



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

“Monitoreo y control a distancia de un interruptor de circuito de distribución,
mediante el microcontrolador MSP430 de Texas Instruments”

Tesis para obtener el título de
Ingeniero en Electrónica

Presenta
Silva López Gaudencio.

Director
M.C. Felipe Santiago Espinosa

Asesor
Ing. Lucio Rojas García

Huajuapán de León, Oaxaca. Marzo del 2010.

Tesis presentada el 19 de marzo del
2010 ante los siguientes sinodales:

M.C. ENRIQUE ESPINOSA JUSTO.
M.C. JORGE LUIS BARAHONA AVALOS.
DR. MARCO ANTONIO CONTRERAS ORDAZ.

Director de Tesis:

M.C. FELIPE SANTIAGO ESPINOSA.

DEDICATORIA

A mi padre Sr. Gaudencio Silva López, porque gracias a sus enseñanzas y apoyo incondicional me fue posible concluir de manera satisfactoria mis estudios.

A mis Hermanos Yolanda, Elvira, Carmen, Víctor, Gaby y Sergio, por darme su cariño y apoyo en todo momento.

Gaudencio Silva López

A G R A D E C I M I E N T O S

A la Comisión Federal de Electricidad, por darme todas las facilidades para el desarrollar este proyecto.

Al Ing. Lucio Rojas García, por creer en mi propuesta y brindarme su apoyo durante el desarrollo del proyecto.

Al M.C. Felipe Santiago Espinosa, porque gracias a su apoyo y motivación me fue posible llevar a buen término mi tesis profesional.

A mis amigas, Anel y Michel por su amistad.

A mi amigo Nacho, por ser uno de mis más grandes amigos y por apoyarme a concluir mis cursos de inglés, así como por el apoyo que me brindo durante el desarrollo de mi tesis.

A mis compañeros de trabajo, Ing. José Carlos, Ing. Rubén, Ing., Nahúm e Ing. Oliver por compartirme sus conocimientos y su amistad.

A mis compañeros de la universidad, Adán, Edwin, Jairo, Liborio, Liliana, Marisol, Jaime y Erick, por compartirme su amistad y conocimientos.

ÍNDICE

RESUMEN	1
CAPÍTULO 1. INTRODUCCIÓN	3
1.1 SISTEMAS ELÉCTRICOS DE DISTRIBUCIÓN	3
1.2 PLANTEAMIENTO DEL PROBLEMA	4
1.3 JUSTIFICACIÓN	5
1.4 OBJETIVOS	6
1.5 METODOLOGÍA	7
1.6 ORGANIZACIÓN DEL DOCUMENTO	9
CAPÍTULO 2. MARCO TEÓRICO	11
2.1 SUBESTACIÓN DE DISTRIBUCIÓN	11
2.1.1 INTERRUPTOR DE POTENCIA	13
2.1.2 CLASIFICACIÓN DE LOS INTERRUPTORES	14
2.2 SISTEMAS EMPOTRADOS	16
2.3 TARJETA EASYWEB 3	19
2.3.1 REQUERIMIENTOS MÍNIMOS	20
CAPÍTULO 3. DESARROLLO	23
3.1 ESPECIFICACIONES	23
3.2 FORMATO INTERMEDIO	24
3.3 PARTICIÓN HARDWARE/SOFTWARE	25
3.4 DESARROLLO DE SOFTWARE	25
3.4.1 INTERFAZ DE USUARIO	25
3.4.2 MONITOREO Y CONTROL	27
3.4.2.1 Software de prueba de la tarjeta	28
3.4.2.1.1 Módulo de aplicación	28
3.4.2.1.2 Módulo TCP/IP	35
3.4.2.1.3 Módulo Ethernet	44
3.5 DESARROLLO DEL HARDWARE	45
3.5.1 ETAPA DE ACOPLAMIENTO PARA MANDOS	46
3.5.2 ETAPA DE ACOPLAMIENTO PARA MONITOREO	50
CAPÍTULO 4. PRUEBAS Y RESULTADOS	55
4.1 INTERFAZ DE USUARIO	55
4.2 MONITOREO Y CONTROL	57
4.3 ETAPAS DE ACOPLAMIENTO DE MONITOREO Y CONTROL	58
4.4 PRUEBA EN LA SUBESTACIÓN	61
4.4.1 CONEXIÓN DEL SISTEMA DE MONITOREO Y CONTROL AL TABLERO	62
CAPÍTULO 5. CONCLUSIONES Y TRABAJO FUTURO	65
5.1 CONCLUSIONES	65

5.2 TRABAJO FUTURO.....	66
BIBLIOGRAFÍA.....	67
APÉNDICE A. CÓDIGO HTML PARA LA PÁGINA DE INTERFAZ DE USUARIO.....	69
APÉNDICE B. AMBIENTE INTEGRADO DE DESARROLLO <i>IAR EMBEDDED</i> <i>WORKBENCH IDE</i>	71
APÉNDICE C. FUNCIONES DESARROLLADAS EN LENGUAJE C	75
APÉNDICE D. HARDWARE DESARROLLADO.	79

LISTA DE FIGURAS

Figura 1.1 Operación general del sistema de monitoreo y control.....	7
Figura 1.2 Flujo de diseño de un sistema empotrado.	8
Figura 2.1 Diagrama unifilar de una subestación.....	11
Figura 2.2 Transformador de potencia.	12
Figura 2.3 Interruptores de circuitos de distribución (Media Tensión).....	14
Figura 2.4 Arquitectura típica de un sistema empotrado.....	17
Figura 2.5 Tarjeta de desarrollo Easyweb 3	20
Figura 2.6 Programador JTAG para la descarga de programas.....	21
Figura 3.1 Diagrama a bloques de la operación funcional del sistema.	24
Figura 3.2 Página de interfaz de usuario.	27
Figura 3.3 Algoritmo del módulo de aplicación del software de prueba de la tarjeta Easyweb 3.	29
Figura 3.4 Identificación de valores dinámicos en código HTML.....	30
Figura 3.5 Asignación de puertos para lectura de parámetros y ejecución de mandos.	32
Figura 3.6 Algoritmo para la asignación de valores a cada variable, en función del estado de los parámetros del interruptor.....	33
Figura 3.7 Algoritmo para el proceso <i>Servidor HTTP</i>	34
Figura 3.8 Algoritmo para la función insertar valores dinámicos.	36
Figura 3.9 Algoritmo para el módulo TCP/IP.....	37
Figura 3.10 Algoritmo para la función que identifica y ejecuta mandos sobre el interruptor..	40
Figura 3.11 Archivo para modificar la dirección IP de la tarjeta Easyweb 3.....	43
Figura 3.12 Algoritmo para el módulo de software Ethernet.....	45
Figura 3.13 Diagrama a bloques de la etapa de acoplamiento para los mandos.	46
Figura 3.14 Configuración del temporizador 555 en modo monoestable.	47
Figura 3.15 Acoplamiento de voltajes e inversión de señales de mandos.	48
Figura 3.16 Configuración del Optoacoplador para protección del circuito lógico.	48
Figura 3.17 Relevador electromagnético y contacto NA para la salida de mando.....	49
Figura 3.18 Diagrama esquemático completo para la etapa de mandos.....	50
Figura 3.19 Diagrama a bloques de la etapa de acoplamiento para el monitoreo.....	50
Figura 3.20 Etapa para la protección galvánica del circuito.	51

Figura 3.21 Optoacoplador y bit para el bus de datos para el puerto 2.	52
Figura 3.22 Diagrama esquemático completo para el monitoreo de un parámetro.....	52
Figura 4.1 Diseño de la página de interfaz de usuario.	56
Figura 4.2 Captura de tramas enviadas para los mandos.....	56
Figura 4.3 Trama capturada por el software Ethernet para el mando Cerrar.	57
Figura 4.4 Circuito de prueba para simular los cambios de estado de los parámetros del interruptor y observar su correcta señalización en la página de interfaz de usuario.	58
Figura 4.5 Prueba realizada a la etapa de acoplamiento para el monitoreo del estado del interruptor.	59
Figura 4.6 Prueba realizada a la etapa de acoplamiento para la ejecución del mando <i>Bloquear</i>	60
Figura 4.7 Integración del sistema, para pruebas desde la página de interfaz de usuario.	61
Figura 4.8 Interruptor utilizado para probar el sistema de monitoreo y control.....	62
Figura 4.9 Interior de la caseta de control de la subestación Tlaxiaco.....	62
Figura 4.10 Tablilla de contactos al interior del tablero de control de la subestación.	63
Figura 4.11 Conexión del sistema al tablero de control del interruptor y la red de intranet de CFE, para realizar las pruebas de monitoreo y control vía IP.	64
Figura 4.12 Cuadro de alarmas del interruptor.....	64
Figura B.1 Creación de un “Workspace”.	71
Figura B.2 Agregar un proyecto existente a un <i>Workspace</i>	71
Figura B.3 Agregar y abrir un proyecto existente.	72
Figura B.4 Modificación de códigos fuente del proyecto.	72
Figura B.5 Propiedades del proyecto.....	73
Figura B.6 Seleccionar el microcontrolador con que cuenta la tarjeta Easyweb 3.	73
Figura B.7 Seleccionar el medio de programación y el puerto.	74
Figura D1 Diagrama completo para las etapas de acoplamiento de monitoreo y control de los parámetros de un interruptor de circuito de distribución.....	79
Figura D2 Distribución de componentes en <i>Layout Plus</i> para realizar el ruteo del circuito....	82
Figura D3 Ruteo de pistas para el circuito impreso.	83
Figura D4 Pistas del circuito impreso.	83
Figura D5 Distribución de componentes en la placa.....	84

LISTA DE TABLAS

Tabla 3.1 Tareas y medios de implementación.	25
Tabla 3.2 Estructura del software.	28
Tabla 3.3 Parámetros a monitorear y su variables relacionadas.	32
Tabla 3.4 Trama enviada para el mando <i>Abrir</i>	38
Tabla 3.5 Trama enviada para el mando <i>Cerrar</i>	39
Tabla 3.6 Trama enviada para el mando <i>Normalizar</i>	39
Tabla 3.7 Trama enviada para el mando <i>Bloquear</i>	39
Tabla 3.8 Relación entre los mandos y los bits del puerto 1.	39
Tabla 3.9 Valor del puerto 1 para cada mando.	41
Tabla 3.10 Trama enviada para el mando <i>Abrir</i>	41
Tabla 3.11 Patrón 1 en el buffer RX para la recepción de la trama para el mando <i>Abrir</i>	41
Tabla 3.12 Patrón 2 en el buffer RX para la recepción de la trama para el mando <i>Abrir</i>	42
Tabla 3.13 Identificación de los caracteres que indican el mando recibido en el buffer RX, para el patrón de recepción 1.	42
Tabla 3.14 Identificación de los caracteres que indican el mando recibido en el buffer RX, para el patrón de recepción 2.	42
Tabla 3.15 Relación entre el parámetro del interruptor y el número de bit del puerto 2.	52
Tabla D1 Lista de componentes.	79

RESUMEN

En el presente trabajo se realizó un sistema para el monitoreo y control a distancia de los parámetros de un interruptor de circuito de distribución de energía eléctrica, vía protocolo de Internet (IP), realizando su implementación en la tarjeta de desarrollo Easyweb 3, la cual brinda grandes beneficios para un proyecto de monitoreo y control, ya que cuenta con el microcontrolador MSP430 de Texas Instruments, periféricos de Entrada/Salida y una interfaz para conectarse a una red de área local. Además, se aprovecha la infraestructura con que actualmente cuenta la Comisión Federal de Electricidad (CFE): red de Intranet y enlaces de microondas y fibra óptica entre subestaciones, lo que hace este proyecto factible y rentable.

El sistema desarrollado consiste en monitorear y controlar un interruptor de circuito de distribución desde una página web; desde donde se pueden enviar mandos al interruptor, así como observar el estado de sus parámetros en tiempo real.

La página web es la interfaz de usuario, fue desarrollada en lenguaje HTML y es almacenada en la memoria del microcontrolador. La conexión a la red de intranet de la CFE se realiza por medio de la interfaz de red con que cuenta la tarjeta Easyweb 3.

El monitoreo y control lo realiza el microcontrolador, para lo cual se desarrollaron funciones en C; para leer el estado de los parámetros del interruptor, actualizarlos en la página de interfaz de usuario, así como para interpretar y ejecutar los mandos enviados al interruptor.

Fue necesario implementar etapas de acoplamiento entre los periféricos de Entrada/Salida de la tarjeta Easyweb 3 y la señalización y ejecución de mandos del interruptor, debido a que operan a niveles de voltaje diferentes. Las etapas de acoplamiento fueron diseñadas para ser compatibles con el actual sistema de monitoreo y control con que cuenta la CFE, con la finalidad de no realizar ninguna modificación a las instalaciones en el momento de poner en operación el sistema desarrollado.

CAPÍTULO 1. INTRODUCCIÓN

El presente capítulo da una descripción general del trabajo desarrollado, las características de cada uno de los elementos que conforman al sistema implementado, los equipos con que va a interactuar y las condiciones bajo las cuales va a operar.

1.1 SISTEMAS ELÉCTRICOS DE DISTRIBUCIÓN.

Los sistemas eléctricos de distribución, son el medio que permite que la energía eléctrica sea entregada a los centros de consumo, una vez que ésta ha sido generada en las centrales o plantas eléctricas [1].

Un sistema de distribución, como eslabón principal del suministro de energía eléctrica, tiene como función principal transportar la energía eléctrica de las subestaciones de potencia o en algunos casos, fuentes de generación, a los lugares de utilización; este suministro de energía eléctrica debe darse bajo parámetros de calidad bien definidos como son: tensión, frecuencia, forma de onda, secuencia de fases y continuidad. En México, los sistemas eléctricos de distribución comprenden principalmente seis partes:

- a) Líneas de sub-transmisión
- b) Subestaciones de distribución
- c) Circuitos de media tensión
- d) Transformadores de distribución
- e) Circuitos de baja tensión
- f) Acometidas

La subestación de distribución es un conjunto de equipos eléctricos interconectados para la conversión y seccionamiento de energía eléctrica recibida en bloque y distribuida en diferentes trayectorias a través de los circuitos de distribución, los cuales son circuitos eléctricos que parten de las subestaciones de distribución y proporcionan la potencia eléctrica a los transformadores de distribución.

Cada circuito de distribución cuenta con un interruptor, con el objetivo de proporcionar una protección al transformador de potencia en el momento que se presente algún tipo de falla, permanente o transitoria, y así evitar que el transformador sea sobrecargado por el nivel de cortocircuito generado por la falla, la cual puede provocar algún tipo de desgaste que reduce su periodo de vida útil o produce algún daño en el transformador, lo que implica una

interrupción del servicio de energía eléctrica a todos los circuitos de distribución dependientes de éste.

Actualmente, el interruptor es monitoreado y telecontrolado mediante un sistema implementado por Comisión Federal de Electricidad (CFE) llamado Siscopromm (Sistema de comunicación, protección y medición); mediante este sistema se pueden realizar aperturas, cierres y bloqueo/desbloqueo de recierres, para cuando se realizan mantenimientos a las líneas de distribución, a las cuchillas asociadas al interruptor (by pass o laterales) o bien al mismo interruptor.

El bloqueo del recierre de un interruptor de circuito de distribución se realiza cuando el personal de CFE realiza mantenimiento o modificación a las instalaciones o realiza alguna maniobra sobre el circuito, con su respectiva licencia y autorización por parte del operador ciudad. El bloqueo del recierre consiste en evitar que al presentarse una falla o mala maniobra, el interruptor realice la apertura del circuito pero no realice su ciclo de recierres; lo anterior con fines de seguridad para el personal de CFE que se encuentra realizando algún trabajo o maniobra sobre el circuito.

El sistema Siscopromm cuenta con unidades terminales remotas (UTR's) las cuales se encargan de establecer la comunicación entre la terminal central (operador ciudad) y las diferentes subestaciones, la comunicación se realiza mediante radiofrecuencia en la banda UHF (Ultra High Frequency); las UTR's se encargan de la codificación y decodificación de los mandos y alarmas, mediante el protocolo DNP 3, y se apoyan en radios de comunicaciones para la modulación y demodulación en UHF de los paquetes de datos.

El monitoreo de los interruptores de circuito actualmente sólo se realiza por el sistema Siscopromm; por esta razón es necesaria la implementación de otro sistema que realice las mismas funciones utilizando otro medio de comunicación, con la finalidad de garantizar y optimizar la comunicación entre la terminal central y los interruptores de circuito de distribución ubicados en las diferentes subestaciones.

1.2 PLANTEAMIENTO DEL PROBLEMA.

El monitoreo y control de un interruptor de circuito de distribución es primordial para garantizar tanto la continuidad del servicio de energía eléctrica como la seguridad del personal de la CFE; por esta razón se considera que es necesario contar con un sistema alternativo que realice las mismas funciones, pero utilizando un medio de comunicación alternativo al

empleado actualmente. En el presente trabajo se busca desarrollar este sistema utilizando como medio de comunicación la red de Intranet de CFE, para lograr la implementación se utilizará la tarjeta de desarrollo Easyweb 3, la cual brinda grandes ventajas para desarrollar un sistema de monitoreo y control vía Protocolo de Internet (IP), ya que cuenta con una interfaz para la conexión a una red de área local, un microcontrolador (MSP430) para realizar el control y periféricos para realizar el monitoreo y ejecutar los mandos deseados.

Los parámetros del interruptor que se desean monitorear son los siguientes:

- *Estado del interruptor (Abierto/Cerrado).*
- *Estado del recierre (Normal/Bloqueado).*
- *Alimentación del sistema de control (Presencia/Ausencia de Vcd).*
 1. *Bobina de cierre.*
 2. *Bobina de apertura.*
- *Alimentación del sistema eléctrico (Presencia/Ausencia de Vca).*
- *Estado del sistema mecánico o de compresión (Resorte cargado/descargado o Tanque lleno/vacío, respectivamente).*

Los parámetros del interruptor que se desean controlar son los siguientes:

- *Apertura y cierre.*
- *Bloqueo y normalización del recierre.*

Estos son los parámetros con los que cuenta un interruptor de circuito de distribución y que determinan su buen funcionamiento y sobre los cuales se centrará el desarrollo del presente trabajo.

1.3 JUSTIFICACIÓN.

Las razones primordiales para desarrollar un sistema alternativo de monitoreo y control de un interruptor de circuito de distribución es que actualmente la CFE busca mayor eficiencia en el sistema eléctrico, en la continuidad del servicio de energía eléctrica y garantizar la seguridad del personal. Por lo que es necesario contar con un sistema alternativo que sirva de respaldo para cuando se presente alguna falla o irregularidad en el sistema principal; ya que de no contar con un sistema de respaldo y se presenta alguna falla o irregularidad en el sistema principal, los mandos que se requieran realizar para un interruptor de circuito de distribución

deberán realizarse de manera local, lo cual implica que se debe trasladar personal de la CFE a las subestaciones a realizarlos y esto genera pérdida de tiempo y recursos para la CFE.

A continuación se expone un posible caso para ilustrar los efectos de una falla en el sistema principal de monitoreo y control de CFE.

- Se va a realizar alguna modificación en línea viva (13.8KV) en la red de distribución, el personal se encuentra ubicado en el punto de trabajo, para poder iniciar solicitan autorización al operador ciudad y solicitan el bloqueo del recierre del circuito sobre el que se van a realizar los trabajos, si el sistema principal falla y no es posible realizar el bloqueo del recierre vía remota en ese momento, el inicio de los trabajos se retrasa lo cual implica una pérdida de recursos para la CFE. De no recuperarse el sistema, es necesario que personal de la CFE se traslade a la subestación a realizar el bloqueo del recierre de manera local, lo cual implica que una cuadrilla de distribución se desviará de sus actividades planeadas y tendrá que permanecer en la subestación o en punto cercano esperando a que se termine la actividad para normalizar el recierre. Esto implica pérdida de recursos y mano de obra.

Se considera que este trabajo es justificable ya que actualmente la CFE busca tener una productividad y eficiencia lo más alta posible, con la finalidad de aprovechar al máximo sus recursos, tanto humanos como económicos.

1.4 OBJETIVOS.

General.

Realizar un sistema alternativo de monitoreo y control a distancia de un interruptor de circuito de distribución vía protocolo de Internet mediante la implementación de la tarjeta de desarrollo Easyweb 3. En la figura 1.1 se muestra un diagrama, el cual ilustra la operación general del sistema a desarrollar.

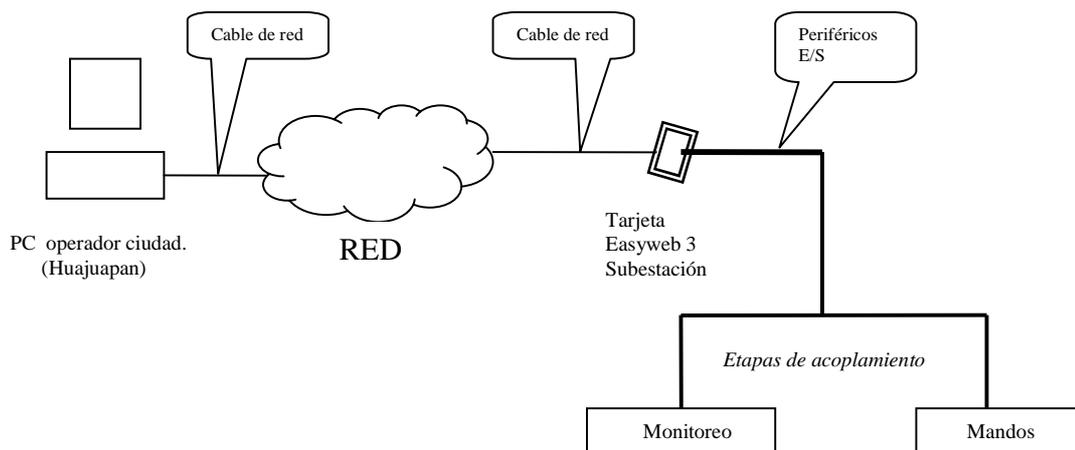


Figura 1.1 Operación general del sistema de monitoreo y control

Específicos.

1. Realizar un programa para el microcontrolador MSP430, contenido en la tarjeta Easyweb 3, que sea capaz de monitorear el estado de un interruptor de circuito, así como realizar mandos sobre éste.
2. Diseñar una página web, la cual será la interfaz de usuario del sistema y mediante ésta se realizarán los mandos y se mostrará el estado de los parámetros (Alarmas) del interruptor.
3. Diseñar e implementar las etapas de acoplamiento entre los periféricos de la tarjeta Easyweb 3 y la señalización de alarmas y ejecución de mandos de un interruptor de circuito.

1.5 METODOLOGÍA.

El sistema que se pretende desarrollar, por sus características, encaja con los llamados sistemas empotrados, los cuales son una combinación de hardware y software que interactúan para lograr una función específica [URL 2]. La combinación de hardware y software marcan una de las principales partes del diseño de cualquier sistema empotrado: decidir que funcionalidad se implementará vía software y cual vía hardware. De hecho, esta decisión impondrá la arquitectura del sistema completo, ya que indica los diferentes componentes que necesitan desarrollarse.

Como en el diseño de cualquier sistema, se debe realizar una división del problema principal en diferentes tareas de menor complejidad; abordarlas y cubrir los requerimientos del sistema. Lo cual permite realizar las correcciones necesarias en el momento adecuado y que tengan un impacto menor en el desarrollo del sistema completo.

La división del problema principal en diferentes tareas en el entorno de los sistemas empotrados es conocido como: flujo de diseño del sistema empotrado [URL 2].

Un flujo de diseño bien establecido para los sistemas empotrados es el mostrado en la figura 1.2.

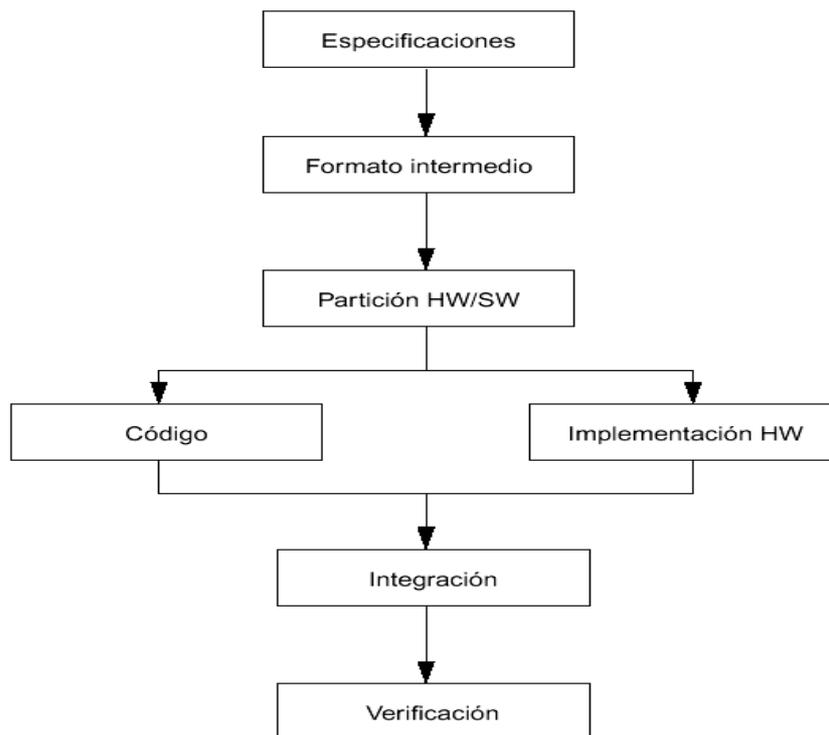


Figura 1.2 Flujo de diseño de un sistema empotrado.

Especificaciones: Consiste en realizar una descripción del sistema completo. El resultado de esta tarea servirá para fijar los diferentes algoritmos que estén involucrados en la operación del sistema (ya sean algoritmos de control o de operación con datos). La descripción deberá contar con los siguientes componentes: una descripción funcional, un conjunto de propiedades, un conjunto de índices de desarrollo y un conjunto de restricciones.

Formato intermedio: Esta tarea consiste en particionar el sistema. Una vez que haya concluido, se tendrá el comportamiento del sistema dividido en varias funciones de menor

complejidad, con el propósito de abordar el problema completo como varios problemas desconectados entre sí.

Partición hardware y software: Esta tarea sirve para determinar qué funciones se realizarán mediante dispositivos de aplicación específica (hardware) o mediante programación (software).

Integración: En esta tarea se deben integrar los elementos del sistema, de tal manera que todos los dispositivos realicen sus funciones de forma correcta, para que el flujo de datos (en el microcontrolador) este de acuerdo con la operación a realizar.

Verificación: Se realiza una descripción funcional del sistema, esta mediante la realización de pruebas, de las cuales ya se espera un resultado específico, de esta manera se verificará la funcionalidad del sistema.

Aunque se ha tratado como un flujo secuencial de tareas, esta situación no se adapta a la realidad. Realmente todas las tareas están relacionadas entre sí, y no únicamente con las adyacentes.

1.6 ORGANIZACIÓN DEL DOCUMENTO.

El presente documento está organizado en 5 capítulos y una sección de apéndices, a continuación se describe, a grandes rasgos, el contenido de cada capítulo:

CAPÍTULO 1, INTRODUCCIÓN: Describe de manera general el sistema desarrollado, la razón por la cual se consideró que es necesario implementar un sistema alternativo de monitoreo y control de un interruptor de circuito de distribución. Describe el entorno donde se pondrá en operación el sistema implementado, plantea los objetivos y la metodología mediante la cual se desarrolló el sistema.

CAPÍTULO 2, MARCO TEÓRICO: Describe los elementos necesarios para el desarrollo del sistema, los elementos del entorno donde será puesto en operación y la plataforma de desarrollo.

CAPÍTULO 3, DESARROLLO: Describe cada una de las etapas mediante las cuales se implementó el sistema de monitoreo y control de un interruptor de circuito de distribución vía IP.

CAPÍTULO 4, PRUEBAS Y RESULTADOS: Describe cada una de las pruebas realizadas en cada etapa del sistema, muestra los resultados obtenidos de la integración del sistema en una subestación real de distribución.

CAPÍTULO 5, CONCLUSIONES Y PERSPECTIVAS: Da las conclusiones obtenidas después de implementar el sistema de monitoreo y control de un interruptor de circuito de distribución vía IP, describe los beneficios que se tendrán si se cuenta con un sistema alternativo de monitoreo y control de las subestaciones de distribución de la CFE, presenta las perspectivas para mejorar el sistema desarrollado.

Una Subestación de Distribución es el punto de unión entre la red de transmisión y los alimentadores de distribución (*circuitos de distribución*). Usualmente el sistema de transmisión opera en anillo, este arreglo permite la transferencia de voltaje entre fuentes y cargas por 2 trayectorias, no así el sistema de alimentadores de media tensión que generalmente es radial, que solo permite la transferencia de voltaje entre la fuente y la carga por una sola trayectoria [1].

El equipo de una subestación consiste de barras (fase A, fase B, fase C en un bus) en el lado primario y el lado secundario, interruptores en los dos niveles de tensión, equipo de protección, medición y control. El equipo más importante son los transformadores de potencia que definen la capacidad de la subestación y que transforman la tensión al nivel de distribución, media tensión, en la figura 2.2 se muestra un transformador de potencia de una subestación.

La capacidad individual de un transformador varía desde 1 hasta 150 MVA, frecuentemente están equipados con cambiadores de tensión automáticos para mantener en el lado de distribución un rango de tensión con poca variación, independientemente de la variación en el lado de transmisión (Alta tensión). La tensión en el lado primario puede variar un 5%, pero la tensión en el lado del secundario se mantiene dentro de un $\pm 0.5\%$ de variación.



Figura 2.2 Transformador de potencia.

Frecuentemente las subestaciones de distribución tienen de uno a seis transformadores de distribución, teniendo más de un transformador se incrementa la confiabilidad porque un transformador se puede sobrecargar hasta un 12% por un período de 4 horas, durante emergencias o tiempos cortos para sustitución o reparación de un transformador.

2.1.1 INTERRUPTOR DE POTENCIA.

El interruptor es un dispositivo destinado al cierre o apertura de la continuidad de un circuito eléctrico bajo carga, en condiciones normales o en condiciones de cortocircuito. Sirve para insertar o retirar a cualquier circuito energizado, máquinas, aparatos, líneas aéreas o cables [1].

El interruptor de potencia es, junto con el transformador de potencia, uno de los dispositivos más importantes de una subestación. Su comportamiento determina el nivel de confiabilidad que se puede tener en un Sistema Eléctrico de Potencia.

El interruptor debe ser capaz de actuar con corrientes eléctricas de intensidades diferentes, pasando desde las corrientes capacitivas de varios cientos de Amperes a las inductivas de varias decenas de KA (cortocircuito).

El interruptor está formado por tres partes principales:

PARTE ACTIVA: Constituida por las cámaras de extinción que soportan los contactos fijos y el mecanismo de operación que soporta los contactos móviles. La cámara de extinción es el punto donde se realiza la interrupción de la corriente eléctrica y tiene como función primordial extinguir el arco eléctrico que se genera en el momento de realizar la separación de los contactos del interruptor.

PARTE PASIVA: Formada por una estructura que soporta uno o tres depósitos de aceite, si el interruptor es de aceite, en los que se aloja la parte activa. En sí, la parte pasiva desarrolla las funciones siguientes:

- a) Protege eléctrica y mecánicamente al interruptor.
- b) Ofrece puntos para el levantamiento y transporte del interruptor, así como espacio para la instalación de los accesorios.

ACCESORIOS: Esta parte incluye lo siguiente:

- a) Boquillas terminales, que a veces incluyen transformadores de corriente.
- b) Válvulas de llenado, descarga y muestreo del fluido aislante.
- c) Conectores de tierra.

d) Placa de datos.

e) Gabinete que contiene los dispositivos de control, protección, medición, accesorios como: mecanismo de operación, bobinas de cierre y de disparo, calefacción, etc.

En la figura 2.3 se muestran 3 interruptores de circuito de distribución de una de las subestaciones de la CFE.



Figura 2.3 Interruptores de circuitos de distribución (Media Tensión).

2.1.2 CLASIFICACIÓN DE LOS INTERRUPTORES.

Los interruptores pueden clasificarse de diferentes maneras [2], éstas se describen en la presente sección.

POR SU MEDIO DE EXTINCIÓN:

- **Interruptor en gran volumen de aceite.** Fueron los primeros interruptores que se emplearon en alta tensión y que utilizaron el aceite para la extinción del arco eléctrico generado en el momento de separar los contactos del interruptor. En este tipo de extinción, el arco eléctrico producido al separar los contactos del interruptor, calienta el aceite dando lugar a una formación de gas muy intensa, aprovechando el diseño de la cámara, el gas empuja un chorro de aceite a través del arco, provocando su alargamiento y enfriamiento hasta llegar a la extinción del mismo. Para grandes

tensiones y capacidades de ruptura, cada fase del interruptor va dentro de un tanque separado, aunque el accionamiento de las tres fases es simultáneo, por medio de un mando común. Cada fase tiene doble cámara interruptora, conectadas en serie, lo cual facilita la ruptura del arco al repartirse la caída de tensión según el número de cámaras. Para conseguir que la velocidad de los contactos sea elevada, de acuerdo con la capacidad interruptora de la cámara, se utilizan poderosos resortes, y para limitar el golpe que se produciría al final de la carrera, se utilizan amortiguadores. En este tipo de interruptores, el mando puede ser eléctrico, con resortes o con compresora unitaria según la capacidad interruptora del dispositivo.

- **Interruptor en pequeño volumen de aceite.** Tiene forma de columna, fue inventado en Suiza por el Dr. J. Landry. Por el pequeño consumo de aceite, son muy utilizados en Europa, en tensiones de hasta 230 KV y de 2500 MVA de capacidad interruptiva. En general se usan en tensiones y potencias medianas. Este interruptor utiliza aproximadamente un 5% del volumen de aceite del caso anterior. Las cámaras de extinción tienen la propiedad de que el efecto de extinción aumenta a medida que la corriente que va a interrumpir crece. Por eso, al extinguir las corrientes de baja intensidad, las sobretensiones generadas son pequeñas. La potencia de apertura es limitada sólo por la presión de los gases desarrollados por el arco, presión que debe ser soportada por la resistencia mecánica de la cámara de arqueo.
- **Interruptores de aire comprimido.** Su uso se origina ante la necesidad de eliminar el peligro de inflamación y explosión del aceite utilizado en los interruptores de los dos casos anteriores. En este tipo de interruptores el apagado del arco se efectúa por la acción violenta de un chorro de aire que barre el aire ionizado por efecto del arco. El poder de ruptura aumenta casi proporcionalmente a la presión del aire inyectado. La presión del aire comprimido varía entre 8 y 13 Kg/cm² dependiendo de la capacidad de ruptura del interruptor. La extinción del arco se efectúa en un tiempo muy corto, del orden de 3 ciclos de la onda de voltaje de 60Hz, lo cual produce sobretensiones mayores que en los casos anteriores.

- **Interruptores en hexafluoruro de azufre (SF₆).** Para la extinción del arco eléctrico se aprovechan las excelentes propiedades del gas hexafluoruro de azufre (fabricado por primera vez en París, Francia, en 1900) ha sido reconocido desde hace muchos años, pero las investigaciones para su aplicación en equipos eléctricos comenzaron en 1950 en Estados Unidos, y en 1953 apareció en el mercado el primer interruptor en hexafluoruro de azufre (SF₆).

POR SU MECANISMO DE OPERACIÓN:

El mecanismo de operación del interruptor para realizar las operaciones de cierre y apertura puede variar de acuerdo a la marca del interruptor, básicamente son las siguientes:

- **Neumático.** El sistema mecánico que realiza las operaciones es excitado mediante un pistón neumático.
- **Hidráulico.** El sistema mecánico que realiza las operaciones es excitado mediante un sistema hidráulico.
- **Resorte.** El sistema mecánico que realiza las operaciones es excitado mediante la energía liberada por un resorte comprimido, en el momento de ser liberado.

Pero el requerimiento que deben cumplir cada uno de los mecanismos es que deben disponer de energía almacenada que permita la ejecución de varias operaciones sin depender de las fuentes de alimentación.

POR NÚMERO DE MECANISMOS DE OPERACION:

- **Monopolar.** Cada polo (fase) cuenta con su mecanismo de operación y medio de extinción
- **Tripolar.** Los tres polos (fase A, fase B, fase C) dependen de un solo mecanismo de operación y medio de extinción.

2.2 SISTEMAS EMPOTRADOS.

El interés por los sistemas, que hoy día se denominan empotrados (embedded), ha sufrido un aumento considerable en los últimos años con su aplicación a los sistemas denominados de tiempo real. No obstante, existen referencias a este tipo de sistemas que datan de la época de los años 70.

En principio, no existe una definición estándar de lo que se entiende por un sistema empotrado; es más, dependiendo del área en la que se investigue, existirán respuestas diferentes. Así, es posible encontrar las siguientes definiciones [URL 2]:

1. Un sistema empotrado es un sistema operativo ejecutándose en un microcontrolador de pocos recursos.
2. Un sistema empotrado es un artefacto (hardware + software) no susceptible de modificación del algoritmo que define su comportamiento.
3. Un sistema empotrado es un procesador, con sus elementos externos que desarrolla una función específica de manera autónoma.
4. Un sistema empotrado es un sistema computador de propósito especial construido en un dispositivo mayor.
5. Un sistema empotrado es una mezcla de hardware y software que constituye un componente dentro de un sistema más complejo y se espera que funcione sin intervención humana.

Un ejemplo más detallado se muestra en la figura 2.4. En este sistema se aprecia la Unidad Central de Procesamiento (CPU) y la memoria, junto con una amplia variedad de interfaces que permite al sistema medir, manipular e interactuar con el exterior. Se puede ver que existen dos tipos de comunicación: una comunicación en el interior del sistema (que se realiza a través de un bus de sistema), y una comunicación externa (del sistema con el exterior).

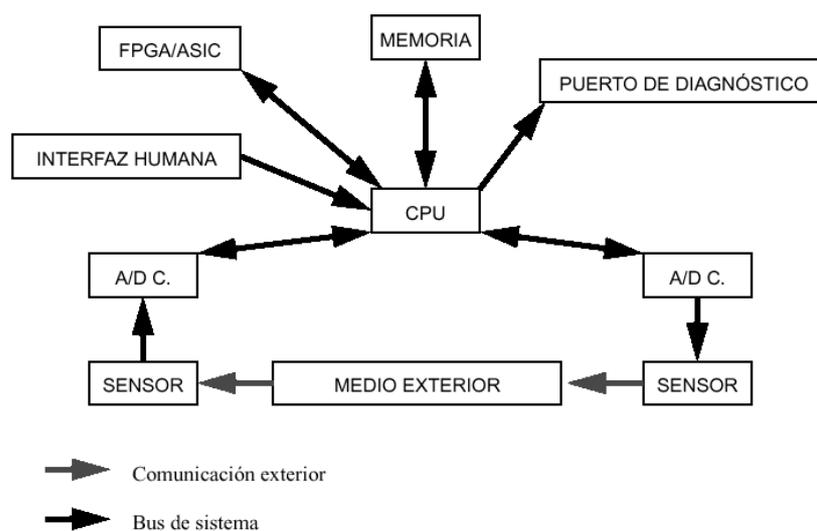


Figura 2.4 Arquitectura típica de un sistema empotrado.

Las principales características de un sistema empotrado son las siguientes [5]:

- Estos sistemas emplearán una combinación de recursos hardware y software para realizar una función específica.
- Estos sistemas realizan una única función o un conjunto muy limitado de funciones (no suelen ser de propósito general).
- El diseño de procesadores de aplicación específica suelen ser un componente significativo de estos sistemas.

Aunque todas las características anteriores son inherentes a cualquier sistema empotrado, para que el sistema sea útil, también deberá tener las siguientes [URL 4]:

- *Concurrencia.* Los componentes del sistema funcionan simultáneamente, por lo que el sistema deberá operar a la vez.
- *Fiabilidad y seguridad.* El sistema debe ser fiable y seguro frente a errores, ya que puede requerir un comportamiento autónomo. El manejo de estos errores puede ser vía hardware o software; aunque la utilización de software proporcionará un sistema menos robusto.
- *Interacción con dispositivos físicos.* Los sistemas empotrados interaccionan con el entorno a través de dispositivos E/S (entrada/Salida) no usuales, por lo que suele ser necesario un acondicionamiento de las diferentes señales.
- *Robustez.* Al sistema empotrado se le impondrá la necesidad de la máxima robustez, ya que las condiciones de uso no tienen porqué ser “buenas”, sino que pueden estar en diferentes condiciones de operación.
- *Bajo consumo.* El hecho de poder utilizar al sistema en ambientes hostiles puede implicar la necesidad de operaciones sin cables. Por lo tanto, un menor consumo implica una mayor autonomía de operación.
- *Precio reducido.* Esta característica es muy útil cuando se busca una puesta en el mercado. Esta situación no es nada inusual en el campo de los sistemas empotrados, ya que tienen una gran cantidad de aplicaciones comerciales, tanto industriales como de consumo.
- *Pequeñas dimensiones.* Las dimensiones de un sistema empotrado no dependen sólo de sí mismo, sino también del espacio disponible en el cual dicho sistema va a ser ubicado.

Dentro de las definiciones expuestas anteriormente, una gran cantidad de dispositivos pueden ser considerados como sistemas empotrados. Se encuentran diferentes clasificaciones de este tipo de sistemas. Una de las principales se puede encontrar al considerar su interacción con el resto del entorno. Atendiendo a esta característica, se tienen los siguientes tipos de sistemas [URL 2]:

- *Sistemas reactivos*. Son aquellos sistemas que dan respuesta a eventos del exterior, de tal forma que su operación está en función de su entorno.
- *Sistemas interactivos*. Son aquellos sistemas que siempre interactúan con el exterior, de tal forma que la velocidad de operación del sistema la establece el propio sistema empotrado.
- *Sistemas transformacionales*. Son aquellos sistemas que no interactúan con el exterior, únicamente toma un bloque de datos de entrada y lo transforma en un bloque de datos de salida, que no es necesario en el entorno.

2.3 TARJETA EASYWEB 3.

La tarjeta de desarrollo Easyweb 3 (figura 2.5) es una plataforma de desarrollo fabricada por Olimex [URL 3], esta tarjeta es capaz de proporcionar una interfaz para redes de internet, con la finalidad de controlar o monitorear dispositivos en forma remota desde cualquier lugar con acceso a una red de Internet. Consta de un microcontrolador MSP430F149, con un cristal de 8MHz, un controlador de red CS8900A de Cirrus Logic, una memoria flash serial de 64 KB, 2 conectores para el manejo de los puertos del microcontrolador y un conector para el programador JTAG del microcontrolador.

El MSP430F149 de Texas Instruments es un microcontrolador de 16 bits con un conjunto reducido de instrucciones (RISC), diseñado especialmente para aplicaciones que requieran un muy bajo consumo de energía, tales como dispositivos portátiles [URL 1]. Por el hecho de trabajar con palabras de 16 bits es especialmente indicado para ser programado en lenguajes de alto nivel como C para sistemas empotrados, generando un código compacto y de rápida ejecución.



Figura 2.5 Tarjeta de desarrollo Easyweb 3

A continuación se enuncian las características principales del microcontrolador MSP430 [URL 1].

- CPU RISC de 16-bits de Arquitectura Von Neuwman
- ALU (Unidad Aritmética y Lógica) de 16 bits.
- Buses de datos y direccionamiento de 16 bits
- 16 Registros de 16 bits.
- Operaciones en 1 ciclo de reloj.
- Solo 27 instrucciones básicas a aprender.(+24 emuladas)
- Múltiples y flexibles tipos de Reloj.
- Consumo nominal: 0.1 ~ 250 μ A,
- Alimentación: 1.8 ~ 3.8v (*operación*).
- Alimentación: 2.7 ~ 3.8v (*programación*).
- Frecuencia: 8MHz.
- Temperatura: -40 a 85 °C
- Diseño optimizado para programación en C

2.3.1 REQUERIMIENTOS MÍNIMOS.

Los requerimientos de Hardware y Software para utilizar la tarjeta Easyweb 3 se mencionan a continuación.

Estación de trabajo para desarrollo: Se requiere de una computadora con las siguientes características mínimas; procesador Pentium 3 a 1.7 GHz, 512 MB de memoria RAM, puerto paralelo y tarjeta de red. La computadora se utiliza para desarrollar los programas en C, descargarlos a la tarjeta mediante el puerto paralelo y para tener acceso a la página de internet contenida en la tarjeta.

Cables y alimentación de la Easyweb 3:

Se tienen los siguientes requerimientos:

- Cable de programación vía JTAG (Figura 2.6).
- Cable de red UTP con conectores RJ45.
- Fuente de alimentación AC/DC con salida de 6 a 9 Volts.



Figura 2.6 Programador JTAG para la descarga de programas

Software requerido:

- CrossWorks for MSP430. Es un software para el desarrollo de programas en C, para el microcontrolador MSP430, se requiere comprar una licencia para poder utilizarlo.
- IAR Systems. Es un software que permite desarrollar programas en C para el microcontrolador MSP430, así como proyectos para la tarjeta Easyweb 3, el software está disponible en el sitio <http://www.iar.com>, y cuenta con una licencia gratuita de 30 días.

CAPÍTULO 3. DESARROLLO

En este capítulo se describe la forma en que fue desarrollado el sistema de monitoreo y control de un interruptor de circuito de distribución vía IP, éste es un sistema empotrado porque combina hardware y software para lograr una función específica, por tanto, su desarrollo se realizó con base en el diagrama de flujo para desarrollo de sistemas empotrados mostrado en la figura 1.2. Cada una de las etapas de dicho diagrama de flujo se describe en el presente capítulo, con la finalidad de mostrar la secuencia de trabajo para el desarrollo del sistema.

3.1 ESPECIFICACIONES.

Se requiere de un sistema de monitoreo y control de un interruptor de circuito de distribución, vía protocolo de Internet (IP), mediante el uso de la tarjeta de desarrollo Easyweb 3, la cual cuenta con una interfaz para conectarse a redes de intranet (cs8900), un microcontrolador (MSP430) y periféricos de entrada y salida. Aprovechando las características de la tarjeta de desarrollo, ésta realizará el monitoreo y control del interruptor de circuito de distribución y mostrará una interfaz de usuario mediante una página web. Los parámetros que se desean monitorear son los siguientes:

- Estado del interruptor (Abierto/Cerrado).
- Estado del recierre (Normal/Bloqueado).
- Alimentación del sistema de control (Presencia/Ausencia de Vcd).
 1. Bobina de cierre.
 2. Bobina de apertura.
- Alimentación del sistema eléctrico (Presencia/Ausencia de Vca).
- Estado del sistema mecánico o de compresión (Resorte cargado/descargado o Tanque lleno/vacío, respectivamente).

Los parámetros del interruptor que se desean controlar son los siguientes:

- Apertura y cierre del interruptor.
- Bloqueo y normalización del recierre.

3.2 FORMATO INTERMEDIO.

El monitoreo y control será realizado por el microcontrolador con el que cuenta la tarjeta de desarrollo, la programación del microcontrolador será mediante el lenguaje de programación C en los compiladores IAR Embedded Workbench o CrossStudio.

La interfaz para el uso de este sistema será una página de web, la cual estará contenida en el microcontrolador, lo que hace una interfaz de usuario sea sencilla. Para tener acceso al sistema, solo se deberá teclear la dirección IP asignada a la tarjeta, en cualquier explorador de internet, como respuesta se desplegará una página web que mostrará el estado de las alarmas y permitirá realizar mandos al interruptor. Vía software se asigna una dirección IP

Para la conexión a la red de intranet de CFE, se utilizará la interfaz de red con la que cuenta la tarjeta de desarrollo.

Para realizar tanto el monitoreo como el control se deben diseñar etapas de acoplamiento entre los periféricos de entrada/salida de la tarjeta de desarrollo y la señalización y ejecución de mandos del interruptor, debido a que operan a niveles de voltaje diferentes, la tarjeta de desarrollo trabaja a 3.3Vcd y el interruptor a 125Vcd.

En la figura 3.1 se muestra un diagrama a bloques en el cual se ilustra la operación funcional del sistema que se pretende desarrollar.

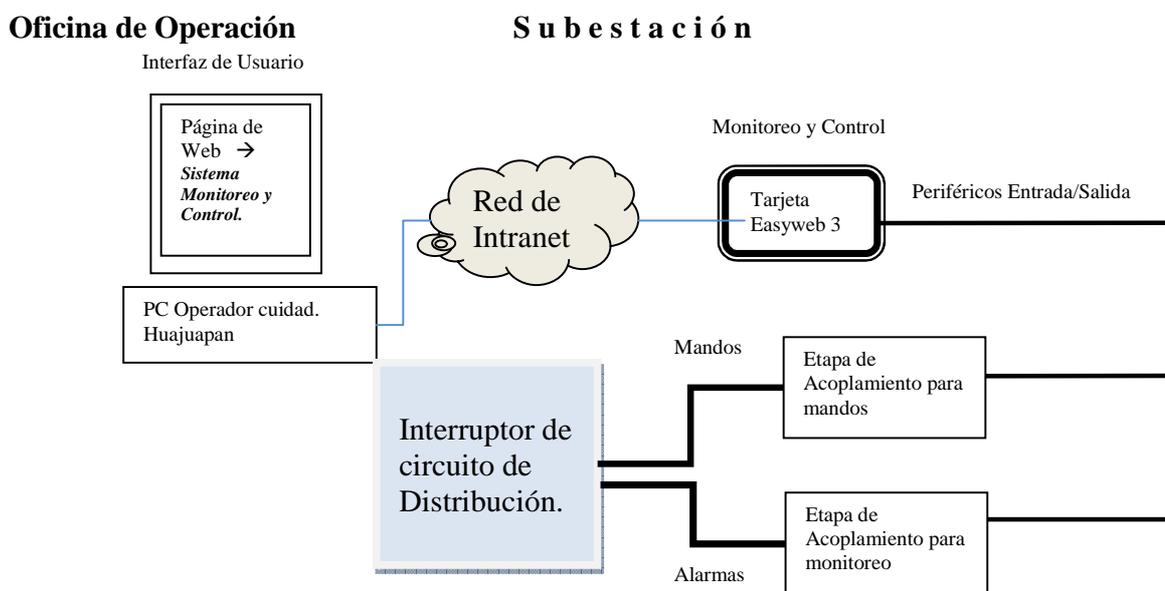


Figura 3.1 Diagrama a bloques de la operación funcional del sistema.

En la figura 3.1, se observa la división del sistema en tres funciones principales:

- Interfaz de usuario.
- Monitoreo y control.
- Etapas de acoplamiento de monitoreo y control.

Las cuales son consideradas como tareas, con el propósito de abordar el problema como 3 tareas independientes de menor complejidad.

3.3 PARTICIÓN HARDWARE/SOFTWARE.

Ahora que ya se tienen definidas cada una de las tareas, es momento de definir cual se implementará vía software y cual vía hardware. En la tabla 3.1 se define con que medio se implementará cada una de las tareas.

TAREA	SOFTWARE	HARDWARE
Interfaz de usuario	√	
Monitoreo y control	√	
Etapas de acoplamiento		√

Tabla 3.1 Tareas y medios de implementación.

3.4 DESARROLLO DE SOFTWARE.

3.4.1 INTERFAZ DE USUARIO.

La interfaz de usuario consiste en una página web, la cual permite al usuario observar el estado de todos los parámetros y realizar mandos al interruptor, con solo seleccionar el mando deseado y dar un click en un botón. La página fue desarrollada mediante el lenguaje HTML y guardada en la memoria del microcontrolador. La página que se implementó es muy sencilla, debido a que sólo requiere funciones para la visualización de información (Monitoreo) y envío de datos (Mandos).

La página contiene un área en la cual se muestra el estado de los parámetros que se monitorean del interruptor. Los parámetros que se van a monitorear y sus posibles estados son representados mediante indicadores en colores verde o rojo, como se muestra a continuación.

- Estado del interruptor:
 - Verde = Interruptor abierto.
 - Rojo = Interruptor cerrado.

- Estado del recierre:
 - Verde = Recierre normal.
 - Rojo = Recierre bloqueado.
- Alimentación del sistema de control:
 1. Bobina de Cierre:
 - Verde = Presencia de potencial (125Vcd).
 - Rojo = Ausencia de potencial (0 Vcd).
 2. Bobina de Apertura:
 - Verde = Presencia de potencial (125Vcd).
 - Rojo = Ausencia de potencial (0 Vcd).
- Alimentación del sistema de eléctrico:
 - Verde = Presencia de potencial (120Vca).
 - Rojo = Ausencia de potencial (0 Vca).
- Estado del sistema mecánico o de compresión:
 - Verde = Resorte cargado/ Tanque lleno Respectivamente.
 - Rojo = Resorte descargado/ Tanque vacío Respectivamente.

También se dispuso de un área para la ejecución de mandos, la cual contiene dos botones de selección múltiple, uno para realizar mandos sobre el *estado* del interruptor (Abrir/Cerrar) y otro para realizar mandos sobre el estado del *recierre* (Normal/Bloqueado). La ejecución de mandos consiste en enviar información a la tarjeta de desarrollo, la cual se encarga de recibir los datos, interpretarlos y ejecutar el mando deseado. El código HTML de la página de interfaz de usuario se muestra en el apéndice A.

En la figura 3.2 se muestra la disposición de los indicadores de estado de los parámetros del interruptor y los botones para la ejecución de mandos.

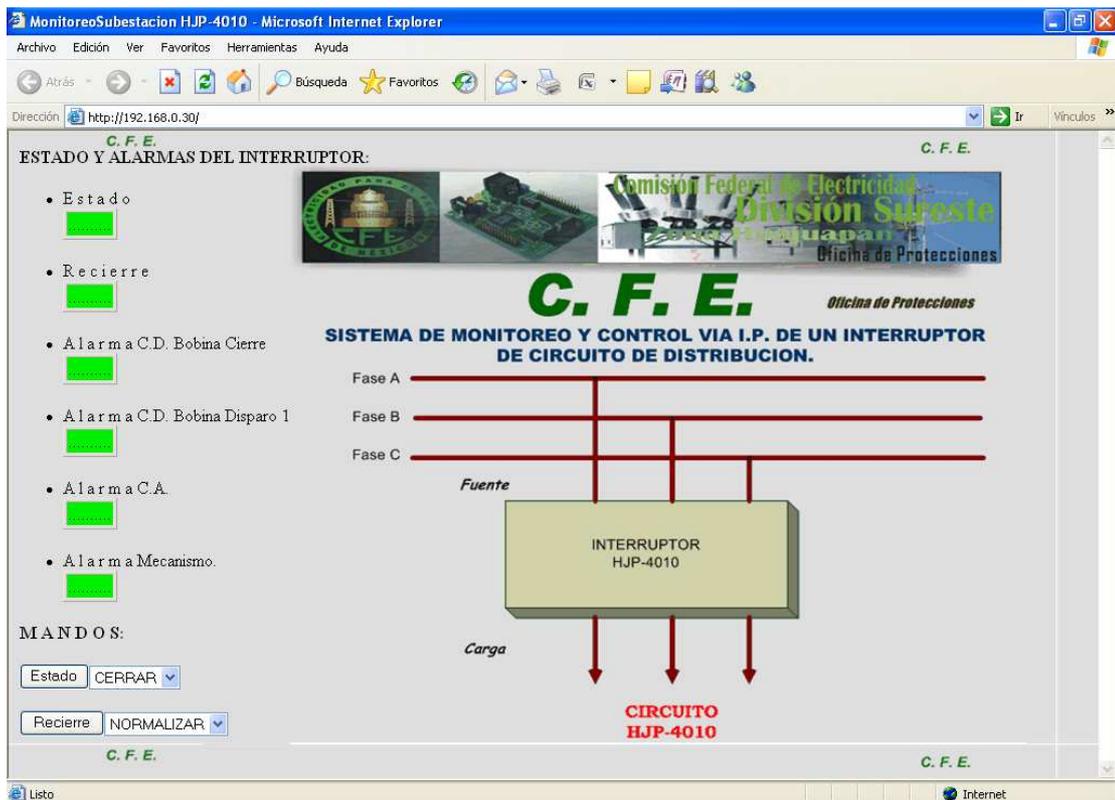


Figura 3.2 Página de interfaz de usuario.

3.4.2 MONITOREO Y CONTROL.

El monitoreo y control es realizado por el microcontrolador MSP430 que está contenido en la tarjeta de desarrollo Easyweb 3, la tarjeta cuenta con la interfaz para la comunicación entre el microcontrolador y el hardware de red (controlador de red CS8900A), el programa para el monitoreo y control fue realizado en el ambiente integrado de desarrollo *IAR Embedded Workbench*, el cual se encuentra disponible de manera gratuita en la dirección: <http://www.iar.com>.

La tarjeta Easyweb 3 incluye un software de prueba que muestra una página con la cadena “hola mundo”, en una dirección IP 192.168.0.30. Para el desarrollo del código requerido, se tomó como base este software, debido a que cuenta con las funciones necesarias para realizar de manera eficiente el envío y recepción de paquetes. Partiendo de este software, para realizar nuestro sistema se implementaron las funciones para la identificación e interpretación de los mandos, la lectura e interpretación del estado de los parámetros del interruptor y la actualización en tiempo real de los datos en la página de interfaz de usuario.

En el apéndice B se muestran los pasos necesarios para crear y modificar proyectos con el entorno de desarrollo IAR Embedded Workbench.

3.4.2.1 Software de prueba de la tarjeta.

El software de prueba con que cuenta la tarjeta Easyweb 3 consiste en un servidor web; este servidor contiene una página escrita en lenguaje HTML, la cual está almacenada en la memoria flash del microcontrolador. El software está estructurado en los tres módulos que se describen en la tabla 3.2.

MÓDULO	DESCRIPCIÓN
Aplicación. (Easyweb.c)	Transfiere datos vía Ethernet y TCP. Es la aplicación del desarrollador. Usa las funciones del módulo de TCP/IP, las cuales organizan y empaquetan los datos.
TCP/IP (tcpip.c y tcpip.h)	Librería para el desarrollo de la aplicación. Implementación de los protocolos ARP, ICMP, IP y TCP. Responde a los eventos de recepción de datos y solicitudes de conexión de un cliente.
Ethernet (cs8900.c y cs8900.h)	Controlador del hardware para usar el controlador de red cs8900. Proporciona funciones para la configuración del controlador de red cs8900, así como para la lectura y escritura de registros y envío/recepción de paquetes Ethernet.

Tabla 3.2 Estructura del software.

3.4.2.1.1 Módulo de aplicación.

En este módulo se programa la aplicación deseada, en el software de prueba la aplicación es un servidor http, el algoritmo para esta aplicación se muestra en la figura 3.3. El programa está en espera una solicitud de conexión, cuando ésta llega transfiere la página web al cliente, cierra la conexión y queda en espera de una nueva solicitud.

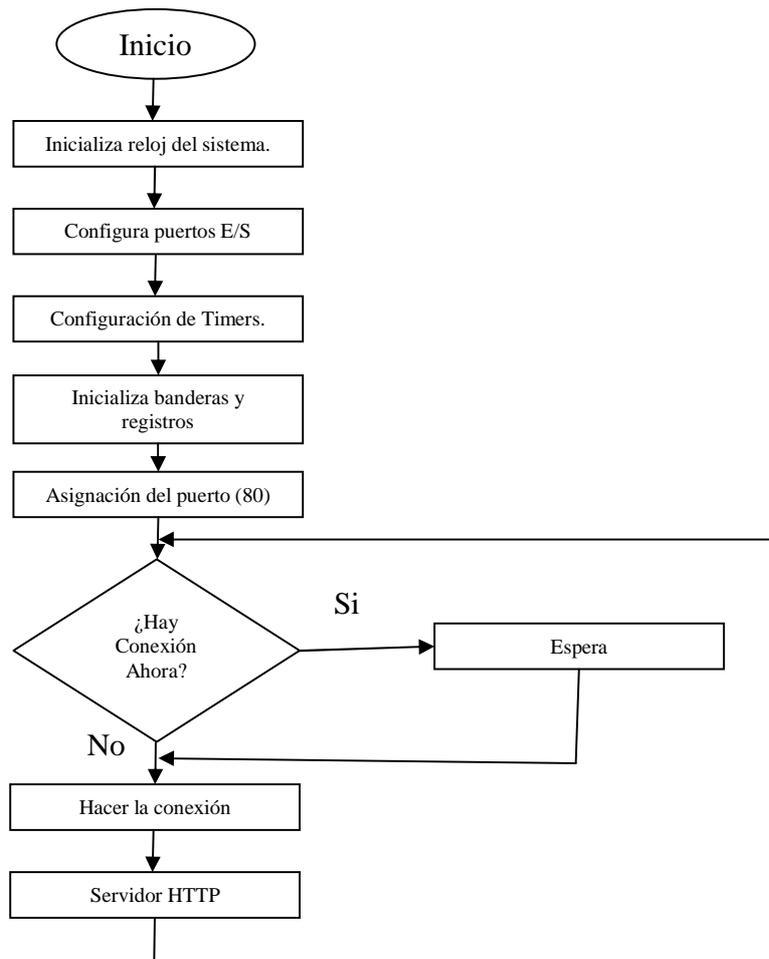


Figura 3.3 Algoritmo del módulo de aplicación del software de prueba de la tarjeta Easyweb 3.

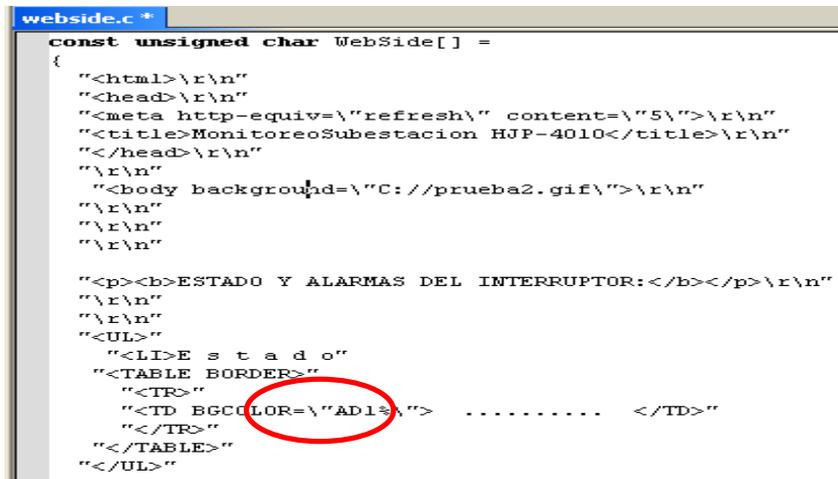
En el proyecto, el archivo que contiene el algoritmo de la figura 3.3 es el *Easyweb.c*, por lo tanto, se debe modificar este algoritmo para realizar el monitoreo y control del interruptor. En el proceso *servidor http*, se deben adherir las funciones que realizarán la lectura del estado de los parámetros del interruptor, el manejo de los valores dinámicos, que son cadenas especiales en el código HTML de la página de interfaz de usuario, los cuales van a ser sustituidos por una cadena que indica el color del estado de cada parámetro. También se debe insertar el código HTML de la página web que servirá como interfaz de usuario.

Código HTML.

El archivo *webside.c* contiene el código HTML de la página de interfaz de usuario, en un arreglo de constantes sin signo denominado *webside*. Por la sintaxis del lenguaje C, cada

renglón está entre comillas dobles ‘ ” ’. Cuando en el lenguaje HTML se utilicen las comillas dobles, éstas deberán estar precedidas del carácter diagonal invertida ‘\’, de esta forma se introducen comillas en una cadena constante. En la figura 3.4 se muestra el código HTML contenido en el arreglo de constantes sin signo, en el archivo *webside.c*.

La página mostrará el estado de los parámetros del interruptor en tiempo real, con color rojo o verde, como se explicó en el apartado de interfaz de usuario, por esta razón, el valor que indica el color para estas etiquetas en el código HTML no es asignado de manera directa. Se colocan cadenas especiales conocidas como valores dinámicos, las cuales serán identificadas en el momento de enviar la página al usuario, y se sustituirán por el valor correspondiente al estado de los parámetros. En la Figura 3.4 se resalta uno de los valores dinámicos (AD1%).



```
webside.c *
const unsigned char WebSide[] =
{
  "<html>\r\n"
  "<head>\r\n"
  "<meta http-equiv=\"refresh\" content=\"5\">\r\n"
  "<title>MonitoreoSubestacion HJP-4010</title>\r\n"
  "</head>\r\n"
  "\r\n"
  "<body background=\"C://prueba2.gif\">\r\n"
  "\r\n"
  "\r\n"
  "\r\n"
  "\r\n"
  "\r\n"
  "<p><b>ESTADO Y ALARMAS DEL INTERRUPTOR:</b></p>\r\n"
  "\r\n"
  "\r\n"
  "\r\n"
  "<UL>"
  "  <LI>E s t a d o"
  "  <TABLE BORDER>"
  "    <TR>"
  "      <TD BGCOLOR=\"AD1%" > ..... </TD>"
  "    </TR>"
  "  </TABLE>"
  "</UL>"
}
```

Figura 3.4 Identificación de valores dinámicos en código HTML.

Valores dinámicos

Con los valores dinámicos se asignan los indicadores de color a los parámetros que se están monitoreando. Se destina un puerto para la lectura del estado de los parámetros del interruptor y con una función se asigna su valor en función de la lectura que se realiza en el puerto. Con ello, se determinan los valores reales (color rojo/verde) para la señalización en la página de interfaz de usuario. Se diseñó el hardware para entregar el estado de estos parámetros en forma de un bus de datos de 6 bits, considerando el 1 lógico como el indicador *rojo* y el 0 lógico como el indicador *verde* en la señalización en la página de interfaz de usuario.

La declaración y asignación de las variables a utilizar, para determinar el estado de cada parámetro, se realiza en la cabecera del archivo Easyweb.c, las variables y su interpretación, en función de su valor, es la siguiente:

- Aestado = Estado del interruptor.
Aestado = 0 = Verde = Interruptor abierto.
Aestado = 1 = Rojo = Interruptor cerrado.
- AREcierre = Estado del recierre.
AREcierre = 0 = Verde = Recierre normal.
AREcierre = 1 = Rojo = Recierre bloqueado.
- Acd1 = Bobina de Cierre.
Acd1 = 0 = Verde = Presencia de potencial (125Vcd).
Acd1 = 1 = Rojo = Ausencia de potencial (0 Vcd).
- Acd2 = Bobina de Apertura.
Acd2 = 0 = Verde = Presencia de potencial (125Vcd).
Acd2 = 1 = Rojo = Ausencia de potencial (0 Vcd).
- Aca = Alimentación del sistema de eléctrico.
Aca = 0 = Verde = Presencia de potencial (120Vca).
Aca = 1 = Rojo = Ausencia de potencial (0 Vca).
- Amecanismo = Estado del sistema mecánico o de compresión.
Amecanismo = 0 = Verde = Resorte cargado/ Tanque lleno.
Amecanismo = 1 = Rojo = Resorte descargado/ Tanque vacío.

La asignación del puerto para la lectura de los parámetros del interruptor se realiza en el proceso *Configurar puertos E/S*, del algoritmo de la figura 3.3. Los puertos disponibles en la tarjeta Easyweb 3 son el puerto 1, puerto 2 y puerto 4 que están en el conector EXT 1. El puerto 1 se utiliza para realizar los mandos (puerto de salida) y el puerto 2 para el monitoreo (puerto de entrada) del estado de los parámetros. La asignación de estos puertos se muestra en la Figura 3.5.

```

easyweb.c *
//-----
static void InitPorts(void)
{
    P1DIR = 0xff; //Pto. 1 salidas para mandos
    P1OUT = 0x00; // 0= ningun mando activado

    P2DIR = 0x00; //Pto. 2 entradas para monitoreo de alarmas

    P4DIR = 0xff; //Pto. 4 libre, asigna como salida con valor 0
    P4OUT = 0x00;
}

```

Figura 3.5 Asignación de puertos para lectura de parámetros y ejecución de mandos.

Para realizar la lectura del puerto y asignar el valor correspondiente a cada variable, es necesario definir una relación entre cada variable y el bit del bus de lectura. Esta relación se muestra en la tabla 3.3.

BIT	VARIABLE RELACIONADA	PARÁMETRO DEL INTERRUPTOR
0	Aestado	Estado del interruptor
1	ARecierre	Estado del recierre
2	Acd1	Bobina de Cierre
3	Aca	Alimentación del sistema de Eléctrico
4	Amecanismo	Estado del sistema mecánico o de compresión
5	Acd2	Bobina de Apertura
6	----	----
7	----	----

Tabla 3.3 Parámetros a monitorear y su variable relacionada.

La función para la asignación de valores a cada variable se implementa con base en el algoritmo de la figura 3.6. Para implementar esta función fue necesario realizar todas las comparaciones, ya que cuando se realizó mediante máscaras, para seleccionar e identificar la variable deseada, el compilador si aceptó las operaciones lógicas. En el momento de descargar el programa en la tarjeta, se observó que la tarjeta no realizaba esta operación y no se obtenía el estado real de cada parámetro, después se depuró el programa y se verificó que la tarjeta no realizaba esta operación lógica, la función abortaba y no afectaba las variables que determinan el estado de cada parámetro del interruptor.

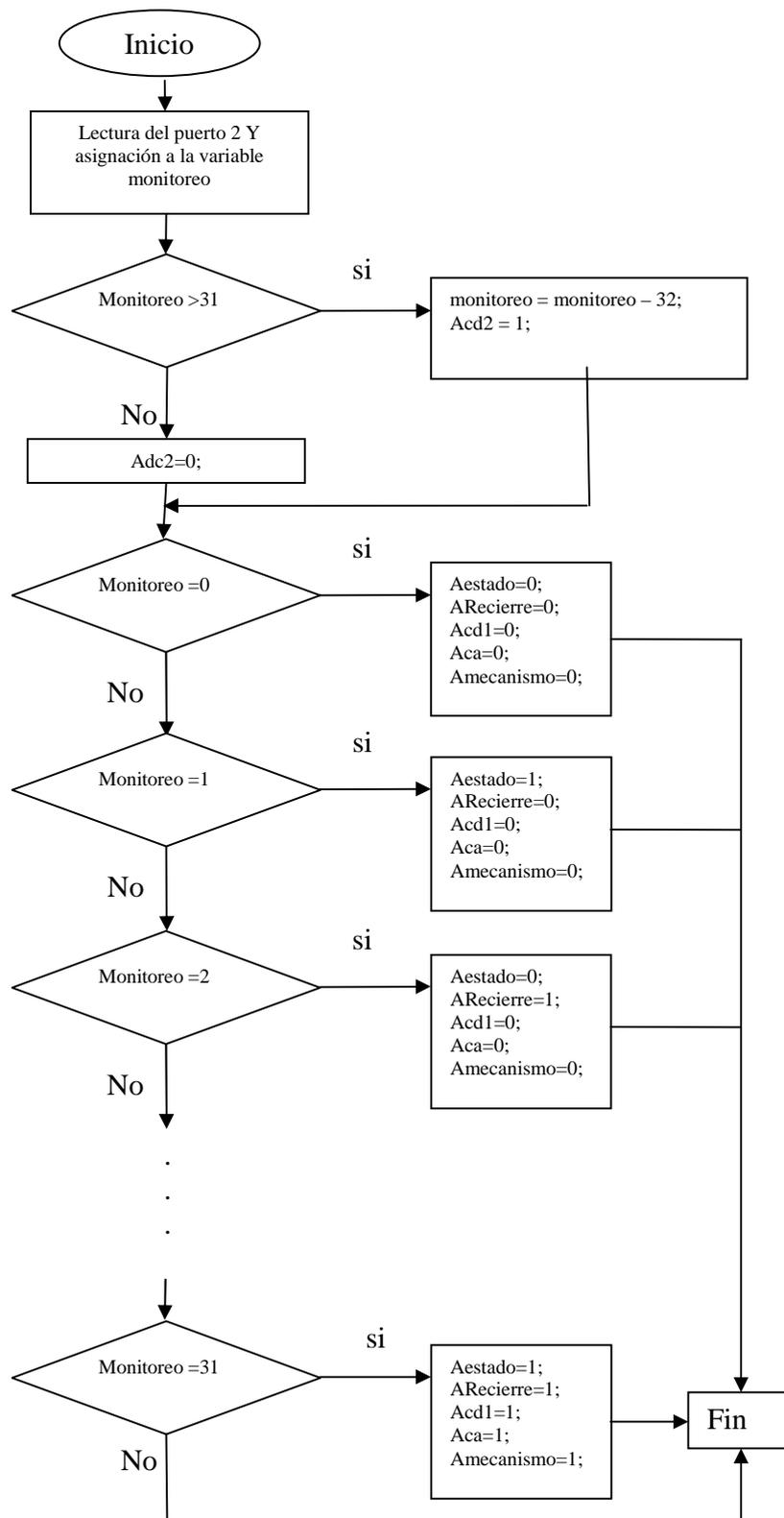


Figura 3.6 Algoritmo para la asignación de valores a cada variable, en función del estado de los parámetros del interruptor.

La función para la asignación de los valores correspondientes a cada variable va a ser llamada antes de realizar la inserción de los valores dinámicos de la página de interfaz de usuario, con la finalidad de que cada vez que se envíe la página, refleje el estado actual de los parámetros del interruptor. El código en lenguaje C de la función del algoritmo de la figura 3.6 se muestra en el apéndice C.

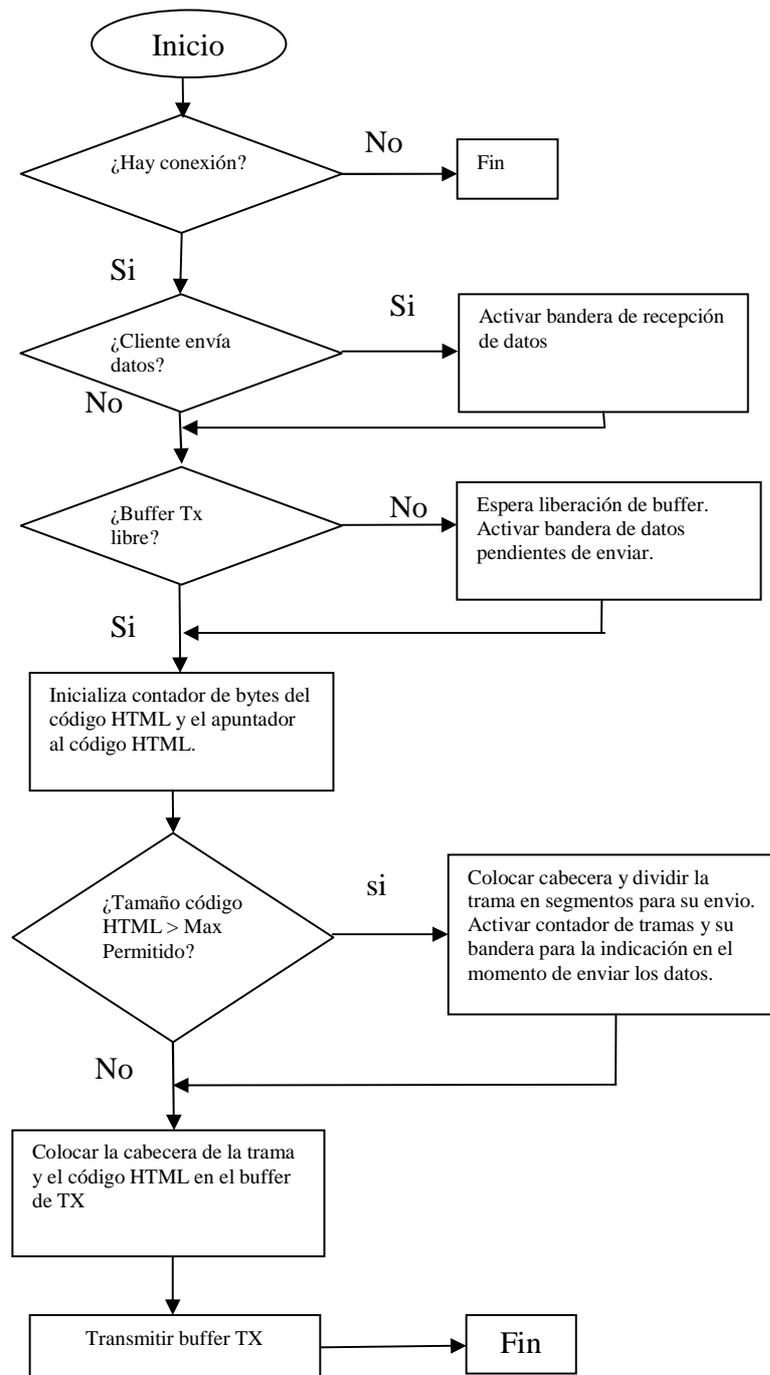


Figura 3.7 Algoritmo para el proceso *Servidor HTTP*.

Para determinar el punto donde conviene realizar la inserción de los valores dinámicos, en la página de interfaz de usuario, es necesario desglosar el algoritmo para el proceso *Servidor HTTP* del algoritmo de la figura 3.3. El algoritmo para este proceso se muestra en la figura 3.7.

Analizando el algoritmo de la figura 3.7, se observa que el punto indicado para realizar la actualización de las variables que monitorean el estado de los parámetros del interruptor y la inserción de los valores dinámicos de la página de interfaz de usuario es antes del proceso *Transmitir buffer Tx*.

La función que inserta los valores dinámicos al código HTML de la página, se implementó basándose en el algoritmo de la figura 3.8, su código en lenguaje C también es parte del apéndice C.

Después de ejecutar esta función, el buffer de transmisión cuenta con los valores correctos que indican el estado real de los parámetros del interruptor y al transmitir los datos al usuario, la página de interfaz los reflejara mediante los colores rojo y verde respectivamente, de manera correcta.

3.4.2.1.2 Módulo TCP/IP.

Este módulo del software representa la parte más importante de la tarjeta Easyweb 3, porque con este se llevan a cabo los protocolos para transferir datos sobre una conexión TCP/IP. El archivo en el proyecto que contiene este módulo es el *tcpip.c*. Usa las funciones del módulo Ethernet para enviar y recibir datos de una manera simple, permite que el módulo de aplicación sea usado por el desarrollador de una manera sencilla.

El módulo TCP/IP es básicamente una colección de procedimientos para el manejo de eventos. Esto se realiza mediante el monitoreo de un registro de banderas. El módulo TCP/IP responde a los siguientes eventos:

- Una trama es recibida.
- La aplicación activa un evento (Abre una conexión, envía datos, etc.).
- El tiempo de espera de respuesta se excede.
- Ocurre un error (error de la red, la conexión es perdida, etc).

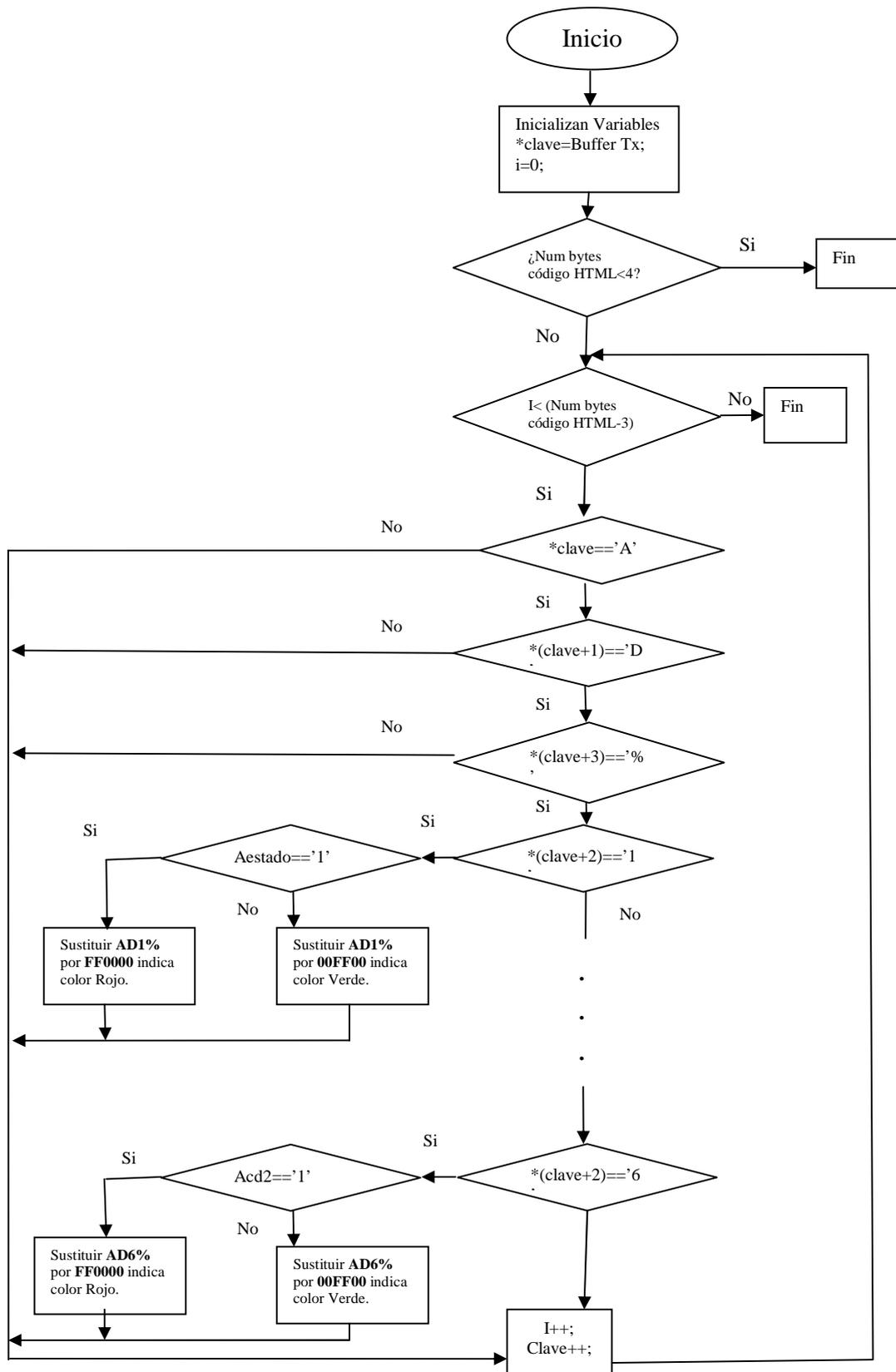


Figura 3.8 Algoritmo para la función insertar valores dinámicos.

Una de las funciones primordiales del módulo es el manejo de tramas recibidas para determinar su tipo, con la finalidad de activar las banderas necesarias para su procesamiento correcto. El algoritmo para este módulo es el mostrado en la figura 3.9.

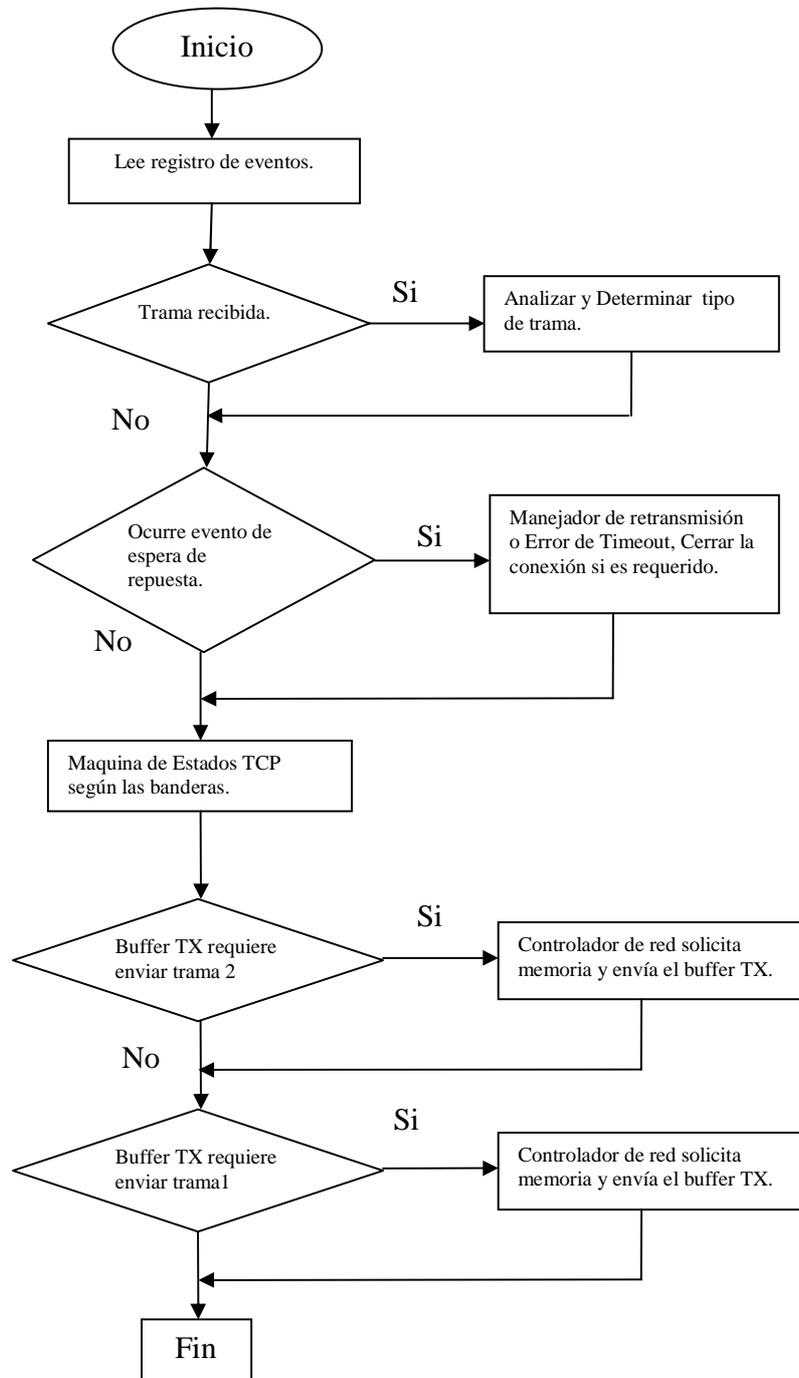


Figura 3.9 Algoritmo para el módulo TCP/IP

En este módulo se implementará una función para identificar y ejecutar los mandos que se deseen realizar al interruptor, esto se realiza haciendo un análisis en el buffer de recepción (buffer RX), con la finalidad de identificar la secuencia de caracteres que se envían para cada mando y ejecutarlo.

Cada mando es enviado desde la página de interfaz de usuario, mediante un procedimiento POST [URL 5], propio del lenguaje HTML, el cual consiste en un formulario que contiene un botón (*Estado/Recierre*) y una lista desplegable (Abrir-Cerrar/Normalizar-Bloquear), el mando seleccionado es enviado a la dirección <http://192.168.0.30>. A continuación se muestra el código para realizar los mandos de apertura y cierre del interruptor:

```
<FORM METHOD="POST" ACTION="http://192.168.0.30">
<INPUT TYPE="SUBMIT" VALUE="Estado">
<SELECT NAME="glista1g">
<OPTION>CERRAR
<OPTION>ABRIR
</SELECT> </FORM>
```

Para realizar el mando de apertura sobre el interruptor, se debe seleccionar de la lista desplegable la opción ABRIR y dar un click sobre el botón *Estado*, entonces se enviarán a la dirección <http://192.168.0.30> los siguientes datos: glista1g=ABRIR, por lo tanto, lo que se debe hacer en la función que identifique y ejecute los mandos, es buscar esta secuencia de caracteres para determinar y ejecutar cada mando.

Para observar la secuencia de caracteres que se envían en la ejecución de los procedimientos POST de la página de interfaz de usuario, se utilizó el software Ethereal, el cual es un analizador de tramas de Ethernet. Mediante esta herramienta se pudo observar la secuencia de caracteres que se envían para cada mando, esto facilitó la implementación de la función que determina y ejecuta los mandos. En las tablas 3.4, 3.5, 3.6 y 3.7 se muestran los segmentos de la trama capturadas por el software Ethereal, correspondiente a los datos enviados para cada mando, con la finalidad de observar la secuencia de valores en hexadecimal que son enviados y así determinar que secuencias serán buscadas en el buffer RX.

Mando ABRIR														
HEX	67	6c	69	73	74	61	31	67	3d	41	42	52	49	52
ASC	g	l	i	s	t	a	l	g	=	A	B	R	I	R

Tabla 3.4 Trama enviada para el mando *Abrir*.

Mando CERRAR															
HEX	67	6c	69	73	74	61	31	67	3d	43	45	52	52	41	52
ASC	g	l	i	s	t	a	l	g	=	C	E	R	R	A	R

Tabla 3.5 Trama enviada para el mando *Cerrar*.

Mando NORMALIZAR																			
HEX	65	6c	69	73	74	61	31	67	3d	4e	4f	52	4d	41	4c	49	5a	41	52
ASC	e	l	i	s	t	a	l	g	=	N	O	R	M	A	L	I	Z	A	R

Tabla 3.6 Trama enviada para el mando *Normalizar*.

Mando BLOQUEAR																			
HEX	65	6c	69	73	74	61	31	67	3d	42	4C	4F	51	55	45	41	52		
ASC	e	l	i	s	t	a	l	g	=	B	L	O	Q	U	E	A	R		

Tabla 3.7 Trama enviada para el mando *Bloquear*.

Una vez identificadas las tramas, se diseñó el algoritmo para detectarlas y ejecutar el mando que corresponda, el algoritmo para esta función se muestra en la figura 3.10.

Antes de implementar la función en C, fue necesario definir el puerto para la ejecución de los mandos, así como la correspondencia entre cada mando y el bit del puerto. El puerto asignado para realizar los mandos es el puerto 1, como se observa en la figura 3.5. La correspondencia entre cada mando y el bit del puerto 1 se muestra en la tabla 3.8.

BIT	MANDO	DESCRIPCIÓN
0	ABRIR	Abre el interruptor
1	CERRAR	Cierra el interruptor
2	NORMALIZAR	Normaliza el recierre del interruptor
3	BLOQUEAR	Bloquea el recierre del interruptor
4	----	----
5	----	----
6	----	----
7	----	----

Tabla 3.8 Relación entre los mandos y los bits del puerto 1.

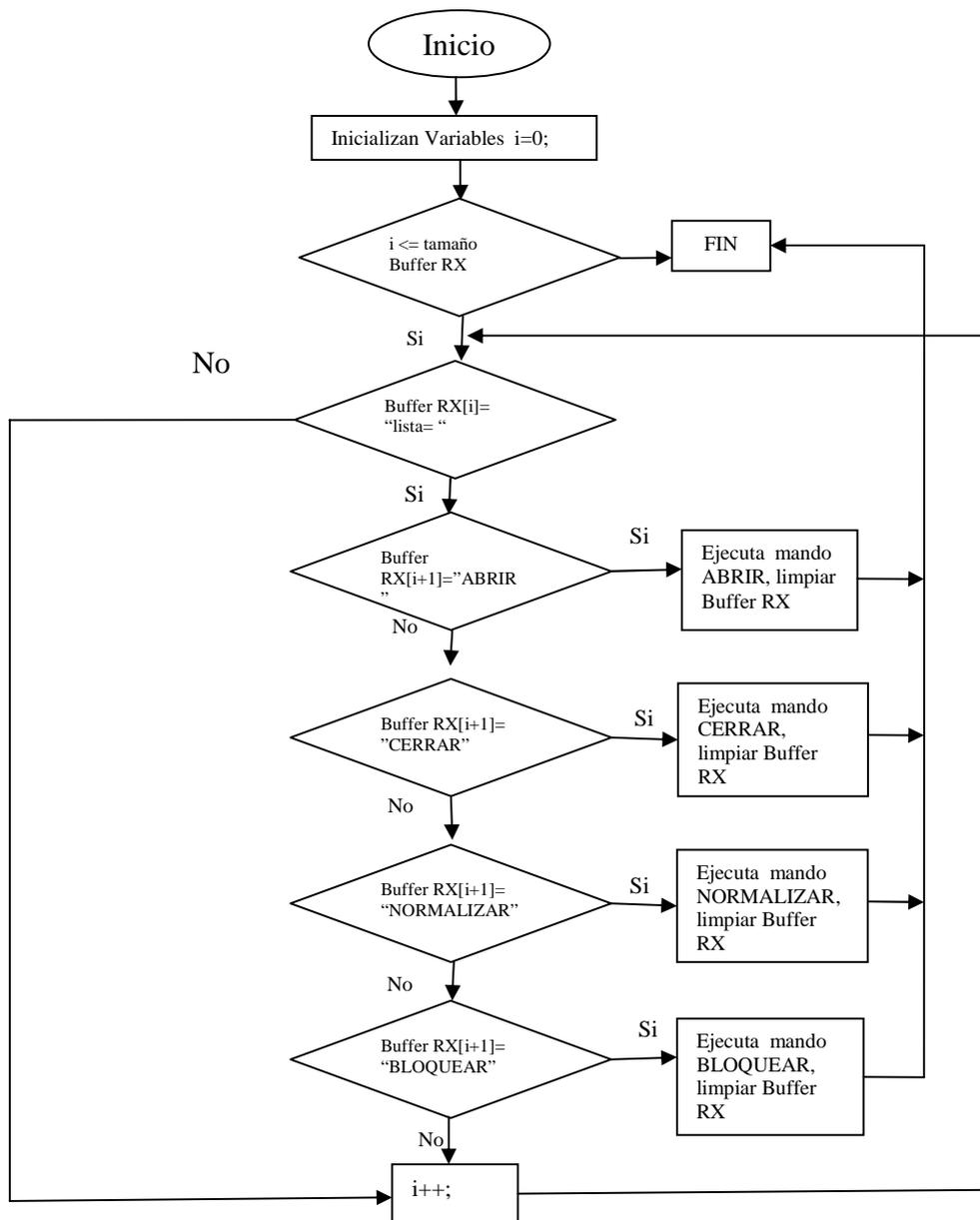


Figura 3.10 Algoritmo para la función que identifica y ejecuta mandos sobre el interruptor.

Solo se utilizan los 4 bits menos significativos del puerto 1, normalmente el puerto 1 tendrá como valor de salida 0, pero cuando se realice un mando será habilitado el bit correspondiente a este mando por un momento y después se volverá a poner en un valor de 0. El pulso que se envía por mando será de 9000 ciclos de reloj, que equivale aproximadamente a 1×10^{-3} s, debido a que es un tiempo muy pequeño, vía hardware se implementará un circuito que amplíe el ancho del pulso a 1 segundo, que es el tiempo que normalmente se le da a cada mando por el sistema de control remoto (UTR's) con que actualmente cuenta la CFE. Se

determinó que este tiempo de 1 segundo por mando fuera realizado vía hardware, debido a que al realizarlo vía software puede provocar pérdidas en la conexión, ya que el programa trabaja de manera secuencial y mantener un proceso de espera por un segundo retrasa la operación general del software, lo cual genera errores de conexión por “timeout” y cierra la conexión.

En la tabla 3.9 se muestra el valor que tomará el puerto 1 por cada mando.

MANDO	VALOR HEXADECIMAL	VALOR BINARIO
ABRIR	0x01	00000001
CERRAR	0x02	00000010
NORMALIZAR	0x04	00000100
BLOQUEAR	0x08	00001000

Tabla 3.9 Valor del puerto 1 para cada mando.

El buffer RX del programa es un arreglo de números enteros extendidos sin signo, los cuales son números de 2 bytes. A continuación se revisa el buffer RX después de recibir los datos correspondientes al mando ABRIR. Después de depurar el programa en varias ocasiones, se observó que en el buffer RX los bytes de cada dato están en un orden invertido al enviado, así también, se encontró que la secuencia puede desplazarse un byte, generando dos patrones que se muestran en las tablas 3.10, 3.11 y 3.12.

Trama enviada para el mando ABRIR														
HEX	67	6c	69	73	74	61	31	67	3d	41	42	52	49	52
ASC	g	l	i	s	t	a	l	g	=	A	B	R	I	R

Tabla 3.10 Trama enviada para el mando *Abir*.

Patrón 1 en buffer RX para mando ABRIR							
HEX	6c 67	73 69	61 74	67 31	41 3d	52 42	52 49
ASC	l g	s i	a t	g l	A =	R B	R I

Tabla 3.11 Patrón 1 en el buffer RX para la recepción de la trama para el mando *Abir*.

Patrón 2 en buffer RX para mando ABRIR								
HEX	67 **	69 6c	74 73	31 61	3d 67	42 41	49 52	00 52
ASC	g *	i l	t s	l a	= g	BA	I R	0 R

Tabla 3.12 Patrón 2 en el buffer RX para la recepción de la trama para el mando Abrir.

Patrones similares al del mando ABRIR en el buffer RX son presentados por los otros mandos, por esta razón, se determinó realizar la búsqueda en primera instancia de una secuencia de caracteres que todos los mandos contuvieran. Observando la secuencia de los datos en el buffer RX, se determinó que las secuencias a buscar iniciarán con “gl” o “=g” con esto se determina que la siguiente celda del buffer RX contiene los caracteres suficientes para identificar el mando que ha sido enviado. En las tablas 3.13 y 3.14 se muestra la identificación de los mandos a partir de los caracteres detectados.

Para el patrón 1, los caracteres a buscar después de “gl” (0x6731)		
CARACTERES QUE INDICAN EL MANDO EN LA SIGUIENTE CELDA DEL BUFFER RX	VALOR HEXADECIMAL EN LA CELDA DEL BUFFER RX	MANDO
A =	41 3d	ABRIR
C =	43 3d	CERRAR
N =	4e 3d	NORMALIZAR
B =	42 3d	BLOQUEAR

Tabla 3.13 Identificación de los caracteres que indican el mando recibido en el buffer RX, para el patrón de recepción 1.

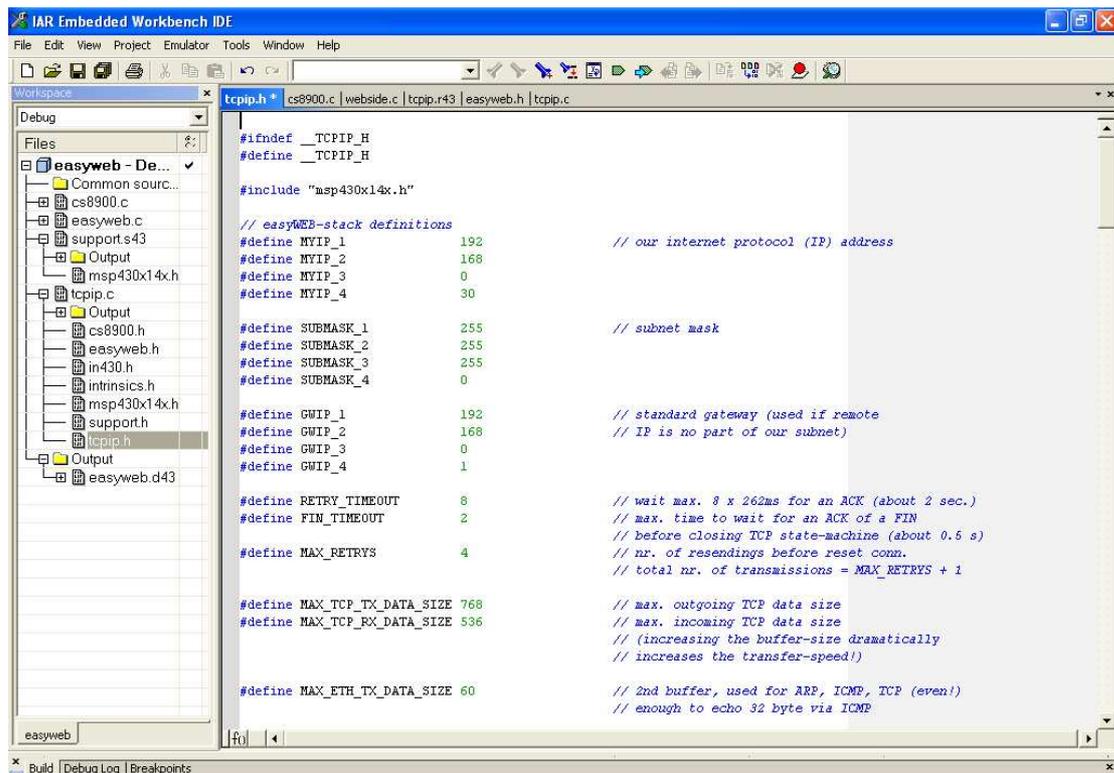
Para el patrón 2, los caracteres a buscar después de “= g” (0x3d67)		
CARACTERES QUE INDICAN EL MANDO EN LA SIGUIENTE CELDA DEL BUFFER RX	VALOR HEXADECIMAL EN LA CELDA DEL BUFFER RX	MANDO
BA	42 41	ABRIR
EC	45 43	CERRAR
ON	4f 4e	NORMALIZAR
LB	4c 42	BLOQUEAR

Tabla 3.14 Identificación de los caracteres que indican el mando recibido en el buffer RX, para el patrón de recepción 2.

Con base en el algoritmo de la figura 3.10 y en las tablas anteriores, se implementó la función en C para la interpretación y ejecución de mandos, el código en lenguaje C también es parte del apéndice C.

Analizando el algoritmo de la figura 3.9 (módulo TCP/IP), se determinó que el punto ideal para insertar el proceso *Identificar y ejecutar mandos* es después del proceso *Analizar y determinar tipo de Trama*, debido a que en ese punto el buffer RX ya cuenta con los datos recibidos, así como con la activación de las banderas correspondientes según el tipo de trama, entonces, al limpiar el buffer RX después de realizar los mandos, no se afectará la operación de este módulo de software.

Una vez que se ha desarrollado la página de interfaz de usuario, la función que actualiza el estado de los parámetros del interruptor, la función que inserta los valores dinámicos del estado de los parámetros y la función que interpreta y ejecuta los mandos, solo falta asignar una dirección IP que corresponda con los segmentos de red de la intranet de CFE, esto se realiza en el archivo tcpip.h del proyecto, en la figura 3.11 se muestra este archivo.



```
#ifndef __TCPIP_H
#define __TCPIP_H

#include "msp430x14x.h"

// easyWEB-stack definitions
#define MYIP_1 192 // our internet protocol (IP) address
#define MYIP_2 168
#define MYIP_3 0
#define MYIP_4 30

#define SUBMASK_1 255 // subnet mask
#define SUBMASK_2 255
#define SUBMASK_3 255
#define SUBMASK_4 0

#define GWIP_1 192 // standard gateway (used if remote)
#define GWIP_2 168 // IP is no part of our subnet
#define GWIP_3 0
#define GWIP_4 1

#define RETRY_TIMEOUT 8 // wait max. 8 x 262ms for an ACK (about 2 sec.)
#define FIN_TIMEOUT 2 // max. time to wait for an ACK of a FIN
// before closing TCP state-machine (about 0.5 s)
#define MAX_RETRYS 4 // nr. of resendings before reset conn.
// total nr. of transmissions = MAX_RETRYS + 1

#define MAX_TCP_TX_DATA_SIZE 768 // max. outgoing TCP data size
#define MAX_TCP_RX_DATA_SIZE 536 // max. incoming TCP data size
// (increasing the buffer-size dramatically
// increases the transfer-speed!)

#define MAX_ETH_TX_DATA_SIZE 60 // 2nd buffer, used for ARP, ICMP, TCP (even!)
// enough to echo 32 byte via ICMP
```

Figura 3.11 Archivo para modificar la dirección IP de la tarjeta Easyweb 3.

Como se muestra en la figura 3.11, la dirección IP, la máscara de sub-red y la puerta de enlace son definiciones (#define), por lo tanto, para modificar estos parámetros en la tarjeta

Easyweb 3 solo se deben modificar estas definiciones, compilar y descargar nuevamente el software en la tarjeta. Aún no se asigna una nueva dirección IP a la tarjeta, debido a que hasta el momento se considera un equipo de prueba, por esta razón, aún está operando con la dirección IP de prueba, en el momento en que se autorice la operación del sistema en la red de CFE, le será asignada una dirección correspondiente a alguno de sus segmentos.

3.4.2.1.3 Módulo Ethernet.

La principal tarea del módulo Ethernet es encapsular, enviar y recibir datos directamente del hardware de red de la tarjeta Easyweb 3. También genera una señal de reloj que es usada para tener acceso al circuito controlador de red (CS8900) configurarlo, leer y escribir en sus puertos y registros. El archivo en el proyecto que contiene este módulo de software es el cs8900.c. El algoritmo para este módulo de software es mostrado en la figura 3.12. Este módulo no se modificó, debido a que es el encargado de enviar y recibir paquetes de datos mediante el circuito controlador de red, básicamente es un controlador de hardware el cual está implementado para operar de una manera eficiente y apegándose a los estándares establecidos para el manejo de paquetes en una red de área local.

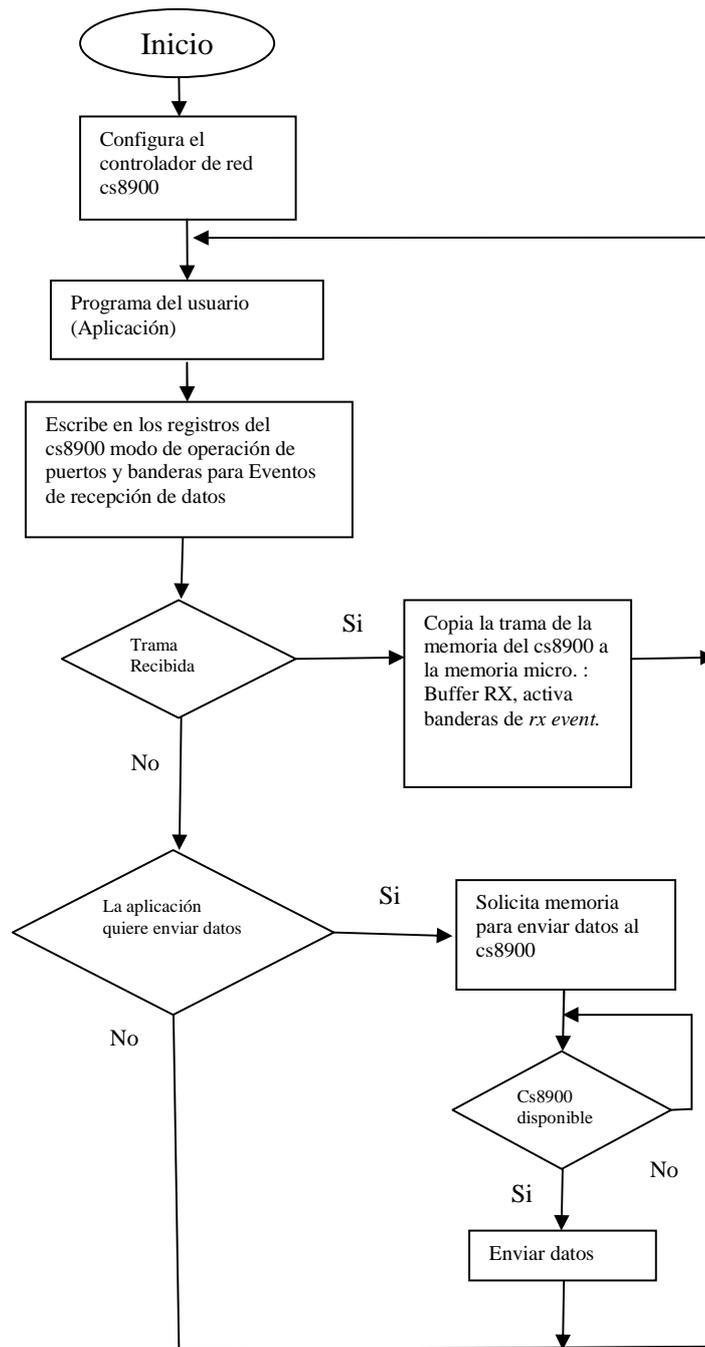


Figura 3.12 Algoritmo para el módulo de software Ethernet.

3.5 DESARROLLO DEL HARDWARE.

El hardware fue desarrollado en función de los requerimientos actuales para la señalización de parámetros y ejecución de mandos del interruptor, con la finalidad de hacer

que el sistema sea completamente compatible con el sistema actual de monitoreo y control con que cuenta la CFE.

La señalización de los parámetros de un interruptor consiste en una serie de contactos normalmente abiertos, los cuales son cerrados en el momento en que el parámetro cambia del estado deseado a uno indeseado (alarmado). Los mandos son realizados mediante un circuito básico de un contacto en serie normalmente abierto; que en el momento en que se desea realizar un mando el contacto es cerrado, dando continuidad a una línea de voltaje de corriente directa (Vcd), energizando un dispositivo que permite la realización del mando deseado.

3.5.1 ETAPA DE ACOPLAMIENTO PARA MANDOS.

La tarjeta Easyweb 3 entrega un pulso de aproximadamente 1ms por mando y como se mencionó anteriormente, el tiempo que el sistema actual de CFE le da a cada mando es de 1 segundo, debido a esto fue necesario implementar un circuito que realice la función de extender el pulso de 1ms a 1s. Después se requiere de una etapa de acoplamiento para aislar el circuito lógico de la etapa de potencia, la etapa de potencia entrega una serie de contactos normalmente abiertos, los cuales van a responder a cada mando cerrándose durante 1s. En la figura 3.13 se muestra un diagrama bloques de la etapa de acoplamiento para realizar los mandos.

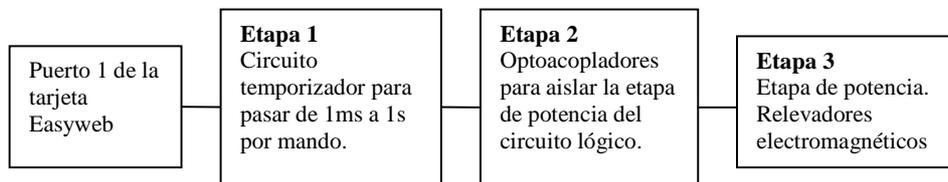


Figura 3.13 Diagrama a bloques de la etapa de acoplamiento para los mandos.

ETAPA 1

Para lograr que el pulso de 1ms que entrega el puerto 1 de la tarjeta Easyweb 3 sea de 1s, se utilizó el circuito integrado 555 que es uno de los temporizadores más comunes, el cual tiene dos modos de operación, monoestable y astable. El modo monoestable se apega a los requerimientos del sistema, ya que en este modo normalmente la salida del 555 es 0V y responde a un cambio de estado de 5V a 0V, en ese momento su salida pasa de 0V a 5V por un tiempo determinado por la ecuación 1[7].

$$t = 1.1RaC \qquad \text{Ec. 1}$$

En la Figura 3.14 se muestra el diagrama esquemático para la configuración monoestable del temporizador 555 [7], en donde se puede ubicar Ra y C.

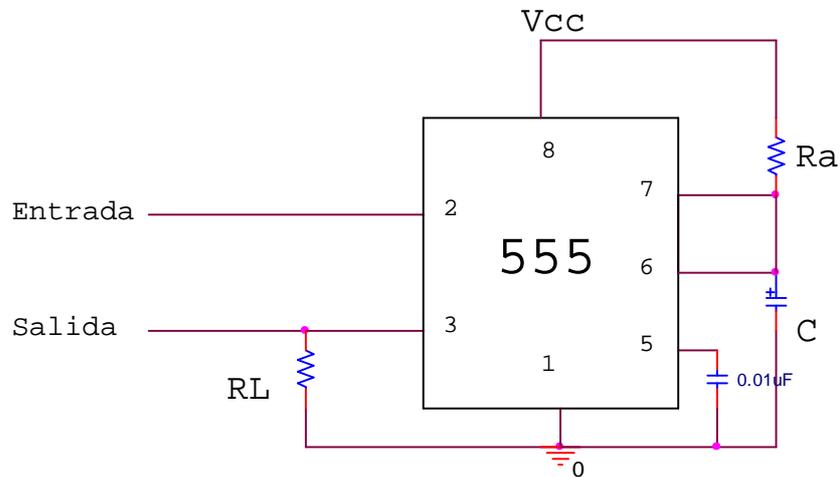


Figura 3.14 Configuración del temporizador 555 en modo monoestable.

Para obtener un pulso de salida de 1s se calcula el valor de Ra y C, considerando $C=10\mu\text{F}$, y al despejar Ra en la ecuación 1, se obtiene la ecuación 2.

$$R_a = \frac{1}{1.1 C} = 90\,909\Omega \approx 90\text{K}\Omega \quad \text{Ec. 2}$$

Con estos valores de Ra y C se construyó y probó el circuito, pero para conectarlo con la tarjeta Easyweb 3, fue necesario utilizar un dispositivo que acople los niveles de voltaje, debido a que la tarjeta Easyweb 3 trabaja a 3.3 Vcd y el temporizador a 5Vcd, para esta función se utilizó un buffer 74LS245, el cual acondiciona los niveles de voltaje dejando transparente el flujo de datos. Solo fue necesario configurar la dirección del flujo de datos.

También fue necesario invertir la salida del puerto 1, esto con la finalidad de hacer que mientras no se tenga ningún mando, la señal se encuentre en 5V y en el momento de realizar el mando se tenga el cambio de estado de 5V a 0V, que es el evento al que responde el temporizador 555, para dar a su salida un nivel de 5V durante un segundo. La inversión de los mandos se realiza de manera externa a la tarjeta Easyweb 3, con la finalidad de evitar que el puerto 1 de la tarjeta esté entregando a su salida voltaje de manera permanente, así evitar algún daño en el puerto o en la fuente de voltaje de la tarjeta y alargar su periodo de vida. En

la figura 3.15 se muestra el diagrama esquemático para el acoplamiento de voltajes entre la tarjeta Easyweb 3 y circuito lógico, así como la inversión de las señales de los mandos.

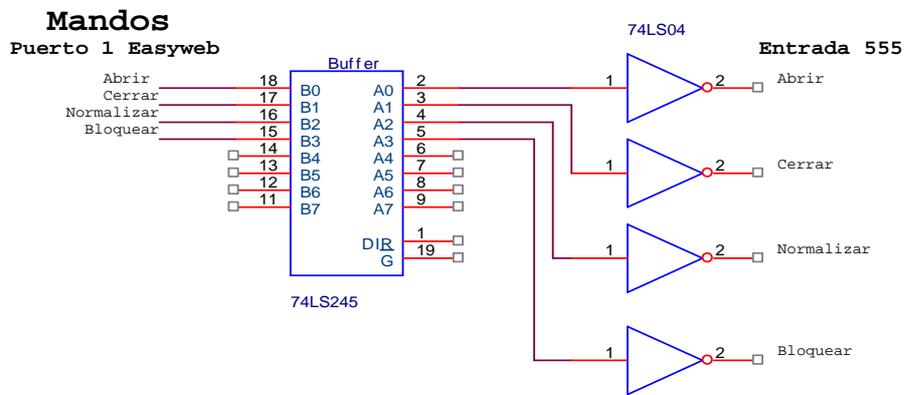


Figura 3.15 Acoplamiento de voltajes e inversión de señales de mandos.

Realizando las conexiones entre la entrada de cada temporizador y las salidas del inversor se tiene el circuito que realiza la función de la etapa 1 de la figura 3.13

ETAPA 2

En esta etapa se requiere diseñar un circuito que proporcione un aislamiento entre la etapa lógica y la etapa de potencia, esto con la finalidad de brindarle una protección a la tarjeta Easyweb 3, para lograr dicho aislamiento se utilizaron optoacopladores MOC3010, los cuales cubren completamente este requerimiento. La entrada de cada circuito de acoplamiento se conecta con la salida de su correspondiente temporizador 555, en la figura 3.16 se muestra el diagrama esquemático para el circuito de acoplamiento.

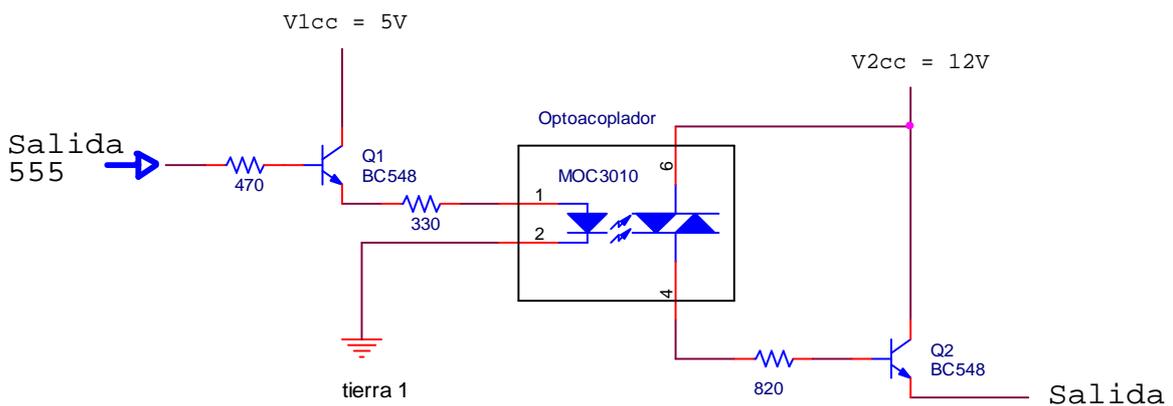


Figura 3.16 Configuración del Optoacoplador para protección del circuito lógico.

En la figura 3.16 se observa el circuito implementado para la etapa 2, la entrada de este circuito es el pulso de 1s que da a su salida el temporizador 555, este pulso energiza la resistencia de 470Ω de la base del transistor Q1(bc548), que está trabajado como interruptor, al ser polarizada la base el transistor, permite el flujo de corriente entre el colector y el emisor para energizar la resistencia de 330Ω y el fotodiodo del optoacoplador, lo que permite que haya flujo de corriente entre los pines 6 y 4 (fototransistor) del optoacoplador, energizando la resistencia de 820Ω de la etapa de potencia y polarizando el transistor Q2, el cual proporciona a su salida un pulso de 12V con una duración de 1s, este pulso energiza la bobina de los relevadores electromagnéticos de la siguiente etapa.

ETAPA 3

Esta etapa consiste en un circuito que proporciona la conmutación de contactos normalmente abiertos a contactos cerrados en el momento en que se realiza un mando, para realizar esta operación se utilizan relevadores electromagnéticos, estos proporcionan dos contactos, uno normalmente abierto (NA) y otro normalmente cerrado (NC), la conmutación de estos contactos ocurre al ser energizada la bobina del dicho relevador, y se mantiene mientras esté energizada. En el momento en que se pierde el voltaje en la bobina los contactos se normalizan, observando la operación de estos relevadores electromagnéticos se determina que se apegan completamente a los requerimientos para esta etapa de hardware, además estos relevadores proporcionan un aislamiento galvánico el cual es una protección adicional al circuito lógico. El diagrama esquemático para esta etapa se muestra en la figura 3.17.

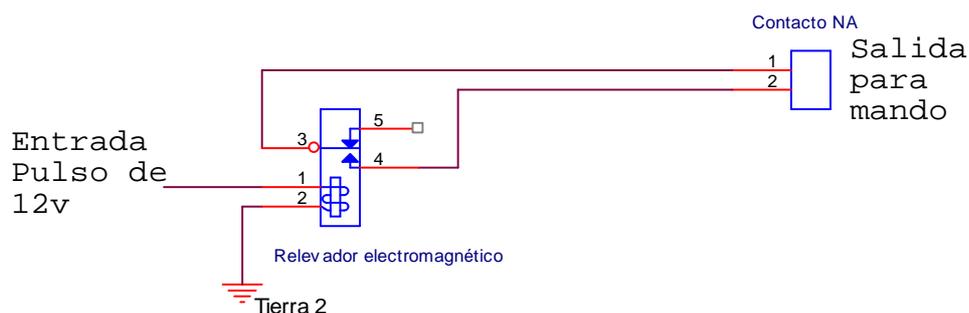


Figura 3.17 Relevador electromagnético y contacto NA para la salida de mando.

En la figura 3.18 se muestra el diagrama esquemático completo para la etapa de acoplamiento para mandos.

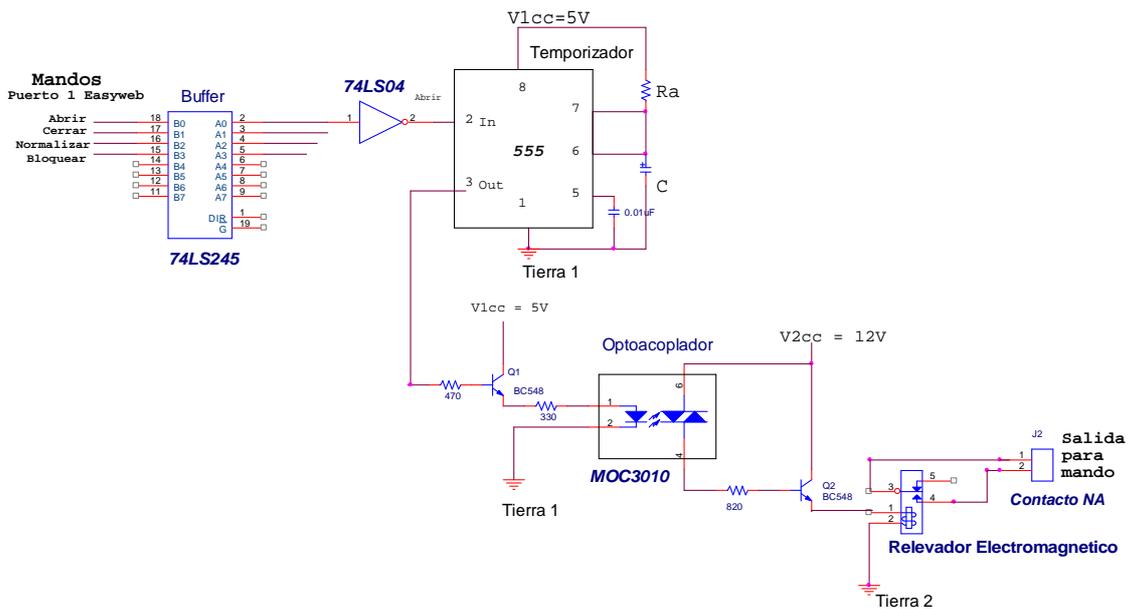


Figura 3.18 Diagrama esquemático completo para la etapa de mandos.

3.5.2 ETAPA DE ACOPLAMIENTO PARA MONITOREO.

Para esta etapa se toma como señal de entrada el contacto NA, que entrega cada parámetro del interruptor, cuando el contacto está abierto se considera como un 0 lógico (color verde en la interfaz de usuario), cuando está cerrado se considera como un 1 lógico (color rojo en la interfaz de usuario). Tomando el contacto NA de cada parámetro como entrada, se requiere que en el momento de ser cerrado cambie el estado de 0 lógico a 1 lógico, para lograr esta relación entre contactos y el circuito lógico, se debe considerar una etapa de protección galvánica para el circuito lógico, una etapa de optoacopladores para aislar la etapa de potencia y la etapa lógica. A la salida de estas etapas se tendrá un bus de datos de 6 bits, el cual es leído por la tarjeta Easyweb 3, y en función de su contenido asigna el valor correspondiente a cada parámetro en la página de interfaz de usuario. El diagrama a bloques para esta etapa se muestra en la figura 3.19.

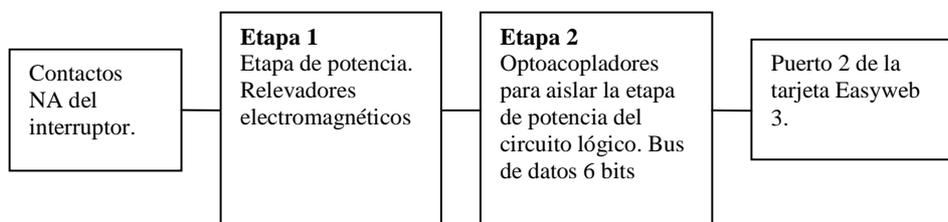


Figura 3.19 Diagrama a bloques de la etapa de acoplamiento para el monitoreo.

ETAPA 1

Para esta etapa se utilizó el contacto NA, que entrega cada parámetro del interruptor para energizar la bobina de un relevador electromagnético en el momento que se genera el cambio de estado (Alarma) en algún parámetro del interruptor. El voltaje positivo que energiza la bobina del relevador va a estar en un punto del contacto NA que entrega el interruptor por parámetro y el negativo de este voltaje será un negativo firme en el otro extremo de la bobina. Cuando ocurre un cambio de estado el contacto se cierra y deja pasar la línea de voltaje de corriente directa (12Vcd) energizando la bobina del relevador, ésta a su vez realiza la conmutación de sus contactos, de los cuales se utiliza el contacto NA para que en el momento en que es energizada la bobina del relevador este contacto se cierre dejando pasar un voltaje que servirá para activar el dispositivo de la siguiente etapa. En la figura 3.20 se muestra el diagrama para esta etapa.

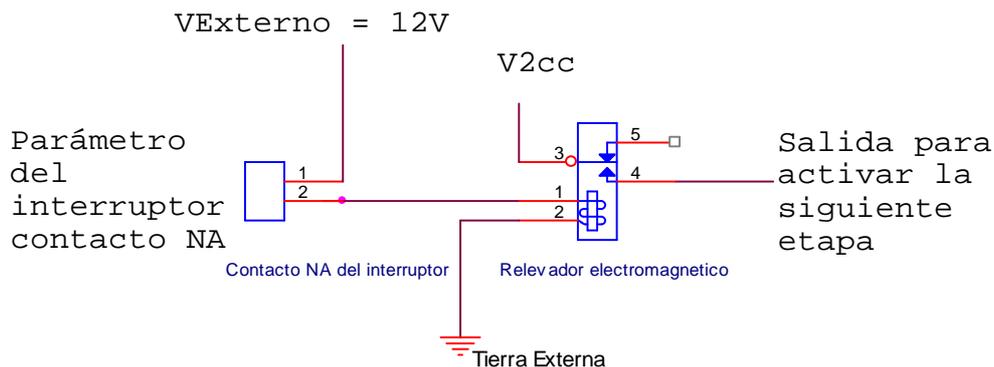


Figura 3.20 Etapa para la protección galvánica del circuito.

ETAPA 2

En esta etapa se utilizan optoacopladores para aislar la etapa de potencia que entregan los parámetros del interruptor de la tarjeta Easyweb 3. La entrada es el voltaje de salida de la etapa 1, mediante este voltaje se polariza un transistor que alimenta el fotodiodo de un optoacoplador, permitiendo el flujo de corriente en las terminales 4 y 6 del mismo para energizar una resistencia de fijación a tierra (*Pull Down*). En esta resistencia normalmente se tiene 0 lógico y en el momento de ser energizada proporciona un 1 lógico. Cada resistencia corresponde con un bit del bus de datos de 6 bits, el cual será leído en el puerto 2 de la tarjeta Easyweb 3. La relación entre el parámetro del interruptor y el número de bit del puerto 2 se

muestra en la tabla 3.15. Esta relación corresponde a la utilizada en la asignación de variables en el software (tabla 3.3).

NUM. DE BIT DEL PUERTO 2	PARÁMETRO QUE MONITOREA DEL INTERRUPTOR
0	Estado del interruptor
1	Estado del recierre
2	Bobina de Cierre
3	Alimentación del sistema de Eléctrico
4	Estado del sistema mecánico o de compresión
5	Bobina de Apertura
6	----
7	----

Tabla 3.15 Relación entre el parámetro del interruptor y el número de bit del puerto 2

En la figura 3.21 se muestra el diagrama esquemático para la etapa 2 de este módulo de hardware.

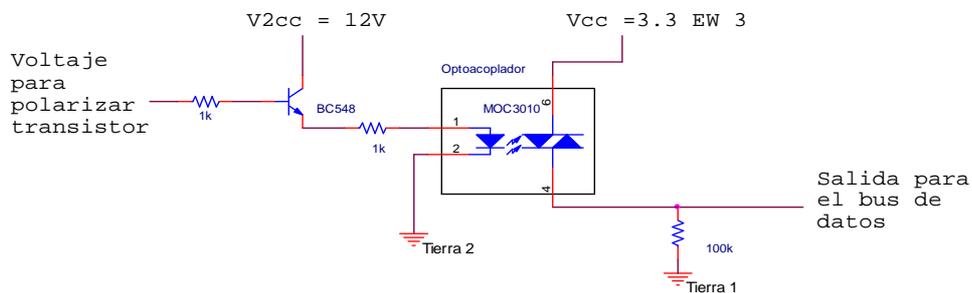


Figura 3.21 Optoacoplador y bit para el bus de datos para el puerto 2.

El diagrama esquemático completo para la etapa de acoplamiento para el monitoreo de un parámetro del interruptor se muestra en la figura 3.22.

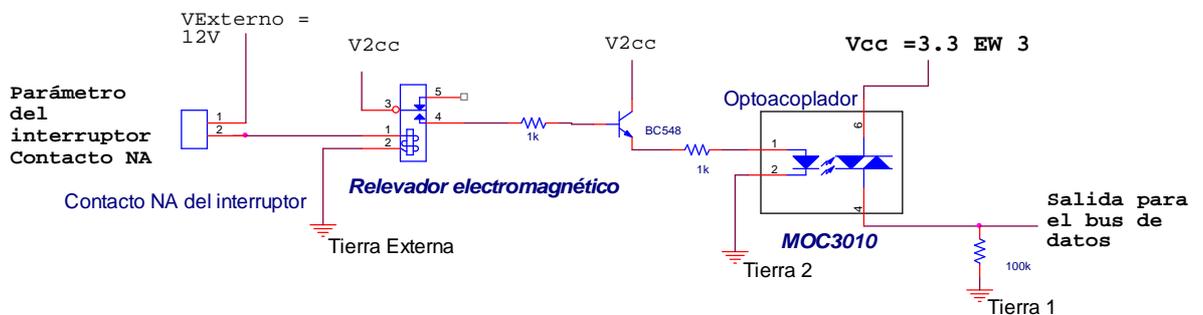


Figura 3.22 Diagrama esquemático completo para el monitoreo de un parámetro.

El diagrama completo para las etapas de acoplamiento de monitoreo y control de los parámetros de un interruptor de circuito de distribución se encuentran el apéndice D, junto con la lista de componentes. El diagrama esquemático fue realizado en el programa Capture CIS de la paquetería Orcad 9.2.

Cada una de las etapas de acoplamiento implementadas en el presente trabajo se diseñaron para cumplir con los requerimientos de operación y señalización de los interruptores de circuito distribución, con la finalidad de no realizar ninguna modificación a las instalaciones de la CFE en el momento de poner en operación el sistema.

Una vez realizadas las pruebas correspondientes en protoboard a cada una de las etapas, se continuó con el ruteo de pistas y el desarrollo del circuito impreso. En el apéndice D también se encuentran los diagramas correspondientes.

CAPÍTULO 4. PRUEBAS Y RESULTADOS

En este capítulo se describen las pruebas realizadas y los resultados obtenidos de cada uno de los módulos que conforman este sistema de monitoreo y control. Las pruebas se realizaron primero en el laboratorio, simulando los mandos y alarmas del interruptor, para observar si el sistema tiene una operación correcta y estable. Después de tener cada una de las etapas del sistema operando de manera correcta, se procedió a realizar la integración del sistema para observar que su funcionamiento sea el esperado, por último se realizó la prueba en campo (subestación de distribución) para observar que la operación del sistema de monitoreo y control de los parámetros de un interruptor de circuito de distribución vía IP sea correcta.

4.1 INTERFAZ DE USUARIO.

La página de interfaz de usuario fue diseñada en lenguaje HTML, en un editor de texto, con la finalidad de realizar una visualización previa e identificar posibles problemas cuando se envíe la página desde la tarjeta Easyweb 3. Los valores dinámicos no son introducidos hasta que el código HTML de la página es adherido al código del programa de la tarjeta, se probó con segmentos de código estáticos que indican un color (Verde/Rojo). En la figura 4.1 se muestra la página de interfaz de usuario antes de ser introducida al código del programa de la tarjeta.

Esta revisión inicial también sirvió para observar que la distribución de información sea la deseada y que la presentación general de la página sea la esperada, para que en el momento que sea incluida en el código del programa de la tarjeta, se observe si existe algún desperfecto en la forma de enviar los datos al usuario.

Para realizar las pruebas referentes al envío de datos para cada mando, se utilizó el software Ethereal, que es una herramienta para el análisis de paquetes. En la figura 4.2 se muestra la captura de las tramas enviadas para la ejecución de los mandos, esto se hizo con la finalidad de observar que los caracteres que se envían sean los deseados, así al realizar la función que analiza e interpreta estos datos ya se cuenta con una referencia de la secuencia de caracteres a esperar por cada mando.

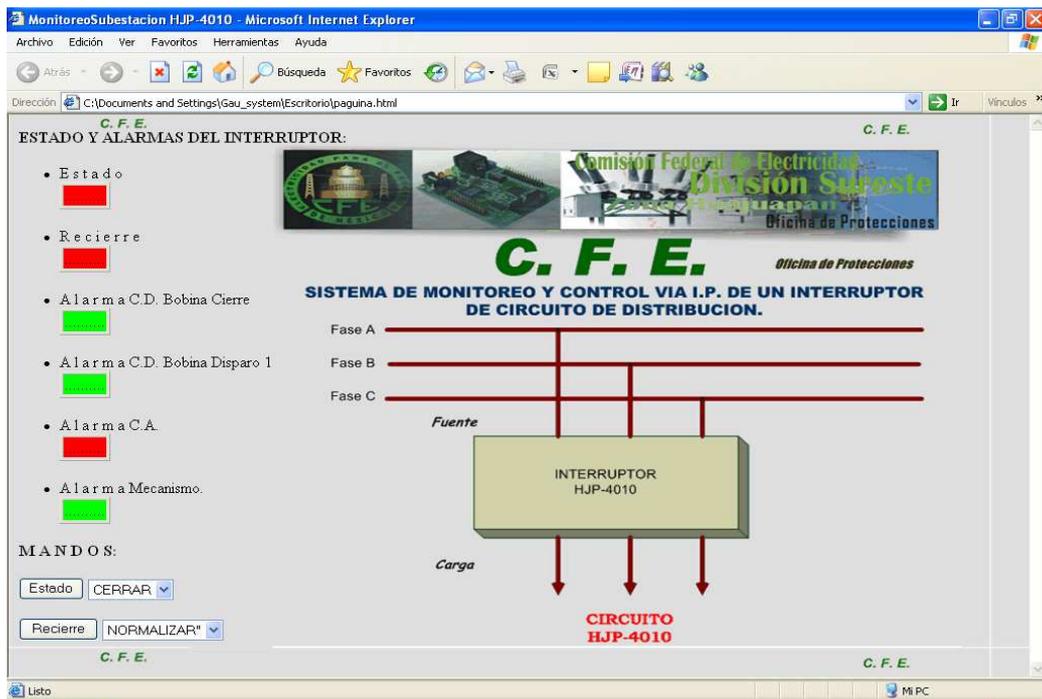


Figura 4.1 Diseño de la página de interfaz de usuario.

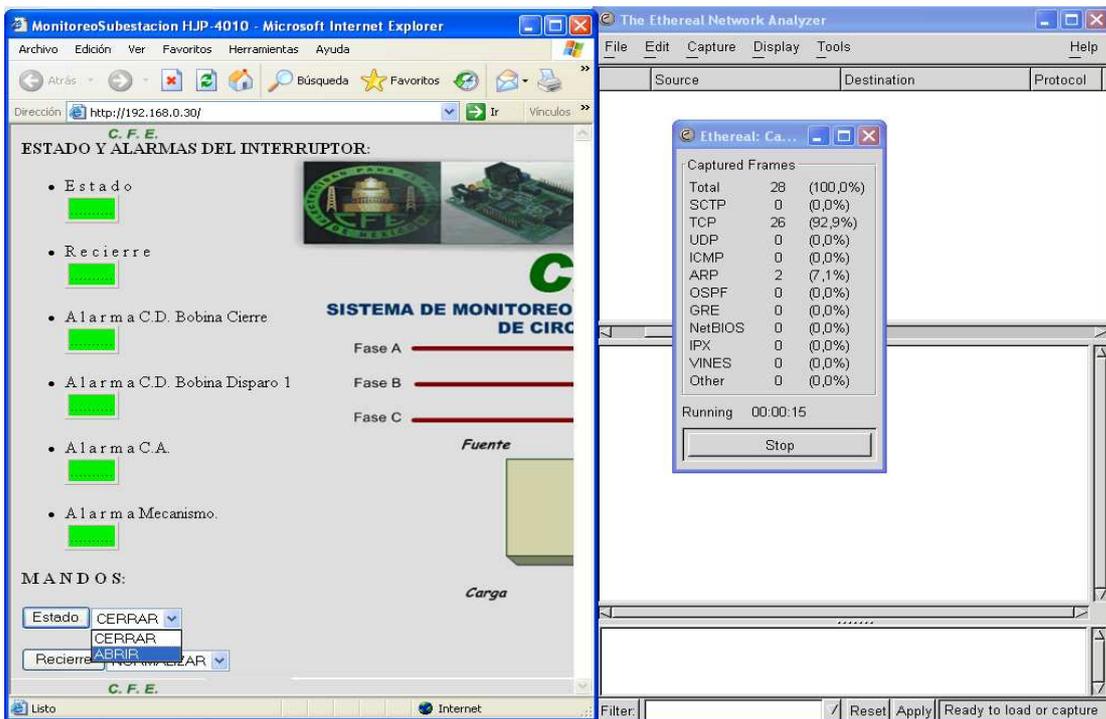


Figura 4.2 Captura de tramas enviadas para los mandos.

En la figura 4.3 se muestra la trama capturada por el software Ethereal para el mando *Cerrar*.

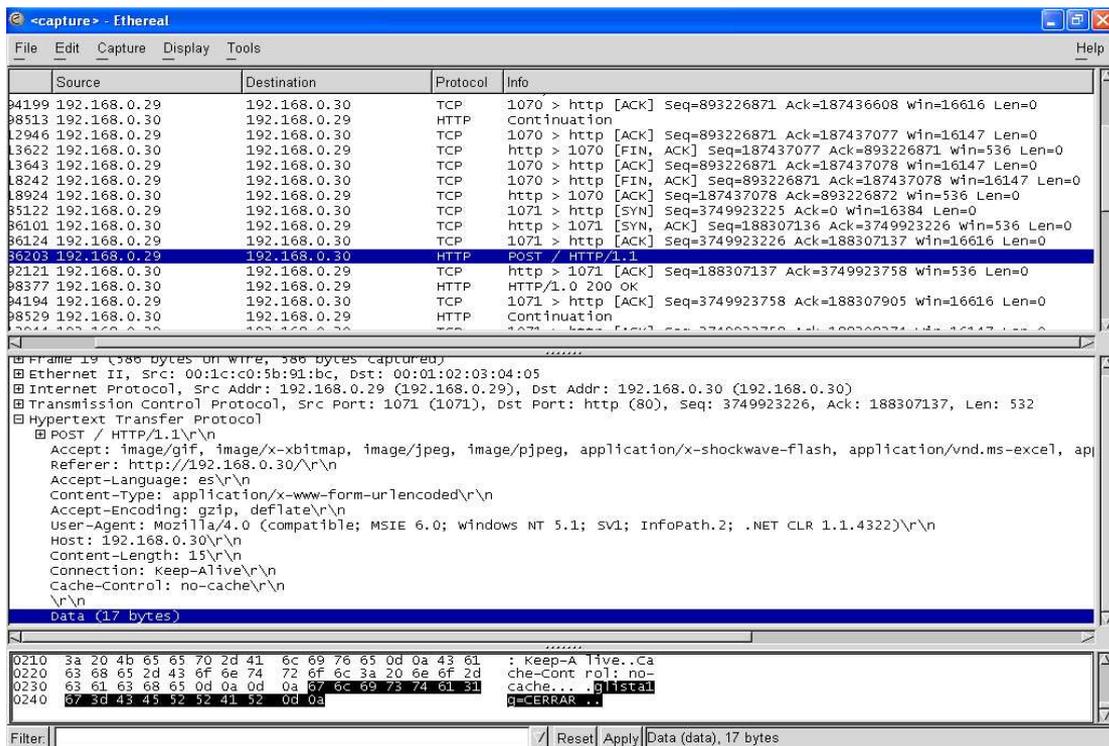


Figura 4.3 Trama capturada por el software Ethereal para el mando Cerrar.

Se realizó esta operación para cada uno de los mandos, para disponer de todas las secuencias de caracteres enviadas a la tarjeta.

4.2 MONITOREO Y CONTROL.

Para realizar las pruebas del monitoreo se construyó un circuito con un arreglo de interruptores y resistencias, el cual simula el cambio de estado de cada uno de los parámetros del interruptor. En la figura 4.4 se muestra una imagen de este circuito y la conexión con la tarjeta Easyweb 3.

Después de realizar todas las combinaciones posibles para estos 6 parámetros, se observó que la respuesta en la página de interfaz de usuario siempre fue la esperada, con esta prueba se determinó que esta etapa del sistema funciona correctamente.



Figura 4.4 Circuito de prueba para simular los cambios de estado de los parámetros del interruptor y observar su correcta señalización en la página de interfaz de usuario.

Para realizar las pruebas a los mandos se colocaron 4 led's a la salida del puerto 1 (P1.0, P1.1, P1.2, P1.3) de la tarjeta Easyweb3 y se alargó el pulso de salida por cada mando de 1ms a 500ms, con la finalidad de observar el destello de cada led, se realizó en varias ocasiones la ejecución de cada mando y se observó que la respuesta siempre fue la esperada, se identificaron y ejecutaron los mandos de manera correcta en todos los casos. En la figura 4.4 se observan los 4 led's colocados en la salida del puerto 1 de la tarjeta Easyweb 3.

4.3 ETAPAS DE ACOPLAMIENTO DE MONITOREO Y CONTROL.

Todos los módulos de las etapas de acoplamiento, tanto de monitoreo y de control, fueron probados en protoboard antes de realizar el diseño final de la placa de circuito impreso, una vez terminada la placa de circuito impreso se realizaron pruebas a cada etapa para verificar su buen funcionamiento

Para la etapa de monitoreo, se simuló el contacto NA que entrega cada parámetro del interruptor mediante un cable, el cual hacía la función de un puente entre cada terminal de la tarjeta y se midió el voltaje en el punto donde el puerto 2, de la tarjeta Easyweb 3, realizará la lectura de estos parámetros. La prueba consistió en verificar que cuando el contacto está abierto, el bit correspondiente tiene un 0 lógico (0Vcd), y cuando el contacto este cerrado tiene un 1 lógico (3.3Vcd). En la figura 4.5 se muestra una imagen en donde se observa la forma en que se realizó esta prueba para la alarma de estado del interruptor.



Figura 4.5 Prueba realizada a la etapa de acoplamiento para el monitoreo del estado del interruptor.

Esta prueba fue realizada para todas las terminales que monitorean los parámetros del interruptor, en todas el resultado fue el esperado, 0 lógico (0Vcd) para cuando el contacto está abierto y 1 lógico (3.3Vcd) cuando el contacto es cerrado.

Para realizar la prueba de la ejecución de los mandos en la tarjeta de circuito impreso, se simuló el pulso generado por la tarjeta Easyweb 3 y se revisó la continuidad en los contactos de salida que entrega la tarjeta para cada mando. En la figura 4.6 se muestra la prueba realizada para el mando *Bloquear*, donde se observó que después de simular el mando, el contacto NA que entrega esta etapa, cambia su estado a cerrado durante aproximadamente 1 segundo, pasado este tiempo regresa a su estado NA, que es el resultado esperado para esta etapa.

Esta prueba fue realizada a cada uno de los mandos, obteniendo en todos los casos el resultado esperado, que es un contacto que se cierra durante aproximadamente 1 segundo por mando.



Figura 4.6 Prueba realizada a la etapa de acoplamiento para la ejecución del mando *Bloquear*.

Después de realizar estas pruebas se determinó que la operación de la tarjeta de circuito impreso, que contiene la etapa de acoplamiento para el monitoreo de los parámetros del interruptor y la etapa de acoplamiento para la ejecución de mandos, funciona de manera correcta para todos los casos.

Concluidas todas las pruebas a cada una de las etapas por separado, se procede a realizar la integración del sistema; la conexión de los periféricos de la tarjeta Easyweb 3 con la tarjeta de circuito impreso, la conexión de la tarjeta Easyweb 3 a un Switch de 8 puertos para simular la red de intranet de CFE y mediante una computadora conectada a esta red realizar pruebas de acceso al sistema, ejecución de mandos y observación del estado de los parámetros del interruptor. En la figura 4.7 se muestra una imagen donde se observan las conexiones mencionadas para la prueba del sistema. Se realizaron pruebas desde la página de interfaz de usuario para observar que la ejecución de mandos sea correcta, así como la señalización del estado de los parámetros del interruptor. La señalización de los parámetros del interruptor se continuó simulando con puentes que hacen la función de los contactos NA, que entrega cada parámetro del interruptor.

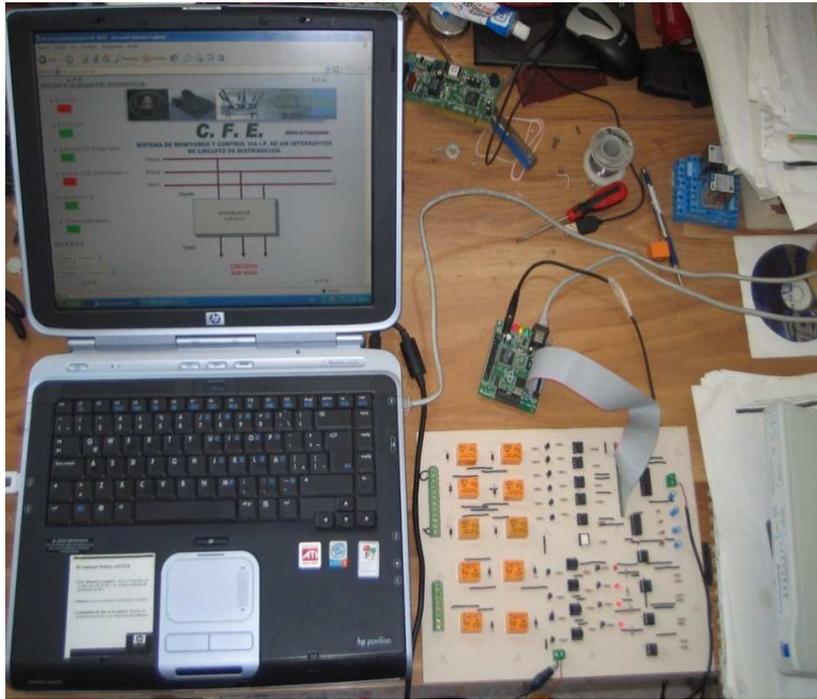


Figura 4.7 Integración del sistema, para pruebas desde la página de interfaz de usuario.

4.4 PRUEBA EN LA SUBESTACIÓN.

La prueba fue realizada en la subestación Tlaxiaco, donde se dispone de un interruptor libre para pruebas. El interruptor para las pruebas es mostrado en la figura 4.8, este interruptor tiene una capacidad de 115KV, de una línea de sub-transmisión. Se utilizó este interruptor debido a que todas sus alarmas y mandos se encuentran conectados a un tablero en el interior de la caseta de control de la subestación, que es donde se pretende colocar la tarjeta Easyweb 3 y sus etapas de acoplamiento. En la caseta se cuenta con las condiciones óptimas para la operación de sus componentes (temperatura y humedad), ahí también se encuentra el equipo para conectarse a la red de CFE. En la figura 4.9 se muestra el interior de la caseta de control de la subestación Tlaxiaco.



Figura 4.8 Interruptor utilizado para probar el sistema de monitoreo y control.



Figura 4.9 Interior de la caseta de control de la subestación Tlaxiaco.

4.4.1 CONEXIÓN DEL SISTEMA DE MONITOREO Y CONTROL AL TABLERO.

Todas las alarmas y mandos del interruptor se encuentran conectados en el interior del tablero, en una tablilla de contactos, como se muestra en la figura 4.10. En estos contactos se encuentran conectados los equipos del sistema actual de monitoreo y control con que cuenta la CFE, para probar el sistema desarrollado se desconectaron estos equipos y se conectaron los

mandos y las alarmas a la tarjeta de acoplamiento, con la finalidad de que ahora el interruptor sólo dependa del sistema diseñado para su monitoreo y control. Para realizar la prueba se conectó la tarjeta EasyWeb3 a la red de CFE y mediante una computadora se realizaron las pruebas de monitoreo de alarmas y ejecución de mandos, en la figura 4.11 se muestra la conexión del sistema diseñado al tablero de control del interruptor.

Las pruebas consistieron en generar todas las alarmas con que cuenta el interruptor y observar su correcta señalización en la página de interfaz de usuario, se realizaron todas las combinaciones posibles y se observó que el sistema siempre mostró el resultado correcto en la página de interfaz de usuario.

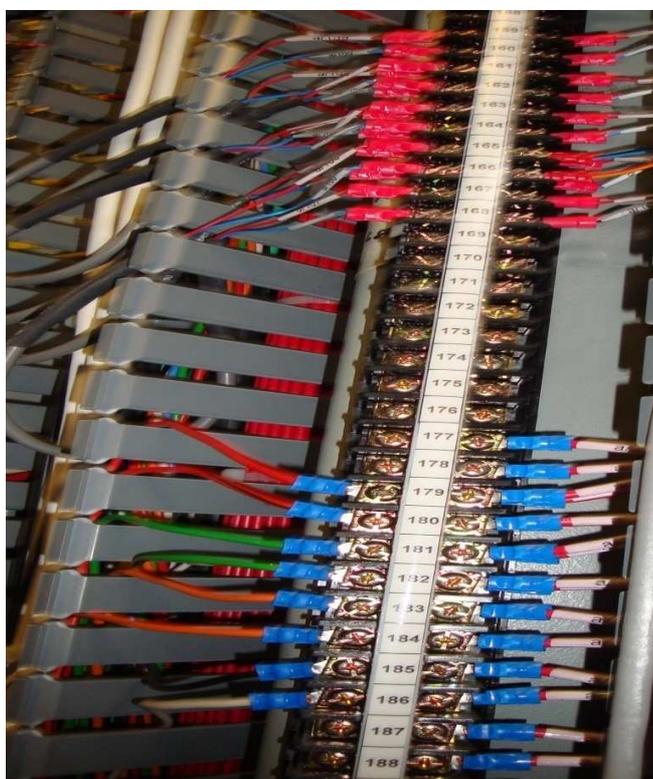


Figura 4.10 Tablilla de contactos al interior del tablero de control de la subestación.

La prueba para los mandos se realizó enviando cada uno de los mandos al interruptor y observar su ejecución correcta en campo, así como su señalización correcta en los led's indicadores (Cuadro de alarmas) con que cuenta el tablero de control del interruptor. Se realizaron pruebas repetitivas para todos los mandos y se observó que siempre se realizaron de manera correcta, así como la señalización de los cambios de estado que estos generan para sus alarmas relacionadas. En la figura 4.12 se muestra el cuadro de alarmas del tablero de control del interruptor después de realizar algunas pruebas.



Figura 4.11 Conexión del sistema al tablero de control del interruptor y la red de intranet de CFE, para realizar las pruebas de monitoreo y control vía IP.



Figura 4.12 Cuadro de alarmas del interruptor.

Todas las pruebas realizadas al interruptor fueron exitosas, ya que el sistema se comportó estable en todo momento, realizó correctamente todos los mandos e identificó de manera correcta el estado de todas las alarmas. Se dejó el sistema operando aproximadamente 2 horas, para observar si se presentaba algún problema en el envío de la página de interfaz de usuario o la realización de algún mando de manera errónea. Después de realizar esta última prueba se observó que el sistema se comportó de manera estable, al término de este periodo se realizó una prueba más de mandos para observar la respuesta del sistema después de estar operando por un tiempo, el resultado fue el esperado ya que el sistema realizó de manera normal todas sus funciones.

CAPÍTULO 5. CONCLUSIONES Y TRABAJO FUTURO

5.1 CONCLUSIONES.

El presente trabajo se enfocó en implementar un sistema de monitoreo y control de un interruptor de circuito de distribución vía IP, con la finalidad de proporcionar un sistema alternativo a la CFE para el monitoreo y control de sus subestaciones de distribución, este sistema se implementó mediante la tarjeta de desarrollo Easyweb 3, la cual cuenta con el microcontrolador MSP430 de Texas Instruments, periféricos de entrada/salida y un circuito controlador de red que permite la fácil conexión a una red de área local.

Se diseñó una interfaz de usuario mediante una página web, la cual permite al usuario observar de manera clara cada uno de los parámetros del interruptor, identificando su estado mediante colores. Esta página permite al usuario realizar mandos sobre el interruptor, cuando se requiera, para realizar esta función, se implementaron 2 botones, uno para controlar el estado del interruptor y otro para controlar el estado del recierre.

La función de monitoreo y control se implementó con el microcontrolador de la tarjeta Easyweb 3, ya que este tipo de dispositivos se prestan para realizar acciones de lectura y almacenamiento, para el monitoreo de los parámetros, y de escritura de puertos, para realizar los mandos. El puerto 2 del microcontrolador fue dedicado al monitoreo de parámetros y el puerto 1 a la ejecución de mandos.

Se realizó un circuito de acoplamiento entre el microcontrolador y el interruptor, este circuito hace que el sistema sea completamente compatible con los requerimientos para la señalización y ejecución de mandos del sistema actual de monitoreo y control con que cuenta la CFE. Con ello, la puesta en operación del sistema no requerirá de alguna modificación a las conexiones actuales. El circuito de acoplamiento es necesario porque la tarjeta Easyweb 3 opera a 3.3Vcd y el interruptor a 125Vcd. Para los mandos se implementó un temporizador para alargar el pulso proporcionado por el microcontrolador, debido a que se requiere de un tiempo aproximado de 1 segundo para que el interruptor realice sus operaciones.

El sistema que se diseñó e implementó opera de manera correcta, es confiable y eficaz, ya que al realizar las pruebas en la subestación con un interruptor real, su operación se mostró estable.

Se concluye que el sistema diseñado e implementado es una alternativa confiable para el monitoreo y control un interruptor de circuito de distribución que utiliza como medio de operación a la intranet de la CFE.

5.2 TRABAJO FUTURO.

Debido a que el presente trabajo solo se centró en realizar un sistema para el monitoreo y control vía IP de un circuito de distribución, la perspectiva para el aumento en la capacidad del sistema queda latente, ya que las subestaciones de distribución con que cuenta la CFE en esta zona, tienen en promedio 3 circuitos de distribución, por lo cual, si se decide poner en operación el sistema diseñado, será necesario realizar una segunda versión en la que se incremente su capacidad para monitorear y controlar por lo menos 3 circuitos de distribución. Esta mejora es posible mediante un acondicionamiento de software y hardware, ya que las funciones básicamente siguen siendo las mismas, mediante software se identificaría al interruptor que se va a trabajar y mediante hardware (multiplexor y buffers) se seccionaría de manera física el interruptor deseado.

En lo referente a la alimentación del sistema, es necesario implementar una fuente de voltaje con respaldo, con la finalidad de garantizar la operación del sistema cuando se tenga algún desperfecto en el sistema de alimentación de la caseta de control, actualmente la alimentación del sistema depende de un UPS (Unit Power System), el cual garantiza su funcionamiento unas horas en el momento en que se quede sin energía eléctrica la caseta de control de la subestación, pero si se presenta alguna falla en el UPS, el sistema se apaga por completo, por esta razón, es necesaria la implementación de una fuente de alimentación con respaldo para garantizar la operación del sistema.

En este trabajo se ha desarrollado una aplicación específica para la tarjeta Easyweb 3, sin embargo, ésta es una plataforma de hardware y software genérica, empleando la misma metodología, podrían desarrollarse aplicaciones diversas que involucren el uso de internet, algunos ejemplos: monitoreo de sensores de una casa o negocio, encendido de luces, control de un invernadero, automatización de una granja, etc.

BIBLIOGRAFÍA

- [1] México, COMISIÓN FEDERAL DE ELECTRICIDAD, *Manual de procedimientos para coordinación de protecciones de sobrecorriente en sistemas de distribución*, Tomo 1, 1997
- [2] México, COMISIÓN FEDERAL DE ELECTRICIDAD, *Manual de diseño de subestaciones eléctricas*, 1999.
- [3] HARPER E., *Introducción al análisis de los sistemas eléctricos de potencia*, Limusa, 1982.
- [4] BERGER A. S., *Embedded Systems Design: An Introduction to Processes, Tools, and Techniques*, 2002.
- [5] GANSSLE J. G., *The Art of Designing Embedded Systems*, 2000.
- [6] LANGSAM Y.; AUGENSTEIN M. J.; TENENBAUM A. M., *Estructuras de datos con C y C++*, segunda edición, Prentice Hall, 1997.
- [7] MALVINO A. P., *Principios de electrónica*, Quinta edición, McGraw-Hill, 1994.
- [8] ARNOLD K., *Embedded Controller Hardware Design*, LLH Technology Publishing, 2001.

URL's.

- [URL1] Texas Instruments, *Especificaciones Microcontrolador MSP430*, <http://www.texasinstruments.com> [Consulta: 15 de agosto 2008].
- [URL2] Raúl Jiménez, *Introducción a los Sistemas empotrados*, <http://www.uhu.es/raul.jimenez/EMPOTRADO/introduccion.pdf> [Consulta: 15 de mayo 2008].
- [URL3] Olimex, *Especificaciones tarjeta de desarrollo Easyweb 3*, <http://www.olimex.com> [Consulta: 06 de marzo 2008].
- [URL4] José Luis Villarroel Salcedo, *Sistemas empotrados*, <http://webdiis.unizar.es/~joseluis/SE.htm> [Consulta: 10 de abril 2008].
- [URL5] *Guía para escribir documentos HTML*, <http://sestud.uv.es/manual.esp/> [Consulta: 10 de julio 2008].

APÉNDICE A. CÓDIGO HTML PARA LA PÁGINA DE INTERFAZ DE USUARIO

```
<html>
<head>
<meta http-equiv="refresh" content="7">
<title>MonitoreoSubestacion HJP-4010</title>
</head>
<body background="C://prueba2.gif">
<p><b>ESTADO Y ALARMAS DEL INTERRUPTOR:</b></p>      <!--monitoreo-->
<UL>
  <LI>E s t a d o
  <TABLE BORDER>
  <TR>
  <TD BGCOLOR="AD%1"> ..... </TD>
  </TR>
  </TABLE>
</UL>
<UL>
  <LI>R e c i e r r e
  <TABLE BORDER>
  <TR>
  <TD BGCOLOR=" AD%2"> ..... </TD>
  </TR>
  </TABLE>
</UL>
<UL>
  <LI>A l a r m a C.D. Bobina Cierre
  <TABLE BORDER>
  <TR>
  <TD BGCOLOR=" AD%3"> ..... </TD>
  </TR>
  </TABLE>
</UL>
<UL>
  <LI>A l a r m a C.D. Bobina Disparo 1
  <TABLE BORDER>
  <TR>
```

```

        <TD BGCOLOR=" AD%6"> ..... </TD>
    </TR>
</TABLE>
</UL>
<UL>
    <LI>Alarma C.A.
    <TABLE BORDER>
    <TR>
    <TD BGCOLOR=" AD%4"> ..... </TD>
    </TR>
    </TABLE>
</UL>
<UL>
    <LI>Alarma Mecanismo.
    <TABLE BORDER>
    <TR>
    <TD BGCOLOR=" AD%5"> ..... </TD>
    </TR>
    </TABLE>
</UL>
    <p><b>M A N D O S: </b></p>
    </Control>
<FORM METHOD="POST" ACTION="http://192.168.0.30">
    <INPUT TYPE="SUBMIT" VALUE="Estado">
    <SELECT NAME="glista1g">
    <OPTION>CERRAR
    <OPTION>ABRIR
    </SELECT>
</FORM>
<FORM METHOD="POST" ACTION="http://192.168.0.30">
    <INPUT TYPE="SUBMIT" VALUE="Recierre">
    <SELECT NAME="elista1g">
    <OPTION>NORMALIZAR"
    <OPTION>BLOQUEAR"
    </SELECT>
</FORM>
</body>
</html>

```

APÉNDICE B. AMBIENTE INTEGRADO DE DESARROLLO *IAR EMBEDDED WORKBENCH IDE*

El ambiente integrado de desarrollo *IAR Embedded Workbench* cuenta con ayuda muy completa, por esta razón, este documento sólo pretende mostrar un resumen del procedimiento que se debe seguir para crear y modificar proyectos. Se tienen los siguientes pasos:

1. Es necesario crear un espacio de trabajo (*Workspace*), esto se hace en el menú: *File* -> *New*-> *Workspace*. Como se muestra en la figura B.1. El “*Workspace*” es el entorno de trabajo, donde se pueden ver los proyectos, los archivos relacionados y modificarlos.

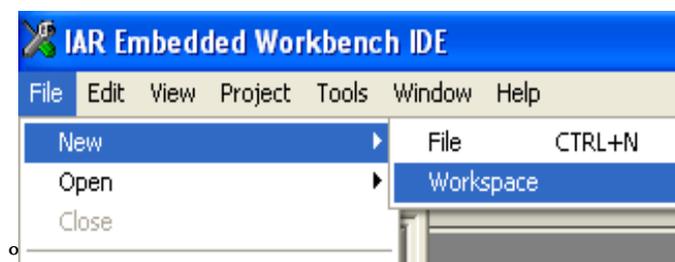


Figura B.1 Creación de un “*Workspace*”.

2. Crear o agregar un proyecto, para crear o agregar un proyecto se va al menú *project*-> *Create new project / add Existing Project*, como se muestra en la figura B.2, si se toma como base el proyecto de ejemplo con que cuenta la tarjeta Easyweb 3, se selecciona la opción *add Existing Project*. En un proyecto son cargados todos los códigos fuente que sean necesarios y las librerías para la correcta ejecución del programa. Al seleccionar *add Existing Project* se muestra una ventana para buscar el proyecto deseado, una vez localizado el proyecto se debe dar un click en el botón *Abrir*, como se muestra en la figura B.3.



Figura B.2 Agregar un proyecto existente a un *Workspace*.

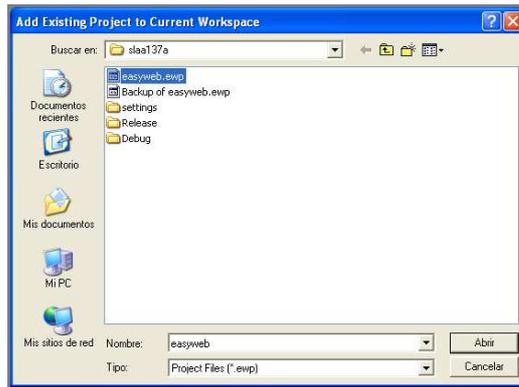


Figura B.3 Agregar y abrir un proyecto existente.

- Una vez agregado el proyecto se procede a modificar los archivos de los códigos fuente del proyecto (archivos .c), para abrirlos se debe seleccionar el archivo deseado en el árbol de archivos y dar 2 clicks, uno de estos archivos se muestra en la figura B.4.

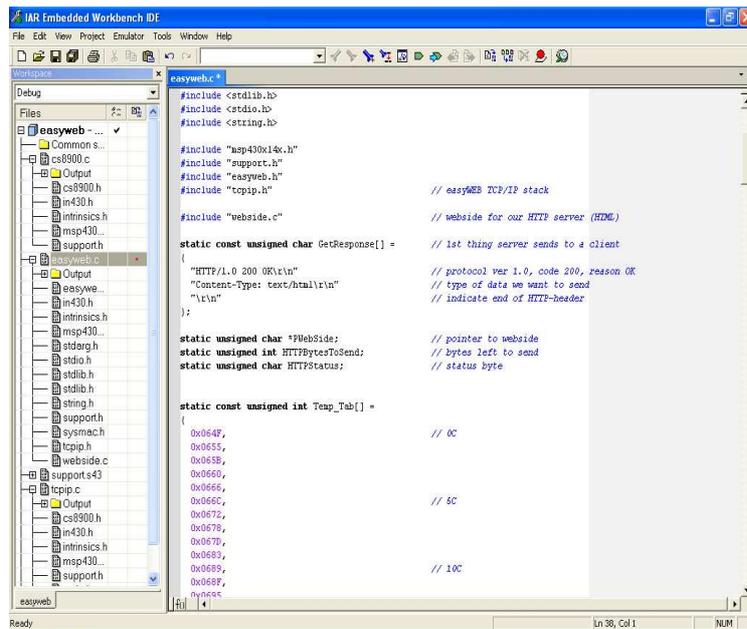


Figura B.4 Modificación de códigos fuente del proyecto.

- Antes de compilar y descargar el programa a la tarjeta se debe configurar algunas propiedades del proyecto, para que la compilación y descarga se realice de manera correcta. Para observar estas propiedades se debe seleccionar con el puntero del ratón en el nombre del proyecto en el árbol de archivos, dar click derecho y seleccionar la opción *Options*, como se muestra en la figura B.5.

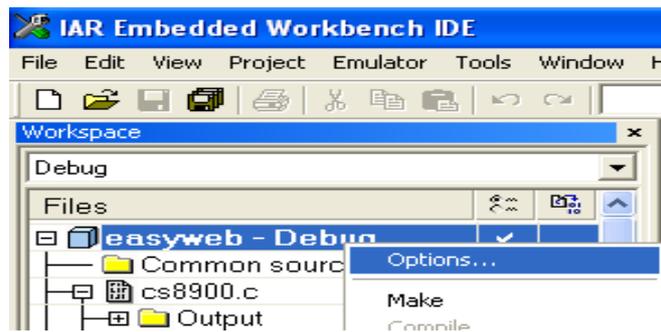


Figura B.5 Propiedades del proyecto.

- En el menú *General Options*, en la pestaña *Target*, en el campo *Device* seleccionar el microcontrolador deseado (MSP430F149), como se muestra en la figura B.6.

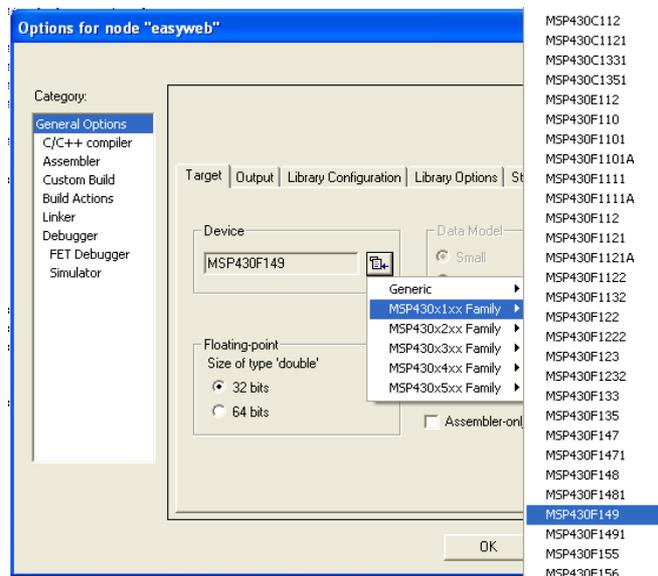


Figura B.6 Seleccionar el microcontrolador con que cuenta la tarjeta Easyweb 3.

- En el menú *FET Debugger*, en la pestaña *Setup*, en el campo *Connection* seleccionar la opción *Texas Instrument LTP-IF* (que indica que la programación será mediante el cable JTAG), y seleccionar el número de puerto, como se muestra en la Figura B.7.

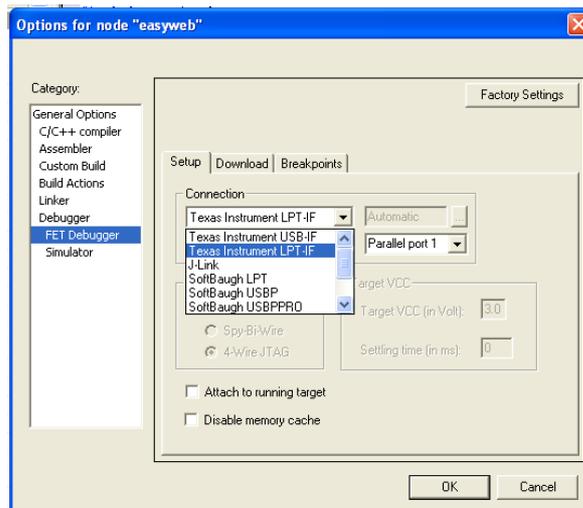


Figura B.7 Seleccionar el medio de programación y el puerto.

5. Una vez que se hayan realizado las modificaciones necesarias a los archivos del proyecto, para ver su efecto en todo el sistema es necesario descargar el programa a la tarjeta, el proceso es el siguiente:
 - Compilar todo el proyecto , para esto ir al menú *project-> Rebuild All*, que guarda y compila todo el proyecto
 - Descargar el software, asegurarse que esté conectada la tarjeta Easyweb 3 a la computadora mediante el cable JTAG, así como alimentada con 6-9Vcd, para evitar algún error. Ir al menú *project->Debug* o con las teclas *Ctrl+D*. se sobrescribe el programa que tiene el microcontrolador de la tarjeta.

APÉNDICE C. FUNCIONES DESARROLLADAS EN LENGUAJE C

Algoritmo de la figura 3.6: Actualización de los valores de las alarmas del interruptor.

<pre>static void Act_Val_Alarmas(void) { unsigned int monitoreog = P2IN; //lectura P 2 if(monitoreog > 31) {Acd2=1; monitoreog=monitoreog-32;} else Acd2=0; if(monitoreog==0) { Aestado=0; ARecierre=0; Acd=0; Aca=0; Amecanismo=0; }else if(monitoreog==1){ Aestado=1; ARecierre=0; Acd=0; Aca=0; Amecanismo=0; }else if(monitoreog==2){ Aestado=0; ARecierre=1; Acd=0; Aca=0; Amecanismo=0; }else if(monitoreog==3){ Aestado=1; ARecierre=1; Acd=0; Aca=0; Amecanismo=0; }else if(monitoreog==4){ Aestado=0; ARecierre=0; Acd=1; Aca=0; Amecanismo=0; }else if(monitoreog==5){ Aestado=1; ARecierre=0; Acd=1; Aca=0; Amecanismo=0; }else if(monitoreog==6){ Aestado=0;</pre>	<pre> ARecierre=1; Acd=1; Aca=0; Amecanismo=0; }else if(monitoreog==7){ Aestado=1; ARecierre=1; Acd=1; Aca=0; Amecanismo=0; }else if(monitoreog==8){ Aestado=0; ARecierre=0; Acd=0; Aca=1; Amecanismo=0; }else if(monitoreog==9){ Aestado=1; ARecierre=0; Acd=0; Aca=1; Amecanismo=0; }else if(monitoreog==10){ Aestado=0; ARecierre=1; Acd=0; Aca=1; Amecanismo=0; }else if(monitoreog==11){ Aestado=1; ARecierre=1; Acd=0; Aca=1; Amecanismo=0; }else if(monitoreog==12){ Aestado=0; ARecierre=0; Acd=1; Aca=1; Amecanismo=0; }else if(monitoreog==13){ Aestado=1; ARecierre=0; Acd=1; Aca=1; Amecanismo=0;</pre>
---	--

<pre> };else if(monitoreog==14){ Aestado=0; ARecierre=1; Acd=1; Aca=1; Amecanismo=0; }else if(monitoreog==15){ Aestado=1; ARecierre=1; Acd=1; Aca=1; Amecanismo=0; }else if(monitoreog==16){ Aestado=0; ARecierre=0; Acd=0; Aca=0; Amecanismo=1; }else if(monitoreog==17){ Aestado=1; ARecierre=0; Acd=0; Aca=0; Amecanismo=1; }else if(monitoreog==18){ Aestado=0; ARecierre=1; Acd=0; Aca=0; Amecanismo=1; }else if(monitoreog==19){ Aestado=1; ARecierre=1; Acd=0; Aca=0; Amecanismo=1; }else if(monitoreog==20){ Aestado=0; ARecierre=0; Acd=1; Aca=0; Amecanismo=1; }else if(monitoreog==21){ Aestado=1; ARecierre=0; Acd=1; Aca=0; Amecanismo=1; }else if(monitoreog==22){ Aestado=0; ARecierre=1; Acd=1; Aca=0; Amecanismo=1; } </pre>	<pre> else if(monitoreog==23){ Aestado=1; ARecierre=1; Acd=1; Aca=0; Amecanismo=1; }else if(monitoreog==24){ Aestado=0; ARecierre=0; Acd=0; Aca=1; Amecanismo=1; }else if(monitoreog==25){ Aestado=1; ARecierre=0; Acd=0; Aca=1; Amecanismo=1; }else if(monitoreog==26){ Aestado=0; ARecierre=1; Acd=0; Aca=1; Amecanismo=1; }else if(monitoreog==27){ Aestado=1; ARecierre=1; Acd=0; Aca=1; Amecanismo=1; }else if(monitoreog==28){ Aestado=0; ARecierre=0; Acd=1; Aca=1; Amecanismo=1; }else if(monitoreog==29){ Aestado=1; ARecierre=0; Acd=1; Aca=1; Amecanismo=1; }else if(monitoreog==30){ Aestado=0; ARecierre=1; Acd=1; Aca=1; Amecanismo=1; }else if(monitoreog==31){ Aestado=1; ARecierre=1; Acd=1; Aca=1; Amecanismo=1; } } </pre>
---	---

Algoritmo de la figura 3.8: Para insertar valores dinámicos a la página de interfaz de usuario.

```

static void Inser_Valor_Dinamicos(void)
{
    char *clave;
    char ColorReal[6];
    int i;
    if (TCPTxDataCount < 4) return;           // no puede haber palabra clave para sustituir.
    clave = (char *)TCP_TX_BUF; //clave apunta al arreglo del buffer de tx.
    for (i = 0; i < TCPTxDataCount - 3; i++) //Ciclo
    {
        if (*clave == 'A')
        if (*(clave + 1) == 'D')
        if (*(clave + 3) == '%')
        if (*(clave + 2) == '1')           //ALARMA ESTADO
        {
            if (Aestado==1)
            {
                ColorReal[0]='F';ColorReal[1]='F';ColorReal[2]='0';ColorReal[3]='0';ColorReal[4]='0';ColorReal[5]='0';
            }
            else
            {
                ColorReal[0]='0';ColorReal[1]='0';ColorReal[2]='F';ColorReal[3]='F';ColorReal[4]='0';ColorReal[5]='0';
            }
            memcpy(clave, ColorReal, 3); //sustituye AD1% por el valor real que indica el color del estado de la alarma.
        }
        else if (*(clave + 2) == '2') //ALARMA RECIERRE
        {
            if (ARecierre==1)
            {
                ColorReal[0]='F';ColorReal[1]='F';ColorReal[2]='0';ColorReal[3]='0';ColorReal[4]='0';ColorReal[5]='0';
            }
            else
            {
                ColorReal[0]='0';ColorReal[1]='0';ColorReal[2]='F';ColorReal[3]='F';ColorReal[4]='0';ColorReal[5]='0';
            }
            memcpy(clave, ColorReal, 3); //sustituye AD2% por el valor real que indica el color del estado de la alarma.
        }
        else if (*(clave + 2) == '3') //ALARMA CDI
        {
            if (Acd1==1)
            {
                ColorReal[0]='F';ColorReal[1]='F';ColorReal[2]='0';ColorReal[3]='0';ColorReal[4]='0';ColorReal[5]='0';
            }
            else
            {
                ColorReal[0]='0';ColorReal[1]='0';ColorReal[2]='F';ColorReal[3]='F';ColorReal[4]='0';ColorReal[5]='0';
            }
            memcpy(clave, ColorReal, 3);
        }
        else if (*(clave + 2) == '4') //ALARMA CA
        {
            if (Aca==1)
            {
                ColorReal[0]='F';ColorReal[1]='F';ColorReal[2]='0';ColorReal[3]='0';ColorReal[4]='0';ColorReal[5]='0';
            }
            else
            {
                ColorReal[0]='0';ColorReal[1]='0';ColorReal[2]='F';ColorReal[3]='F';ColorReal[4]='0';ColorReal[5]='0';
            }
            memcpy(clave, ColorReal, 3);
        }
        else if (*(clave + 2) == '5') //ALARMA MECANISMO
        {
            if (Amecanismo==1)
            {
                ColorReal[0]='F';ColorReal[1]='F';ColorReal[2]='0';ColorReal[3]='0';ColorReal[4]='0';ColorReal[5]='0';
            }
            else
            {
                ColorReal[0]='0';ColorReal[1]='0';ColorReal[2]='F';ColorReal[3]='F';ColorReal[4]='0';ColorReal[5]='0';
            }
            memcpy(clave, ColorReal, 3);
        }
        else if (*(clave + 2) == '6') //ALARMA CD2
        {
            if (Acd2==1)
            {
                ColorReal[0]='F';ColorReal[1]='F';ColorReal[2]='0';ColorReal[3]='0';ColorReal[4]='0';ColorReal[5]='0';
            }
            else
            {
                ColorReal[0]='0';ColorReal[1]='0';ColorReal[2]='F';ColorReal[3]='F';ColorReal[4]='0';ColorReal[5]='0';
            }
            memcpy(clave, ColorReal, 3); //sustituye AD6% por el valor real que indica el color del estado de la alarma.
        }
    }
    clave++; //incrementa al apuntador
}
}

```

Algoritmo de la figura 3.10: Interpretación y ejecución de mandos.

```
static void decod_mandos(void)
{
    extern unsigned int mando;
    int cta=0;
    while(cta<=MAX_TCP_RX_DATA_SIZE)// analizar Buffer de rx
    {
        if(RxTCPBufferMem[cta]==0x3d67) //== 'g'
        {
            mando=RxTCPBufferMem[cta+1];
            if(mando==0x4241){ P1OUT = 0x01;DelayCycles(9000);}//Abrir
            else if(mando==0x4543){ P1OUT = 0x02;DelayCycles(9000);}//Cerrar
            else if(mando==0x4f4e){ P1OUT = 0x04;DelayCycles(9000);}//Normalizar
            else if(mando==0x4c42){ P1OUT = 0x08;DelayCycles(9000);}//Bloquear
            RxTCPBufferMem[cta]=0x00;
            mando=0;
            cta=MAX_TCP_RX_DATA_SIZE; //terminar el ciclo
            P1DIR = 0xff;
            P1OUT = 0x00; //se pone a cero el puerto 1
            int cta3=0;
            while(cta3<MAX_TCP_RX_DATA_SIZE)
            {
                RxTCPBufferMem[cta3]=0x00;cta3++; //se limpia el buffer RX
            }
        }else
        if(RxTCPBufferMem[cta]==0x6731) //== 'gl'
        {
            mando=RxTCPBufferMem[cta+1];
            if(mando==0x413d){ P1OUT = 0x01;DelayCycles(9000);}//Abrir
            else if(mando==0x433d){ P1OUT = 0x02;DelayCycles(9000);}//Cerrar
            else if(mando==0x4e3d){ P1OUT = 0x04;DelayCycles(9000);}//Normalizar
            else if(mando==0x423d){ P1OUT = 0x08;DelayCycles(9000);}//Bloquear
            RxTCPBufferMem[cta]=0x00;
            mando=0;
            cta=MAX_TCP_RX_DATA_SIZE; //terminar el ciclo
            P1DIR = 0xff;
            P1OUT = 0x00; //se pone a cero el puerto 1
            int cta3=0;
            while(cta3<MAX_TCP_RX_DATA_SIZE)
            {
                RxTCPBufferMem[cta3]=0x00;cta3++;
            }
        }
        cta++;
    }
}
```

APÉNDICE D. HARDWARE DESARROLLADO.

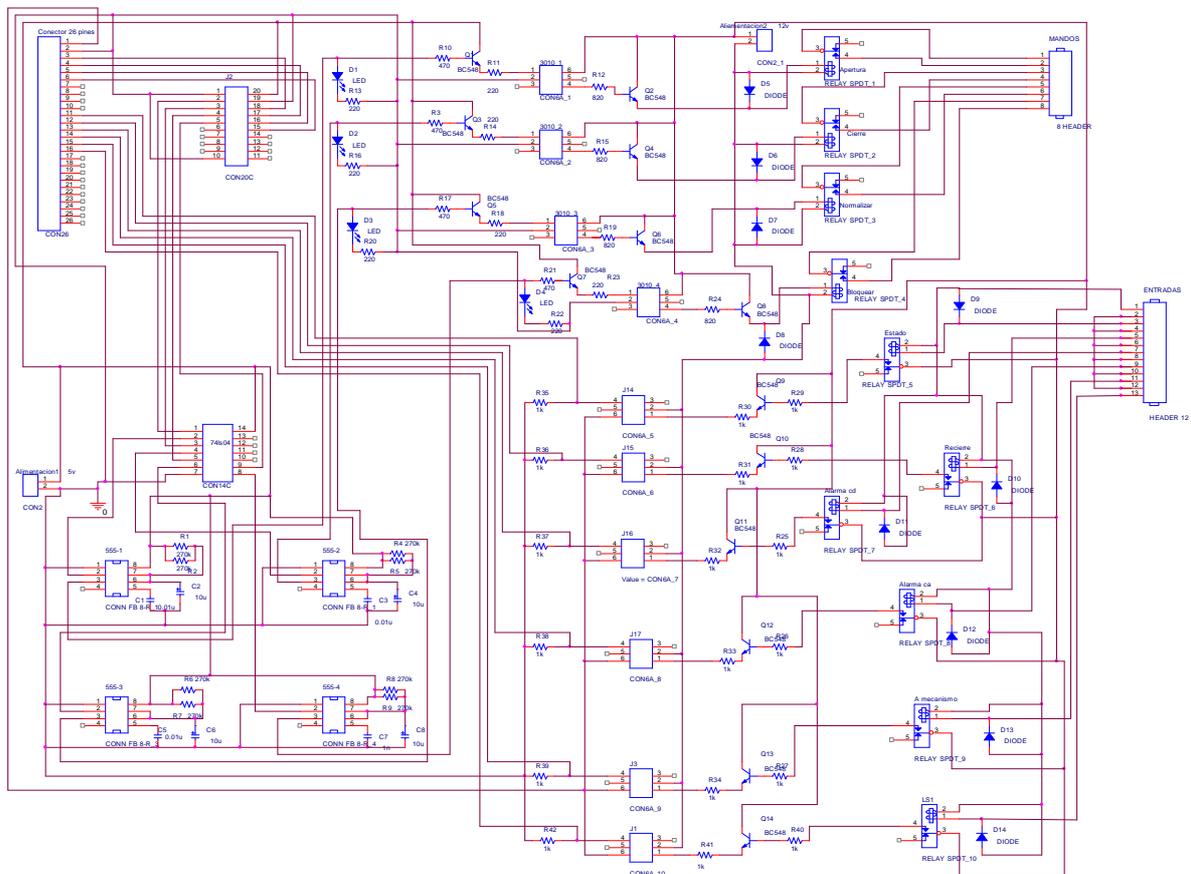


Figura D1 Diagrama completo para las etapas de acoplamiento de monitoreo y control de los parámetros de un interruptor de circuito de distribución.

Tabla D1 Lista de componentes.

NUM	IDENTIFICADOR	COMPONENTE
1	CON26	TIRA DE PINES
2	CON2	TERMINAL TIPO TORNILLO
3	CON2_1	TERMINAL TIPO TORNILLO
4	CON20C	SN74LS245N
5	CON14C	74LS04
6	CONN FB 8-R_1	NE555
7	CONN FB 8-R_2	NE555
8	CONN FB 8-R_3	NE555
9	CONN FB 8-R_4	NE555
10	CON6A_2	MOC3010
11	CON6A_3	MOC3010
12	CON6A_4	MOC3010
13	CON6A_5	MOC3010

NUM	IDENTIFICADOR	COMPONENTE
14	CON6A_6	MOC3010
15	CON6A_7	MOC3010
16	CON6A_8	MOC3010
17	CON6A_9	MOC3010
18	CON6A_10	MOC3010
19	RELAY SPDT_1	RELEVADOR ELECTROMAGNETICO 12V
20	RELAY SPDT_2	RELEVADOR ELECTROMAGNETICO 12V
21	RELAY SPDT_3	RELEVADOR ELECTROMAGNETICO 12V
22	RELAY SPDT_4	RELEVADOR ELECTROMAGNETICO 12V
23	RELAY SPDT_5	RELEVADOR ELECTROMAGNETICO 12V
24	RELAY SPDT_6	RELEVADOR ELECTROMAGNETICO 12V
25	RELAY SPDT_7	RELEVADOR ELECTROMAGNETICO 12V
26	RELAY SPDT_8	RELEVADOR ELECTROMAGNETICO 12V
27	RELAY SPDT_9	RELEVADOR ELECTROMAGNETICO 12V
28	RELAY SPDT_10	RELEVADOR ELECTROMAGNETICO 12V
29	Q1	BC548
30	Q2	BC548
31	Q3	BC548
32	Q4	BC548
33	Q5	BC548
34	Q6	BC548
35	Q7	BC548
36	Q8	BC548
37	Q9	BC548
38	Q10	BC548
39	Q11	BC548
40	Q12	BC548
41	Q13	BC548
42	Q14	BC548
43	D1	LED
44	D2	LED
45	D3	LED
46	D4	LED
47	D5	1N4001
48	D6	1N4001
49	D7	1N4001
50	D8	1N4001
51	D9	1N4001
52	D10	1N4001
53	D11	1N4001
54	D12	1N4001

NUM	IDENTIFICADOR	COMPONENTE
55	D13	1N4001
56	D14	1N4001
57	C1	0,01 μ F
58	C2	10 μ F
59	C3	0,01 μ F
60	C4	10 μ F
61	C5	0,01 μ F
62	C6	10 μ F
63	C7	0,01 μ F
64	C8	10 μ F
65	R1	R=270 Ω
66	R2	R=270 Ω
67	R3	R=470 Ω
68	R4	R=270 Ω
69	R5	R=270 Ω
70	R6	R=270 Ω
71	R7	R=270 Ω
72	R8	R=270 Ω
73	R9	R=270 Ω
74	R10	R=470 Ω
75	R11	R=220 Ω
76	R12	R=820 Ω
77	R13	R=220 Ω
78	R14	R=220 Ω
79	R15	R=820 Ω
80	R16	R=220 Ω
81	R17	R=470 Ω
82	R18	R=220 Ω
83	R19	R=820 Ω
84	R20	R=220 Ω
85	R21	R=470 Ω
86	R22	R=220 Ω
87	R23	R=220 Ω
88	R24	R=820 Ω
89	R25	R=1K Ω
90	R26	R=1K Ω
91	R27	R=1K Ω
92	R28	R=1K Ω
93	R29	R=1K Ω
94	R30	R=1K Ω
95	R31	R=1K Ω

NUM	IDENTIFICADOR	COMPONENTE
96	R32	R=1K Ω
97	R33	R=1K Ω
98	R34	R=1K Ω
99	R35	R=100K Ω
100	R36	R=100K Ω
101	R37	R=100K Ω
102	R38	R=100K Ω
103	R39	R=100K Ω
104	R40	R=1K Ω
105	R41	R=1K Ω
106	R42	R=100K Ω
107	8 HEADER	TERMINALES TIPO TORNILLO
108	12HEADER	TERMINALES TIPO TORNILLO

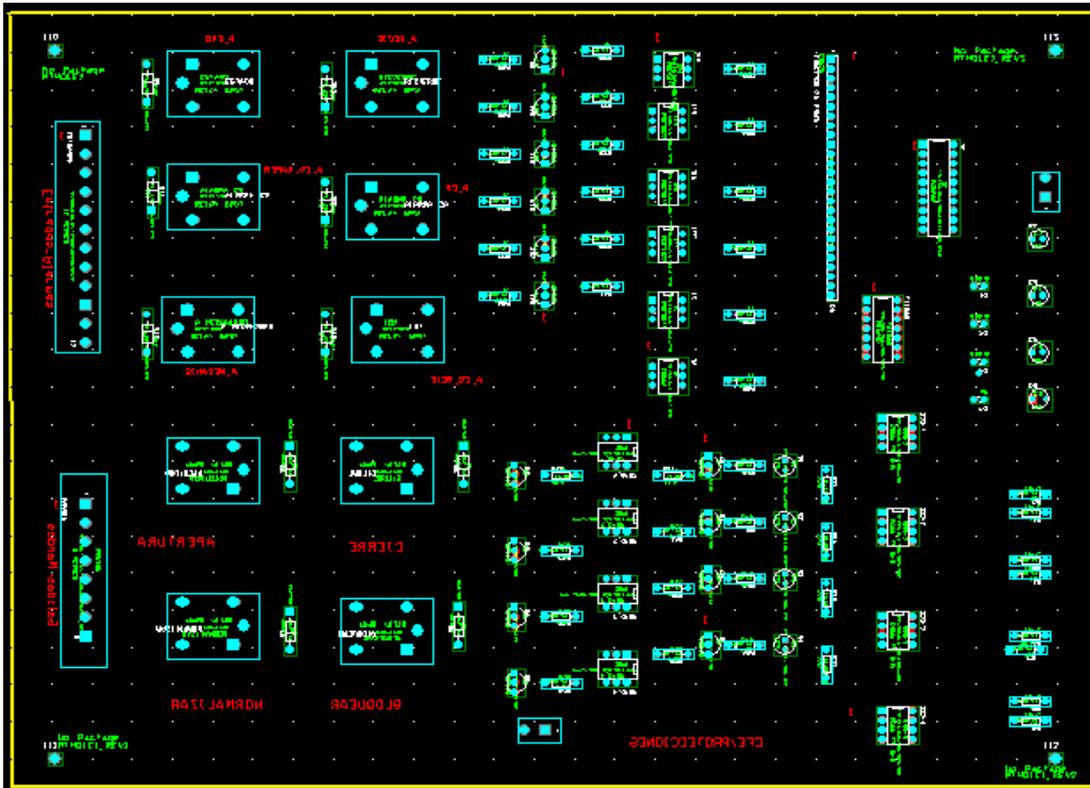


Figura D2 Distribución de componentes en *Layout Plus* para realizar el ruteo del circuito.

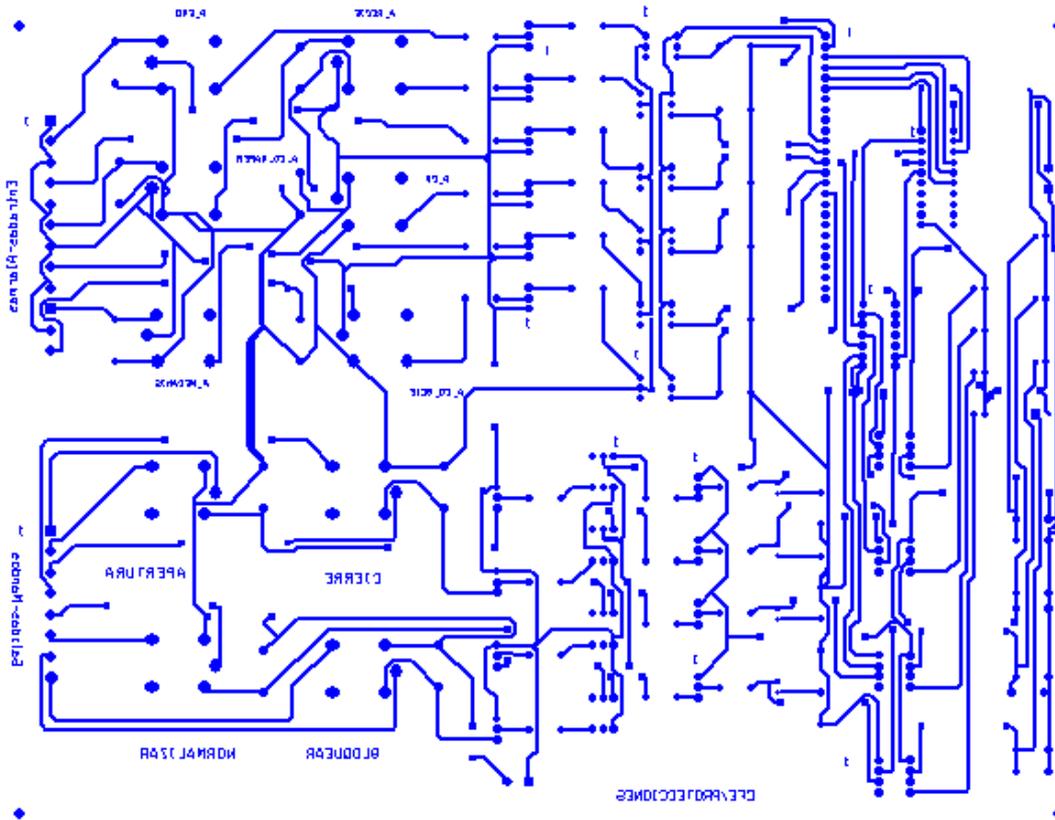


Figura D3 Ruteo de pistas para el circuito impreso.

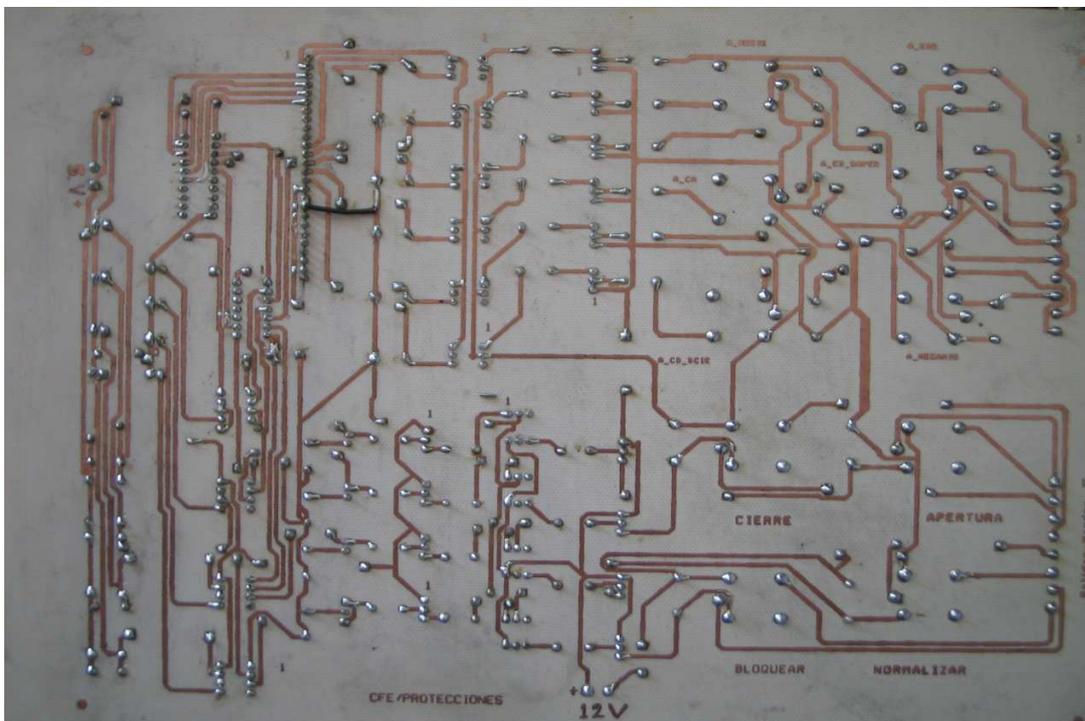


Figura D4 Pistas del circuito impreso.

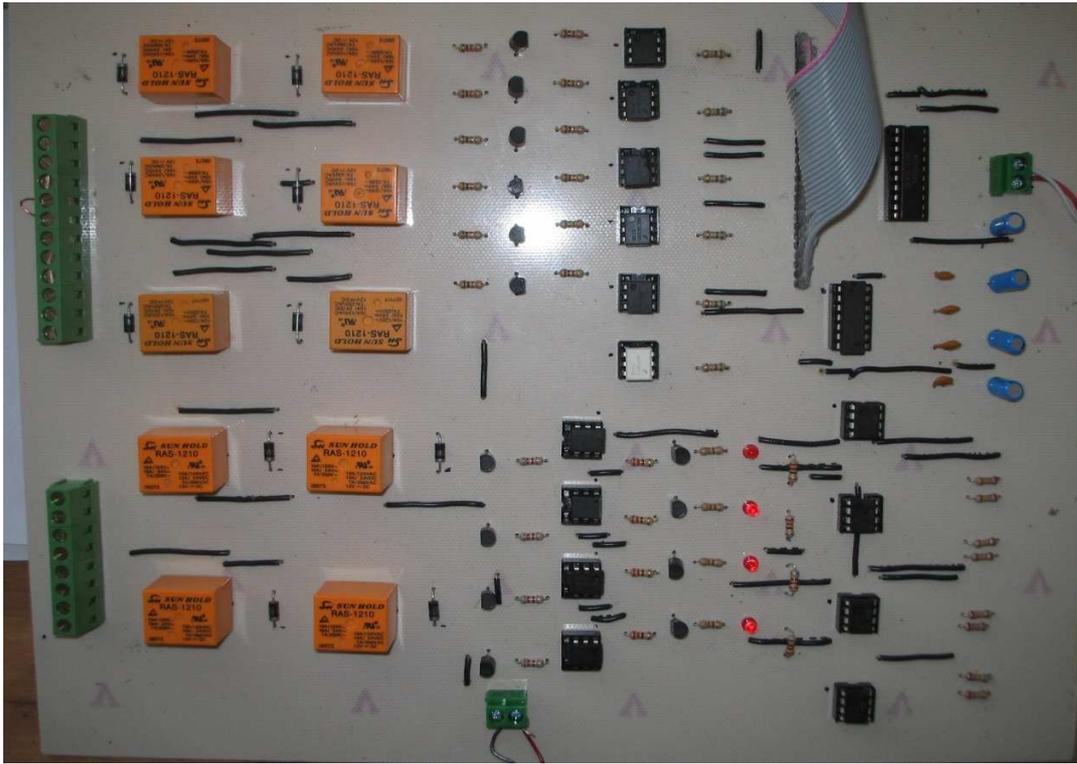


Figura D5 Distribución de componentes en la placa.