfo, España, 2001.
- [8] Ramírez, H., Acevedo, J., Maldonado, R. y Espinosa, E.: “Wireless System for Electrical Networks Testing Based on MODBUS Protocol”. CONIELECOMP 2004, XIV International Conference on Electronics, Communications and Computers, Veracruz, México, Febrero de 2004.
- [9] Santiago, F. y Ramírez, F.: “Diseño y Construcción de un Simulador de Medidor de Energía Eléctrica para Controladores de Demanda Máxima”. Memorias SOMI XIV: Congreso de Instrumentación, Puebla, México, Octubre de 1999.
- [10] Kissell, T.: *Industrial Electronics: Applications for Programmable Controllers, Instrumentation and Process Control, and Electrical Machines and Motor Controls*. Prentice Hall, USA, 2003.

- [11] Wildi, T.: *Electrical Machines, Drives and Power Systems*. Prentice Hall, USA, 2002.
- [12] GPI Mexicana de Alta Tecnología, S.A. de C.V.: Omnipotencihorímetro OPH-03/C Instructivo de Operación. Kitron, México, Octubre de 1994.
- [13] Esparza, M. y Altamira, J.: “Controlador de Demanda Máxima”. Departamento de Ingeniería Eléctrica y Electrónica, Instituto Tecnológico de Aguascalientes México, 1999.
- [14] Molina, R.: “Definitions for KYZ Output and Meter Pulse Constant Ke”. Tucson Electric Power Company, USA, 2000.
- [15] SquareD: Power-Logic Circuit Monitor Series 2000: Manual de Referencia, Boletín de Instrucciones n° 3020IB9820. SquareD Company, USA, Agosto de 1999.
- [16] Anybus: Fielbus Appendix ANYBUS®-S MODBUS RTU, Doc. Abs-Apendix-RTU Rev-1.0. HMS Industrial Networks, USA, Marzo de 2003.
- [17] MODBUS-IDA.ORG: MODBUS Application Protocol Specification V1.1, USA, 2002.
- [18] MODBUS-IDA.ORG: MODBUS over Serial Line (V1) Specification and Implementation Guide V1.0, USA, 2002.
- [19] B&B Electronics: RS-422 and RS-485 Application Note. B&B Electronics Mfg. Co. Inc., USA, 1997.
- [20] Atmel: 8-bit AVR® Microcontroller with 2K Bytes of In-System Programmable Flash AT90S2313, Rev. 08391-AVR-06/02. Atmel Corporation, USA, 2002.
- [21] Fairchild Semiconductor: LM7805: 3-Terminal IA Positive Voltage Regulator. Fairchild Semiconductor International Inc., USA, 2001.
- [22] Coughlin, R. y Driscoll F.: *Amplificadores Operacionales y Circuitos Integrados Lineales*. 5ª Edición, Prentice Hall Hispanoamericana, México, 2000.
- [23] Fairchild Semiconductor: General Purpose 6-Pin Phototransistor Optocoupler 4N25. Fairchild Semiconductor International Inc., USA, Mayo de 2003.
- [24] Boylestad, R. y Nashelsky, L.: *Electrónica: Teoría de Circuitos*. 6ª Edición, Prentice Hall Hispanoamericana, México, 1993.
- [25] Maxim/Dallas: Line-Powered, RS-232 Transceiver Chip DS275. Maxim Integrated Products Inc., USA, 2003.
- [26] Maxim/Dallas: Low Power, Slew-Rate-Limited RS-485/RS-422 Transceiver Max489. Maxim Integrated Products Inc., USA, Junio de 2003.
- [27] Maxstream: Xstream Wireless OEM Operation Manual V 3.0 Part Number MD002 V3.0. Maxstream Inc., USA, Junio de 2002.
- [28] SquareD: Power-Logic® Power Meter clase 3020: Manual de Instrucciones n° 3020IB9904. SquareD Company, USA, Diciembre de 1998.
- [29] Ritter, D.: *LabVIEW GUI: Essential Techniques*. McGraw Hill, USA, 2002.
- [30] National Instruments: G Programming Reference Manual, National Instruments Corporation, USA, 1998.

- [31] Pressman, R.: *Ingeniería del Software: Un Enfoque Práctico*. 5ª Edición, McGraw Hill Interamericana de España, España, 2001.
- [32] Sommerville, I.: *Ingeniería de Software*. 6ª Edición, Pearson Education, México, 2002.
- [33] National Instruments: *LabVIEW Básico I, Manual*. National Instruments Corporation, USA, 1998.
- [34] Krug, S.: *Don't Make Me Think*. New Riders, USA, 2000.
- [35] Ávila, J.: *Medidor de Potencia Trifásico y Calidad de la Energía*. Tesis de Licenciatura, Universidad Tecnológica de la Mixteca, México, 2003.
- [36] López, Z.: *Diseño y Construcción de un Módulo de Almacenamiento para el Medidor de Variables Eléctricas POWER LOGIC PM 650 Basado en el Microcontrolador 8031*. Tesis de Licenciatura, Universidad Tecnológica de la Mixteca, México, 2002.

Internet

- [URL1] <http://www.cfe.gob.mx>, Comisión Federal de Electricidad, “Consumo de Electricidad en México”, “Demanda Máxima Medida”, 1 de Julio de 2004.
- [URL2] http://en.wikipedia.org/wiki/ISM_band, wikipedia.org, “ISM Band”, 1 de Agosto de 2004.
- [URL3] <http://www.ni.com/labview>, National Instruments, “LabVIEW graphical development”, 1 de Agosto de 2004.
- [URL4] <http://www.agilent.com/>, Agilent Technologies, “VEE Pro”, 1 de Agosto de 2004.
- [URL5] <http://www.directindustry.com/>, Direct Industry: The Virtual Industrial Exhibition, “Power Meters”, 1 de Agosto de 2004.
- [URL6] <http://www.conae.gob.mx>, Comisión Nacional para el Ahorro de la Energía, “Sistemas de Control de la Demanda”, 1 de Agosto de 2004.
- [URL7] <http://www.obvius.com/documentation/technotes>, Obvius® Documents: Technical Notes, “Pulse Output Energy Meters”, 1 de Agosto de 2004.
- [URL8] <http://www.dencorinc.com/kyz.htm>, Dencor: Energy Control Systems, “Using KYZ Pulses to Monitor Demand”, 1 de Agosto de 2004.
- [URL9] <http://www.modbus.org>, MODBUS-IDA: the Architecture for Distributed Automation, “MODBUS Standard Library”, 1 de Agosto de 2004.
- [URL10] <http://www.atmel.com>, Atmel Corporation, “AVR Studio”, “AVR-Assamblar”, 1 de Agosto de 2004.
- [URL11] <http://cobalt.et.tudelft.nl/~glas/ssc/techn/techniques.html>, Jack P.F. Glas, “Principles of Spread Spectrum Communications”, 20 de Agosto de 2004.
- [URL12] <http://www.ni.com/latam>, National Instruments de Latinoamérica, “Desarrollo Gráfico en LabVIEW”, 1 de Agosto de 2004.
- [URL13] http://www.useit.com/papers/heuristic/heuristic_list.html, Jakob Nielsen, “Ten Usability Heuristics”, 1 de Agosto de 2004.

Apéndice A. El microcontrolador AT90S2313

El AT90S2313, cuyo diagrama de conexiones aparece en la *figura A.1*, es un microcontrolador de la firma ATMEL que incluye varias características destacadas de la arquitectura 8-BIT AVR-RISC también desarrollada por esta firma. A continuación se enumeran las características eléctricas funcionales y programables que presenta este microcontrolador.

A.1. Características funcionales

- Puede ser alimentado en un rango de voltajes desde 2.6 hasta 6.0 V de CD.
- Posee un repertorio de 120 instrucciones, la mayoría de ellas ejecutadas en un solo ciclo de reloj, por ejemplo: para un cristal de 20 MHZ (máximo soportado por este microcontrolador), las instrucciones se ejecutan en un tiempo de 50 ns.
- Cuenta con memoria:
 - FLASH de 2KB para el código del programa.
 - RAM de 128 bytes para almacenamiento temporal de datos.
 - EEPROM de 128 bytes para almacenamiento no volátil de datos.
- Cuenta con 32 registros de propósito general que pueden ser acumuladores de datos y 16 de ellos, ejecutan instrucciones inmediatas.

A.2. Características programables

- 15 líneas de E/S, de las cuales 13 tienen funciones alternativas programables.
- 1 UART para comunicación dúplex bajo el estándar RS-232 con la PC u otros dispositivos.
- 2 temporizadores/contadores (*timer/counter*), uno de 8 y otro de 16 bits con escalas de conteo independiente.
- 1 modulador de ancho de pulso (PWM, *Pulse Width Modulator*) de 8, 9 ó 10 bits que se construye a partir del *timer/counter* de 16 bits.

- 2 interrupciones externas para el control de eventos en un *pin* de entrada.
- Un *watchdog-timer* con oscilador independiente que permite reiniciar el microcontrolador después de un tiempo determinado.
- Un comparador analógico que activa una interrupción después de un evento de comparación de 2 señales analógicas.
- Permite ahorrar consumo de energía utilizando alguno de sus dos modos de bajo consumo, el inactivo o el dormido.

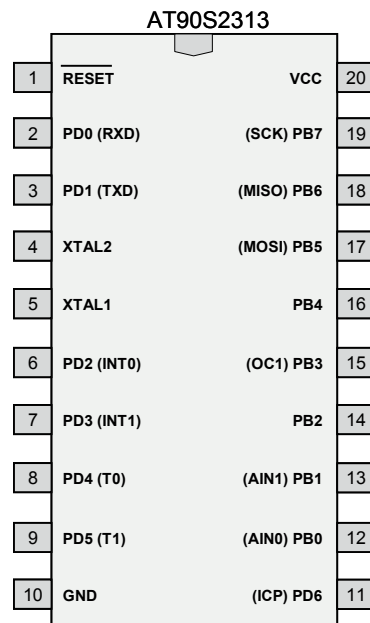


Figura A.1. Configuración de los pines del AT90S2313.

Apéndice B. Manual de usuario del CODKYZ

Introducción

En este manual se describen las características del CODKYZ que lo constituyen como un dispositivo listo para instarse y operar en campo dentro de una red MODBUS.

Se recomienda ampliamente leer este manual y examinar el Codificador de Pulsos KYZ (CODKYZ) antes de instalarlo o hacerlo operar para familiarizarse con él y prevenir que el mal uso de este dispositivo provoque su malfuncionamiento, averío o descomposición total.

A lo largo de este manual se emplea la siguiente simbología para dar a conocer al usuario alguna sugerencia o aviso importante que debe tomar en cuenta en el uso, instalación y/o conexión del CODKYZ.



- **Referencia:** se utiliza para señalar alguna otra parte de este manual donde se da una descripción más amplia de algún tema en particular.



- **Información:** se utiliza para dar alguna información o sugerencia importante sobre un tema en particular referente al uso del equipo.



- **Advertencia:** se utiliza para anticipar al usuario sobre algún procedimiento incorrecto que pueda dañar al equipo.



- **Peligro:** se utiliza para avisar al usuario sobre algún peligro hacia su persona por no seguir las instrucciones adecuadamente. El incumplimiento de una instrucción de **Peligro** puede causar la **muerte o lesiones severas** en la integridad física de la persona.

B.1. Descripción general del CODKYZ

El CODKYZ es un dispositivo electrónico compacto que sirve como sistema alternativo para obtener información del consumo eléctrico que registra una planta. Su función es convertir las conmutaciones de los relevadores KYZ de los medidores electrónicos de energía a señales eléctricas que puedan ser interpretadas y codificadas electrónicamente. Los datos codificados de los pulsos KYZ se almacenan en la memoria del dispositivo para que sean consultados preferentemente por un programa de computadora y sirvan como información para calcular las variables de consumo eléctrico de la planta.

La consulta de los datos se lleva a cabo a través del protocolo de comunicaciones MODBUS, el cual es soportado por el CODKYZ, de modo que puede ser incluido dentro de una red industrial que opere bajo este protocolo.

B.1.1. Características físicas

El circuito electrónico del CODKYZ tiene como chasis de protección un gabinete color gris claro con dimensiones: 7x12x6 cm (ancho x largo x alto).

B.1.2. Características eléctricas

El CODKYZ cuenta con las características eléctricas citadas en la *tabla B.1*.

Tabla B.1. Características eléctricas del CODKYZ.



Parámetro	Mínimo	Típico	Máximo	Unidades
Voltaje de alimentación	4.8	5	5.25	V
Corriente de alimentación	9	14	21	mA
Voltaje para pulsos KYZ	4.5	5	6	V
Corriente para pulsos KYZ	10	19	38	mA



Los valores de la tabla anterior representan los requerimientos de voltaje y corriente del CODKYZ. Sin embargo, éste ya cuenta con un módulo de alimentación independiente que le proporciona esos parámetros eléctricos y que requiere conexión a una toma eléctrica de 120 V de CA. Consume 100 mA máximo.

B.1.3. Identificación de componentes

En la *figura B.1* se ilustra el CODKYZ señalando sus componentes físicos que se enumeran y describen a continuación:

1. Módulo externo para la alimentación de voltaje de CD (2 fuentes de poder).
2. Terminales con tornillos (PKWh, PKVARh y PSINC) para conectar las terminales K-Y de los relevadores KYZ del medidor electrónico.  *Conexión de los Pulsos KYZ (apartado B.2.2).*
3. Puerto RS-232 con conector DB9 hembra de conexión directa: (2) TXD, (3) RXD y (5) COM (únicas señales empleadas del conector).  *Uso y conexión del puerto RS-232 (inciso B.2.3.1).*

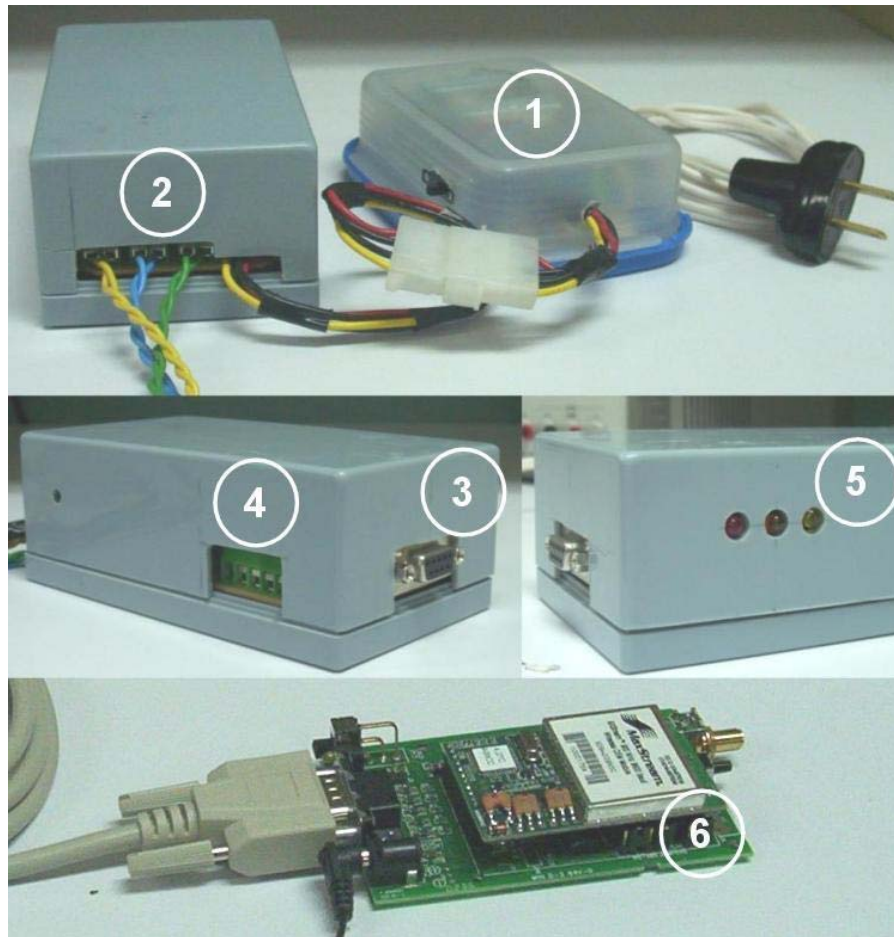


Figura B.1. Ubicación de los componentes del CODKYZ.

4. Puerto RS-485 de terminales con tornillos para enlace a 4 y 2 hilos. *Uso y conexión del puerto RS-485 (inciso B.2.3.2).*
5. LEDS indicadores de funcionamiento. *Descripción de los LEDS indicadores (apartado B.3.9).*
6. Radio-módem externo modelo 9XStream de 900MHz de la firma MaxStream®. *Conexión del radio-módem 9XStream (inciso B.2.3.3).*

B.2. Instalación y conexión

B.2.1. Dónde instalar el CODKYZ

El lugar más adecuado para instalar el CODKYZ es en la acometida eléctrica de la planta que se va a analizar, pues es ahí donde generalmente se encuentra instalado el medidor electrónico del cual se necesitarán sus relevadores KYZ para conectarlo.



Antes de usar los relevadores KYZ del medidor electrónico que su proveedor de electricidad (CFE) instala en la acometida para propósitos de facturación, usted debe contar con el permiso correspondiente para realizar las conexiones que el CODKYZ requiere para operar. También es válido emplear un medidor electrónico comercial que posea relevadores KYZ de salida.



Antes de instalar el CODKYZ, es muy recomendable que se emplee algún chasis de protección extra a base de metal, plástico o nylon para protegerlo contra la intemperie: lluvia, humedad, contacto directo con el sol, etc.

B.2.2. Conexión de los pulsos KYZ



Para evitar daños físicos personales por descargas eléctricas o cortos circuitos, se recomienda ampliamente que la instalación/conexión del CODKYZ la lleve a cabo una persona experimentada en instalaciones eléctricas.

El CODKYZ es un sistema electrónico cuyas entradas se conectan directamente a los relevadores KYZ de salida de los medidores electrónicos de energía. Para ello es necesario destapar el chasis del CODKYZ y atornillar los cables que provienen de las terminales del relevador a los conectores con tornillo del CODKYZ (PKWh, PKVARh y PSINC).

En la *figura B.2* se ilustra un ejemplo en donde se conectan las terminales K-Y del medidor de variables eléctricas PM650 de la marca SquareD® a la entrada PKWh del CODKYZ. La orientación de las terminales K-Y es indistinta, lo importante es conectar las terminales de cada relevador a la entrada que le corresponde de acuerdo a la *tabla B.2*.

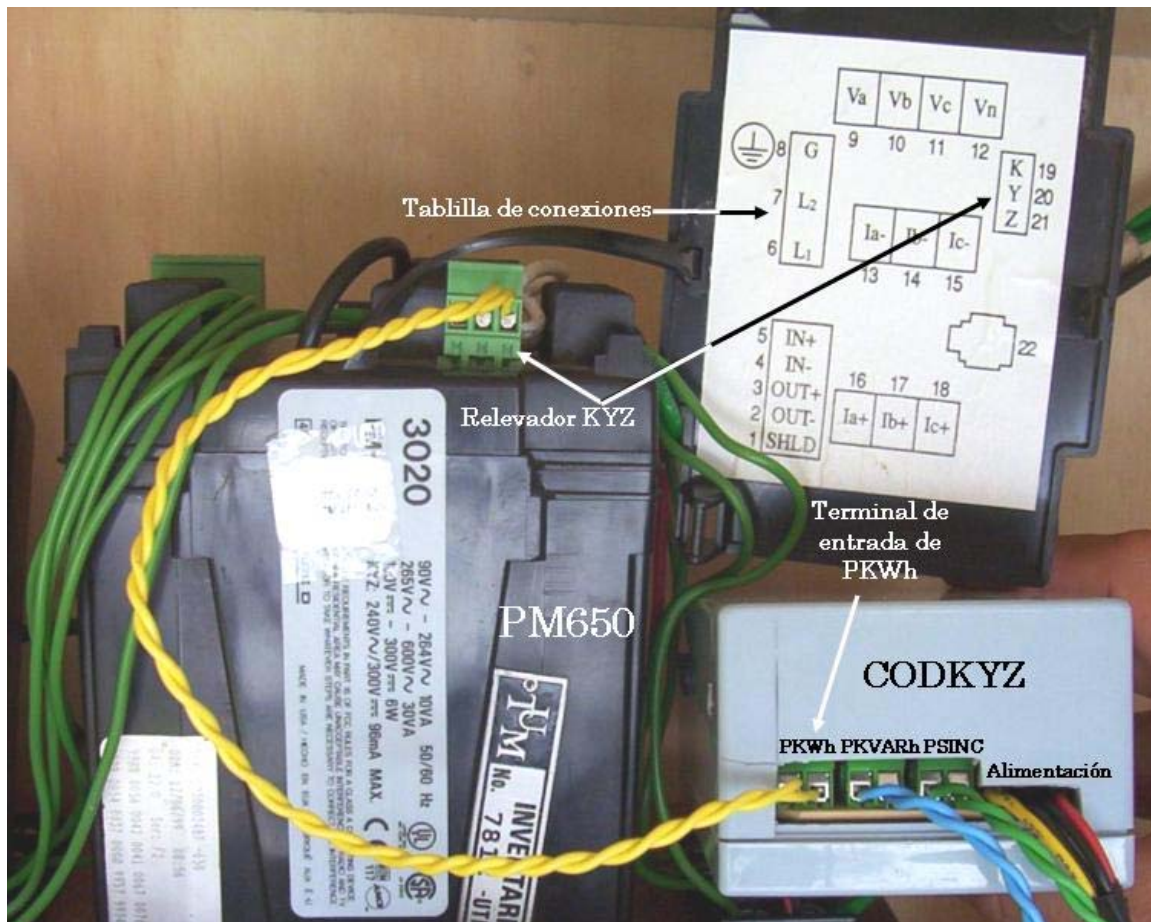


Figura B.2. Conexión de las terminales K-Y del PM650 a las entradas PKWh del CODKYZ.

Tabla B.2. Conexión entre los relevadores KYZ del PM650 y las entradas de pulsos del CODKYZ.

Relevador KYZ de:	Entrada del CODKYZ
Energía activa	PKWh
Energía reactiva	PKVARh
Sincronía	PSINC



Asegúrese de realizar la conexión tal y como se señala en la *tabla B.2*, ya que si ésta no se respeta, la información de los pulsos KYZ almacenada en el CODKYZ no será la correcta y, como consecuencia, los valores de los parámetros de consumo eléctrico calculados serán erróneos.


B.2.3. Conexión de los puertos de comunicaciones

El CODKYZ cuenta con dos puertos de comunicación serial para enlace alambrado que son: el puerto RS-232 para conexión punto a punto y el puerto RS-485 para conexión multipunto.

Para cambiar entre un estándar de comunicación y otro es necesario destornillar el chasis del CODKYZ y cambiar la posición de los **dos puentes** que seleccionan el estándar deseado, tal y como se ilustra en la *figura B.3*.

Si el estándar elegido es el RS-485, primero se conectan los cables de las señales del estándar a los conectores con tornillos correspondientes y después se vuelve a montar el chasis.

B.2.3.1. Uso y conexión del puerto RS-232

Este puerto se ocupa para establecer una comunicación de conexión directa con la PC o para conectar el radio-módem y establecer una conexión inalámbrica, con la cual se puede ser parte de una red de uno o más Esclavos.  *Conexión del radio-módem 9XStream (inciso B.2.3.3).*

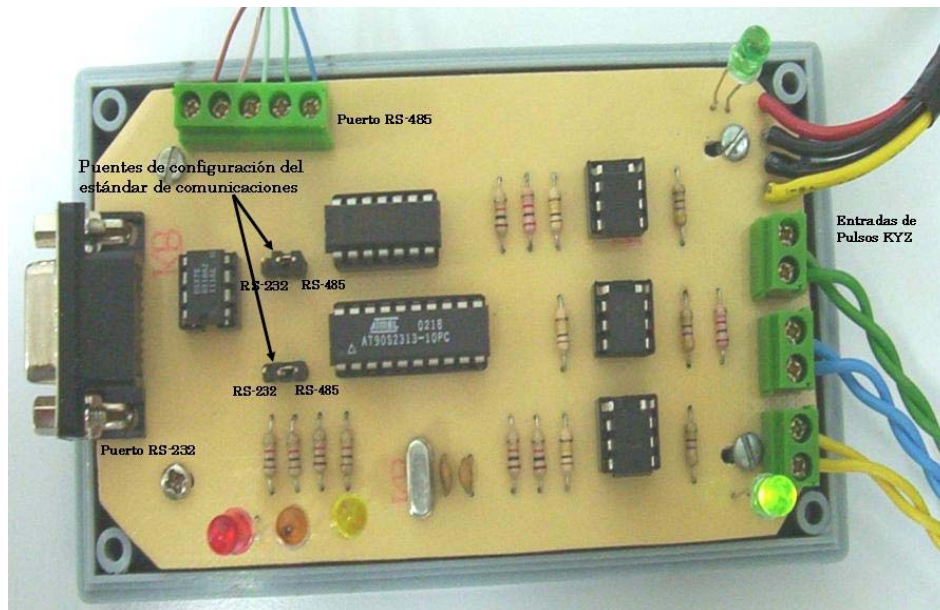


Figura B.3. Ubicación de los puentes que seleccionan el estándar de comunicaciones.

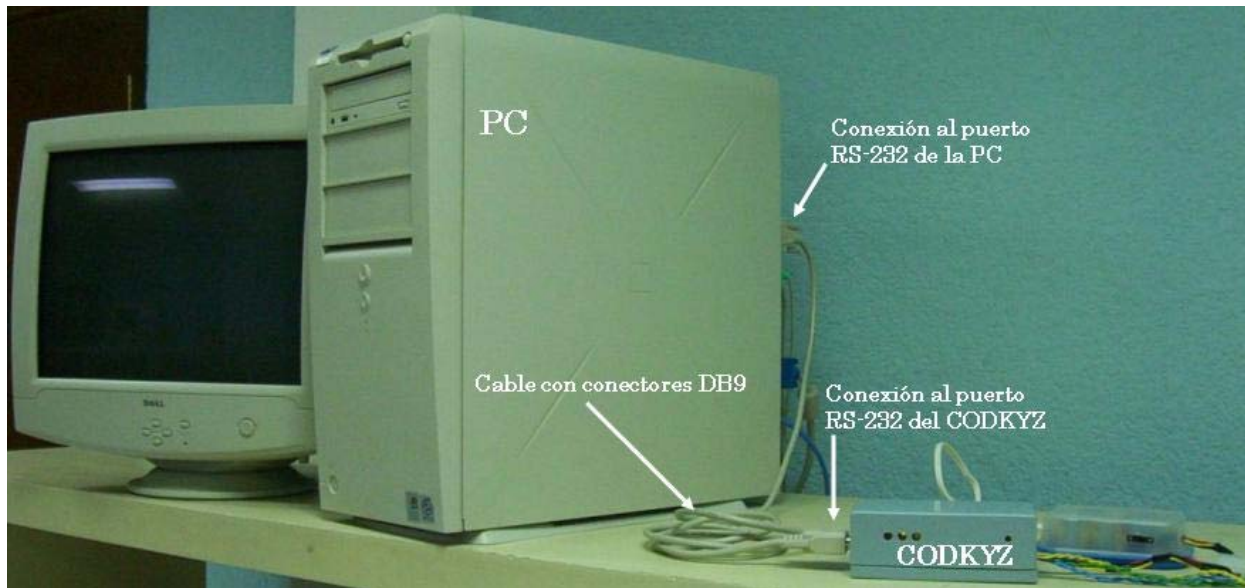


Figura B.4. Conexión del CODKYZ y la computadora a través del puerto RS-232.

En la *figura B.4* se ilustra la conexión del puerto RS-232 del CODKYZ a la PC, en la cual se emplea un cable con un conector DB9 hembra en el extremo de la PC y un DB9 macho en el extremo del CODKYZ.

B.2.3.2. Uso y conexión del puerto RS-485

El puerto RS-485 permite al CODKYZ incluirse dentro de una red MODBUS RS-485 con conexión alambrada a distancia a través de un bus de 4 hilos (4W) o de 2 hilos (2W). Cabe hacer la aclaración que en ambos casos existe además una señal de referencia común (C, GND o SHIELD) que, aunque no se cuenta en el nombre (2W y 4W), siempre se conecta.

B.2.3.2.1. Conexión al bus de 4 hilos

En la *figura B.5* se muestra el diagrama de conexión de las señales del puerto RS-485 del CODKYZ a una red MODBUS 4W.

B.2.3.2.2. Conexión al bus de 2 hilos

En la *figura B.6* se muestra el diagrama de conexión de las señales del puerto RS-485 del CODKYZ a una red MODBUS 2W. Como puede observarse en la *figura B.7*, basta con interconectar las señales RD+ con TD+ y RD- con TD- para obtener las señales B+ y A- que marca la norma RS-485.

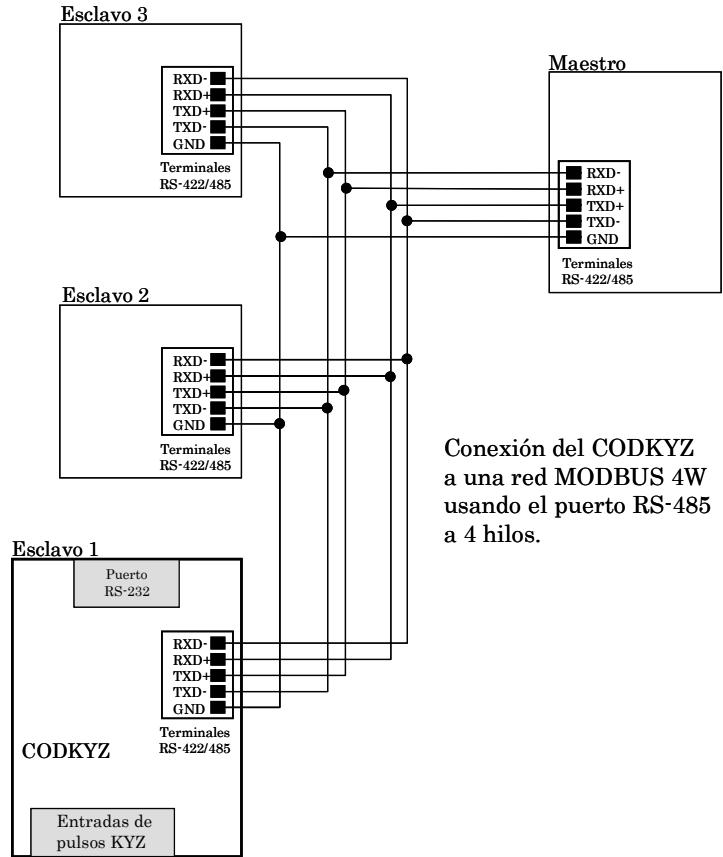


Figura B.5. Conexión del CODKYZ a una red MODBUS 4W a través del puerto RS-485 de 4 hilos.

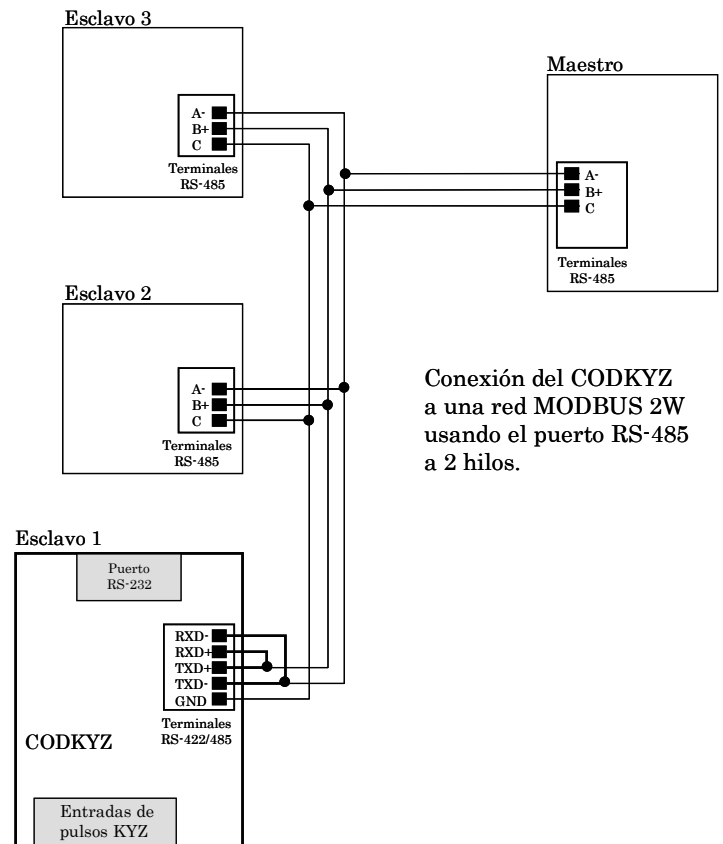


Figura B.6. Conexión del CODKYZ a una red MODBUS 2W a través del puerto RS-485 de 2 hilos.

B.2.3.3. Conexión del radio-módem 9XStream



Los radio-módems 9XStream que se emplean para el enlace inalámbrico entre el CODKYZ y la Estación Maestra, vienen previamente configurados para operar a una velocidad de 9600 baudios, por lo que el parámetro velocidad de transmisión/recepción del CODKYZ debe configurarse con este valor.

Para conectar el radio-módem 9XStream al CODKYZ, se emplea un cable con un conector DB9 macho en cada extremo, o bien puede emplearse un convertidor de hembra a macho, como es el caso de la conexión mostrada en la *figura B.7*. Los radio-módems se alimentan a través de un eliminador de pilas que se conecta a la toma de corriente de 120V de CA. Para mayor detalle del uso del radio-módem 9XStream se recomienda leer el manual de usuario correspondiente.

B.2.3.4. Conexión e instalación de la antena

Los radio-módems modelo 9XStream cuentan con un conector SMA (*figura B.7*) al cual se puede conectar su antena externa a través de cable coaxial. Estas antenas tienen un alcance de hasta 5 Kms cuando existe línea de vista entre ellas. En la *figura B.8* se muestra la antena del radio-módem instalada en campo.

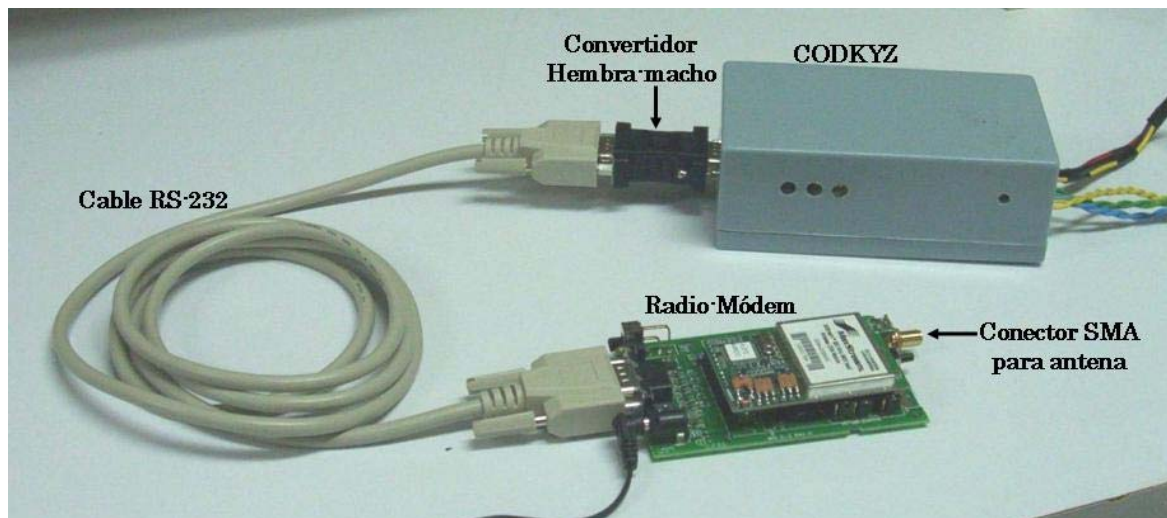


Figura B.7. Conexión del CODKYZ al radio módem a través del puerto RS-4232.



Figura B.8. Antena empleada para el enlace entre los radio-módems.

B.3. Soporte del protocolo MODBUS

El CODKYZ se diseñó para ser incluido dentro una red de comunicaciones que opere bajo el protocolo de comunicaciones MODBUS. En esta sección se presentan las características del protocolo MODBUS implementadas en el CODKYZ.

B.3.1. Parámetros de comunicación MODBUS

Tabla B.3. Parámetros de comunicación MODBUS del CODKYZ.

Parámetro	Valor
Modo	RTU
Velocidad	1200, 2400, 4800, 9600 y 19200 baudios
Bits de datos	8
Señalización	1 bit de inicio, 1 bit de fin
Paridad	Ninguna
Estándar eléctrico	RS-232 y RS-485
Tipo de comunicación	Semidúplex

B.3.2. Funciones MODBUS soportadas

Tabla B.4. Funciones MODBUS soportadas por el CODKYZ.

Decimal	Hex	Descripción de la función
3	03	Lectura del contenido de los registros mantenidos
4	04	Lectura del contenido de los registros de entrada
6	06	Escritura del valor de un registro mantenido
8	08	Función de diagnóstico (ver subfunciones de diagnóstico)
11	0B	Lectura del contador de comunicaciones exitosas y del registro de estado
16	10	Escritura del valor de múltiples registros mantenidos
23	17	Escritura y lectura de múltiples registros mantenidos o de entrada
43	2B	Lectura de los datos de identificación del dispositivo
65*	41	Escribe los registros mantenidos en la EEPROM.
69*	45	Reinicializa los registros de entrada con el valor 0.

*Función definida por el usuario, no especificada en la norma del protocolo MODBUS.

B.3.3. Subfunciones de diagnóstico MODBUS

Tabla B.5. Subfunciones de diagnóstico (función 08) MODBUS soportadas por el CODKYZ.

Decimal	Hex	Descripción de la subfunción
0	00	Solicitud de eco
1	01	Reinicio de comunicación y de contadores de eventos de comunicación
2	02	Retornar el registro de diagnóstico
4	04	Activar el modo sólo escuchar (MSE) del dispositivo
10	0A	Reinicio de contadores de eventos de comunicación
11	0B	Retornar el registro contador de mensajes en bus (MENSBUS)
12	0C	Retornar el registro contador de errores de comunicación (CRCERR)
13	0D	Retornar el registro contador de repuestas de excepción (RESPEXC)
14	0E	Retornar el registro contador de mensajes hacia el esclavo (MENRESP)
15	0F	Retornar el registro contador de mensajes de difusión (MENSdif)

B.3.4. Códigos de excepción MODBUS

Tabla B.6. Códigos de excepción MODBUS soportadas por el CODKYZ.

Decimal	Hex	Descripción del código
1	01	Se ha solicitado al dispositivo ejecutar una función que no soporta
2	02	Se ha intentado leer o escribir una dirección ilegal
3	03	El valor que se desea escribir en un registro no es válido

B.3.5. Registros mantenidos

Tabla B.7. Registros mantenidos (configuración) del CODKYZ.

Dirección lógica	Dirección MODBUS	L/E	Nombre	Descripción del registro
40001	0000	L-E	AÑO-MES	Año y Mes (Reloj/Calendario)
40002	0001	L-E	DÍA-HORA	Día y Hora (Reloj/Calendario)
40003	0002	L-E	MINS-SEGS	Minutos y Segundos (Reloj/Calendario)
40004	0003	L-E	DIRECCIÓN	Dirección de Esclavo del dispositivo
40005	0004	L-E	BAUDRATE	Velocidad (1200,2400,4800,9600 ó 19200 baudios)
40007	0006	L-E	INTERV	Intervalo de medición de pulsos KYZ (100 ó 1000 ms)
40006	0005	L-E	RETARDO	Tiempo programable de respuesta (0, 50 ó 100 ms)
40008	0007	L-E	EEPROM	Indica si la EEPROM almacena datos (0 = NO, 170 = Sí)

B.3.6. Valores por omisión de los registros mantenidos

Tabla B.8. Valores por omisión de los registros mantenidos del CODKYZ.

Registro	Valor (HEX)	Valor (DEC)	Descripción
AÑO-MES	0x0001	0	2000-ENERO-1 00:00:00 hrs.
DÍA-HORA	0x0100	0	
MINS-SEGS	0x0000	0	
DIRECCIÓN	0x0001	1	Esclavo 1
BAUDRATE	0x2580	9600	9600 baudios
INTERV	0x0064	1000	1000 ms
EEPROM	0x0066	102	EEPROM vacía

B.3.7. Formato comprimido de fecha y hora

La fecha y hora del sistema se almacenan en tres registros mantenidos (0000-0002) y se comprimen de la siguiente manera:

- Registro 0000 (AÑO-MES): *Año* (byte 1) = 1–12, *Mes* (byte 2) = 1–31
- Registro 0001 (DÍA-HORA): *Día* (byte 1) = 0–31, *Hora* (byte 2) = 0–23,
- Registro 0002 (MINS-SEGS): *Mins.* (byte 1) = 0–59, *Segs.* (byte 2) = 0–59.

Año es cero en base al año 2000. Por ejemplo, 07 de Agosto de 2004 a las 10:55:42 hrs. se representa (en hexadecimal) como:

- AÑO-MES: **0x0408**, DÍA-HORA: **0x070A**, MINS-SEGS: **0x372A**

B.3.8. Registros de entrada

Tabla B.9. Registros de entrada del CODKYZ.

Lógica	MODBUS	L/E	Nombre	Descripción de registro
30009	0008	L	KWH	Cantidad de pulsos KYZ de KWH (PKWh)
30010	0009	L	Tiempo1	Intervalos de medición después del último PKWh
30011	000A	L	KVARH	Cantidad de pulsos KYZ de KVARH (PKVARh)
30012	000B	L	Tiempo2	Intervalos de medición después de último PKVARh
30013	000C	L	SINC1	Día, Hora, Minutos y segundos en que ocurrió el último pulso KYZ de sincronía (PSINC)
30014	000D	L	SINC2	

B.3.9. Descripción de los LEDS indicadores

Tabla B.10. LEDS indicadores de funcionamiento del CODKYZ.

Color	Indicación
Verde	Dispositivo encendido y operando normalmente
Amarillo	El CODKYZ está recibiendo datos seriales
Naranja	El CODKYZ está transmitiendo datos seriales
Rojo	Se presentó un error de comunicación: CRC erróneo o una respuesta de excepción.

B.4. Diagrama esquemático

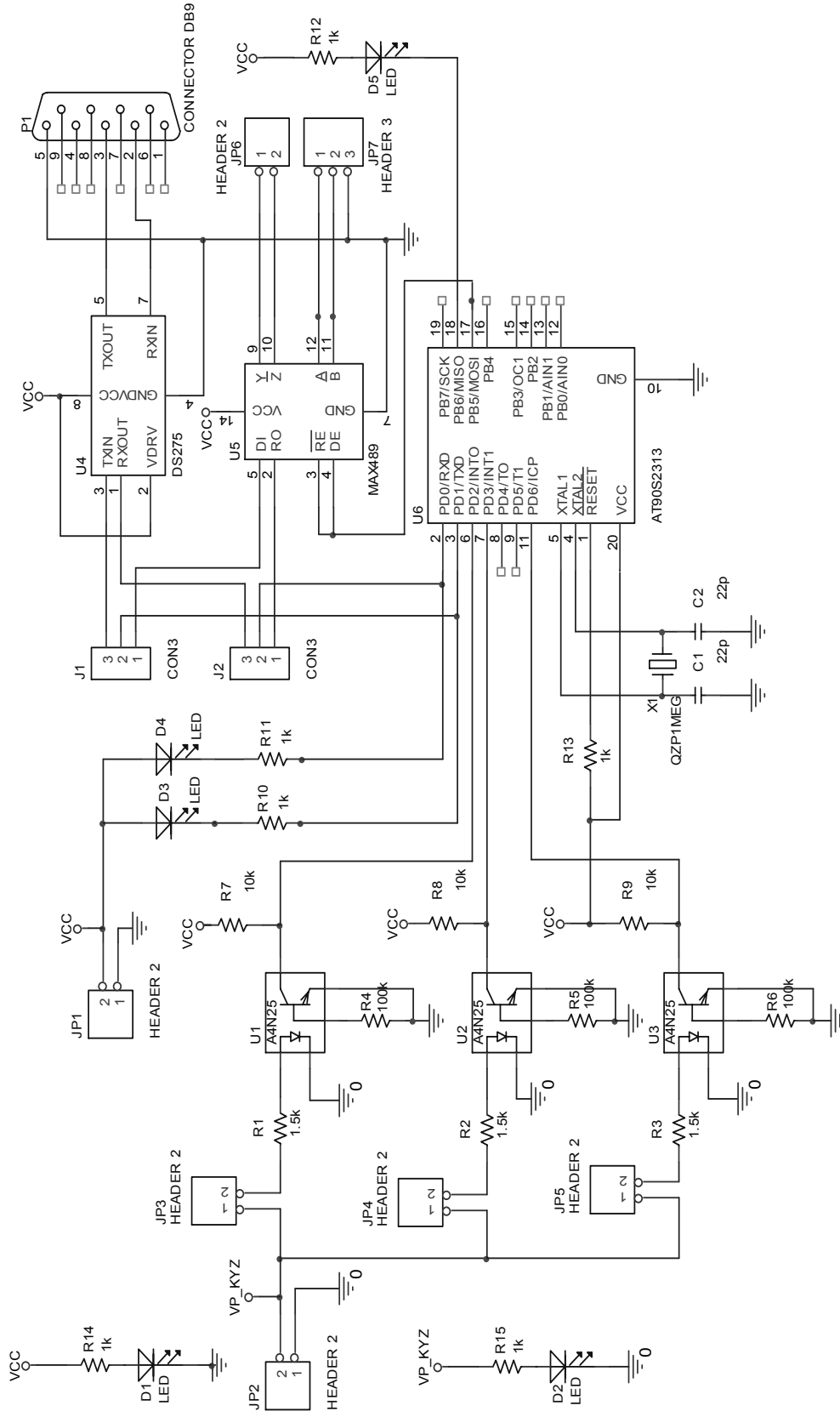


Figura B.9. Diagrama esquemático del CODKYZ.

B.5. Lista de componentes

Tabla B.11. Lista de componentes del CODKYZ.

No. de parte	Cantidad	Referencia	Descripción	Modelo
1	2	C1, C2	Capacitor de 22 pF	C.022-50
2	4	D1, D2, D3, D4	LED	E5/AMB-D
3	6	JP1, JP2, JP3, JP4, JP5, JP6	Terminal doble con tornillos	TRT-02
4	1	JP7	Terminal triple con tornillos	TRT-03
5	2	J1, J2	Puente de tres posiciones	F36-S
6	1	P1	Conector DB9 hembra de montaje superficial	500-020
7	3	R1, R2, R3	Resistor de 470 Ω , 0.5 W	R470
8	3	R4, R5, R6	Resistor de 100 k Ω , 0.5W	R100K
9	3	R7, R8, R9	Resistor de 10 k Ω , 0.5 W	R10K
10	5	R10, R11, R12, R13	Resistor de 1 k Ω , 0.5W	R1K
11	3	U1, U2, U3	Optoacoplador con salida a transistor	4N25
12	1	U4	Convertidor RS-232 a TTL	DS275
13	1	U5	Convertidor RS-485 a TTL	MAX489
14	1	U6	Microcontrolador AVR 8-bit RISC	AT90S2313
15	1	X1	Cristal de cuarzo de 4MHz	C4.0
Total:	37			

B.6. Diagrama de circuito impreso

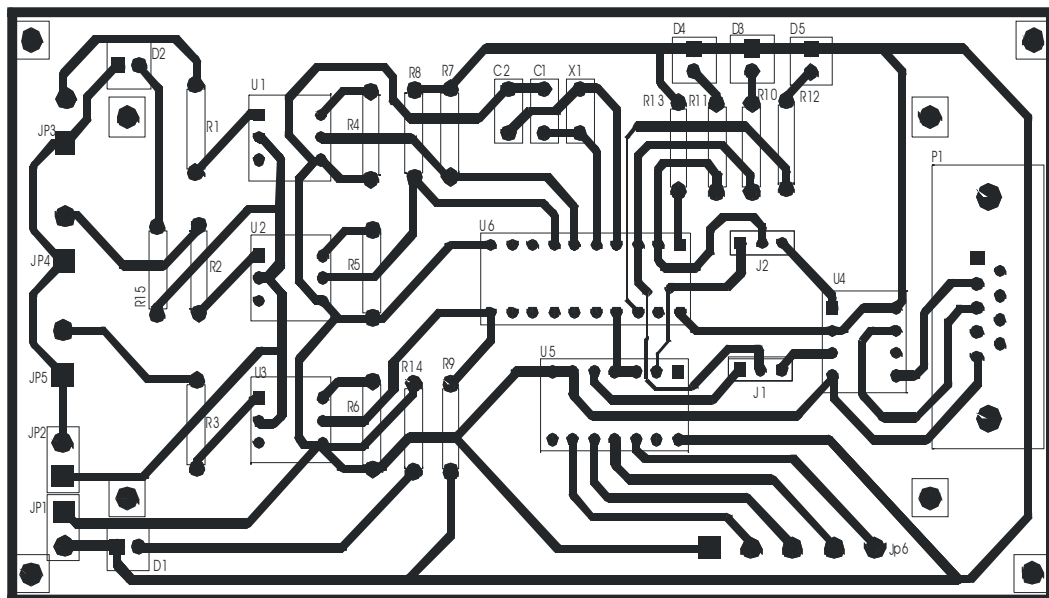




Figura B.10. Diagrama de circuito impreso del CODKYZ.

B.7. Solución de problemas comunes

Tabla B.12. Problemas más comunes que se presentan al usar el CODKYZ y cómo se resuelven.

Problema	Causa	Solución
El CODKYZ no responde a las solicitudes MODBUS que se le realizan.	El cable de comunicación no está conectado a la red RS-485 o al puerto RS-232 de la PC.	Realice la conexión del CODKYZ a la red de manera correcta. Reemplace el cable en caso de que éste se encuentre roto. ☞ <i>Conexión de los puertos de comunicaciones (apartado B.2.3).</i>
	Se está trabajando con un estándar de comunicaciones diferente al de la configuración de la red.	Coloque los puentes que seleccionan el estándar eléctrico de comunicación en las posiciones correctas y realice las conexiones de acuerdo al estándar con el que está trabajando (RS-232 ó RS-485). ☞ <i>Conexión de los puertos de comunicaciones (apartado B.2.3).</i>
	Los parámetros de comunicación como dirección del dispositivo o baudrate no están configurados adecuadamente.	Asegúrese de que la dirección de dispositivo y la velocidad de transmisión y recepción tengan los valores correctos. ☞ <i>Parámetros de comunicación MODBUS (apartado B.3.1).</i> ☞ <i>Valores por omisión de los registros mantenidos (apartado B.3.6).</i>
	Si está usando una comunicación inalámbrica, además de verificar los casos anteriores, cerciórese de que el radio-módem opera normalmente.	Consulte el manual de operación correspondiente.
Aunque responde a las solicitudes MODBUS, el CODKYZ no reporta información de ocurrencia de pulsos después de mucho tiempo	Es posible que la planta haya disminuido considerablemente su consumo eléctrico y, por tanto, la frecuencia de los pulsos es baja.	En el software del Maestro es necesario aumentar el tiempo entre cada consulta de datos.
	Los cables que se conectan de las terminales K-Y a las terminales con tornillo del CODKYZ no coinciden con la conexión correcta.	Conecte las terminales K-Y de los relevadores a las terminales correctas del codificador. ☞ <i>Conexión de los pulsos KYZ (apartado B.2.2).</i> ☞ <i>Tabla B.2.</i>
	El módulo de alimentación de los pulsos KYZ está apagado.	Verifique el módulo de alimentación directamente en campo y, si está apagado, encienda el módulo. Si el problema persiste o el módulo de alimentación no enciende, es muy probable que el CODKYZ o que su módulo de alimentación hayan sufrido alguna avería. Consulte al fabricante: avefiles@yahoo.com.

<p>Se reportan pulsos KYZ, pero los valores parecen ser incorrectos.</p>	<p>Los cables que se conectan de las terminales K-Y a las terminales con tornillo del CODKYZ no coinciden con la conexión adecuada.</p>	<p>Conecte las terminales K-Y de los relevadores a las terminales correctas del codificador.  <i>Conexión de los pulsos KYZ (apartado B.2.2).</i>  <i>Tabla B.2.</i></p>
--	---	--

Si después de hacer las correcciones anteriores el problema persiste: Consulte al fabricante: avefiles@yahoo.com.

