

**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería Electrónica**



**Diseño De Prototipo Concentrador De Datos Modbus Tcp/Ip**

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en  
Electrónica con el grado académico de Licenciatura/Bachillerato**

**Jeffrey Eduardo Faerron Ugarte**

**Cartago, Mayo del 2013**

**INSTITUTO TECNOLÓGICO DE COSTA RICA  
ESCUELA DE INGENIERIA ELECTRÓNICA**

**PROYECTO DE GRADUACIÓN**

**TRIBUNAL EVALUADOR**

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. Marvin Hernández Cisneros

Profesor asesor



Ing. Juan Carlos Jiménez Robles

Profesor lector

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, junio 25, 2013



Por medio de la presente, yo, Jeffrey Faerron Ugarte declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 31 de Mayo del 2013



---

Jeffry Eduardo Faerron Ugarte

Céd: 5-0324-0085

## Resumen

Últimamente en cualquier industria dedicada a la producción de materia prima, electrónicos, alimentos, juguetes, objetos de vidrio, automóviles, etc. Suelen existir una cantidad considerable de máquinas o controladores programables que hacen ciertos trabajos en forma automatizada, o sea sin la intervención humana, además debido a diversos factores como evolución de tecnología, diversidad de conocimiento o economía, dentro de la misma industria muchas maquinas suelen ser diferentes aunque realicen el mismo objetivo.

Esta situación presenta dificultades para interconectar maquinas que fueron hechas para comunicarse con sus similares solamente, es aquí donde se presenta el problema por restricciones de comunicación.

La meta de esta solución es lograr interconectar una cantidad específica de máquinas que deben ser controladas mediante un dispositivo principal utilizando algún protocolo que sea estándar entre ellos para así concertar todas las maquinas terminales en un solo dispositivo al que un procesador central pueda acceder.

Se implementara esta solución utilizando conexiones estandarizadas tipo Ethernet, que poseen protocolos de aplicación Modbus TCP/IP para tener la capacidad de enfocar todo el tráfico de comunicación.

Palabras Clave: PLC, HC900, Ethernet, Control, PID, RED, Switch, programas, retraso, trama, dispositivo, controlador PC.

## **Abstract.**

Lately in any industry engaged in the production of raw materials, electronics, food, toys, glassware, cars, etc. There have usually a considerable amount of machines or programmable controllers doing some processes automatically, in other words without human intervention, also due to various factors such as technology updates, economic problems or knowledge diversity, those industries have different styles and brands machines performing the same goal.

This situation presents difficulties for interconnect machines that were made to communicate with their peers only; here is where the problem appears: communications restrictions.

The goal of this solution is interconnect a specified number of machines to be controlled by a master device using some international standard protocol.

The solution is implemented using standard Ethernet connections, which have Modbus TCP / IP like application data protocol to focus all communication traffic into one device.

Keywords: PLC, HC900, Ethernet, Control, PID, RED, Switch, programs, delay, pattern, device, PC controller.

## **Dedicatoria**

*A mis tres padres, el que está en los cielos y a las dos grandes personas que me han brindado el mayor cuidado, apoyo y cariño que puedo pedir en la vida, les dedico este esfuerzo, que es tan de ellos como mío.*

## **Agradecimiento**

Le agradezco a mis hermanas Dietrich, Ana Gabriela, a mi madrina Patricia y a mi tía-mamá Amelia por apoyarme tanto en este trayecto y estar presente siempre con su incondicionalidad y motivación.

A los ingenieros Rodolfo Piedra y Eddy Bustamante los cuales fueron un apoyo muy importante en la realización de este proyecto.

## INDICE GENERAL

<b>Capítulo 1: Introducción .....</b>	<b>12</b>
<b>1.1 Entorno del proyecto .....</b>	<b>12</b>
<b>1.2 Problema existente e importancia de su solución.....</b>	<b>14</b>
<b>1.3 Solución seleccionada.....</b>	<b>16</b>
<b>Capítulo 2: Meta y objetivos .....</b>	<b>19</b>
<b>2.1 Meta.....</b>	<b>19</b>
<b>2.2 Objetivo general .....</b>	<b>19</b>
<b>2.3 Objetivos específicos .....</b>	<b>19</b>
a. <b>Objetivos de hardware.....</b>	<b>20</b>
b. <b>Objetivos de software.....</b>	<b>21</b>
c. <b>Objetivos de documentación .....</b>	<b>21</b>
d. <b>Objetivos de implementación.....</b>	<b>21</b>
<b>Capítulo 3: Marco teórico .....</b>	<b>22</b>
<b>3.1 Descripción del sistema o proceso a mejorar .....</b>	<b>22</b>
<b>3.2 Antecedentes Bibliográficos .....</b>	<b>22</b>
<b>3.3 Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema.....</b>	<b>22</b>
3.3.1 <b>Modbus .....</b>	<b>22</b>
3.3.2 <b>Ethernet .....</b>	<b>25</b>
3.3.3 <b>Modelo OSI .....</b>	<b>25</b>
3.3.4 <b>Protocolo TCP / IP .....</b>	<b>27</b>
3.3.5 <b>Dirección IP .....</b>	<b>27</b>
3.3.6 <b>Mascara de Red .....</b>	<b>28</b>
3.3.7 <b>Modbus TCP/IP.....</b>	<b>29</b>
<b>CAPITULO 4: Procedimiento Metodológico .....</b>	<b>33</b>
<b>4.1 Reconocimiento y definición del problema.....</b>	<b>33</b>
<b>4.2 Obtención y análisis de información .....</b>	<b>34</b>
<b>4.3 Evaluación de las alternativas y síntesis de una solución. ....</b>	<b>36</b>

4.4	Implementación de la solución .....	36
4.5	Reevaluación y rediseño .....	41
<b>CAPITULO 5: Descripción detallada de la solución (Explicación del diseño)</b>		<b>42</b>
5.1	Análisis de soluciones y selección final .....	42
5.1.1	Solución A. ....	42
5.1.2	Solución B. ....	43
5.1.3	Solución C. ....	45
5.2	Descripción del hardware.....	51
5.2.1	Kit de desarrollo.....	51
5.2.2	Controlador HC900 .....	56
5.3	Descripción del software.....	57
5.3.1	Modulo de recepción maestro.....	60
5.3.2	Módulo de Redireccionamiento .....	61
5.3.3	Módulo de consulta esclavo.....	63
5.3.4	Entorno de programación del controlador hibrido HC900.....	64
<b>Capítulo 6: Análisis de Resultados.....</b>		<b>66</b>
<b>Capítulo 7: Conclusiones y recomendaciones .....</b>		<b>77</b>
7.1	Conclusiones. ....	77
7.2	Recomendaciones .....	79

## INDICE DE FIGURAS

<b>Figura 1.1</b>	Controlador industrial HC900 .....	<b>15</b>
<b>Figura 1.2</b>	Esquema de comunicación general del dispositivo concentrador de datos .....	<b>18</b>
<b>Figura 3.1</b>	Estructura de datos de aplicación (ADU).....	<b>24</b>
<b>Figura 3.2</b>	Dirección IP representada en forma decimal.....	<b>28</b>
<b>Figura 3.3</b>	Trama de datos Modbus dentro de trama TCP/IP .....	<b>31</b>
<b>Figura 4.1</b>	Diagrama de Gantt del proyecto.....	<b>41</b>
<b>Figura 5.1</b>	Módulo "C" Programable con aplicaciones Ethernet para interfaz de CompactLogix. ....	<b>43</b>
<b>Figura 5.10</b>	ChipKIT Network Shield .....	<b>54</b>
<b>Figura 5.11</b>	Diagrama de bloque de integrado LAN8720 .....	<b>55</b>
<b>Figura 5.12</b>	Dos sistemas redundantes con HC900 consultados por una PC y 2 HMI .....	<b>57</b>
<b>Figura 5.13</b>	Diagrama de bloques general de la estructura de software para el MDC.....	<b>58</b>
<b>Figura 5.14</b>	Diagrama de bloques del sistema de programación completo .....	<b>60</b>
<b>Figura 5.15</b>	Entorno de programación gráfico para el HC900 .....	<b>65</b>
<b>Figura 5.2</b>	Esquema general de solución A mediante módulo MVI69-ADMNET de Prosoft Technology .....	<b>44</b>
<b>Figura 5.3</b>	Esquema general de solución B que interconecta Modbus RTU con Modbus TCP/IP.....	<b>45</b>
<b>Figura 5.4</b>	Esquema general de solución C que interconecta dispositivos Modbus TCP/IP.....	<b>47</b>
<b>Figura 5.5</b>	Diagrama interno general de funciones que realiza el MDC.....	<b>48</b>
<b>Figura 5.6</b>	Transmisión de una trama Modbus TCP/IP para una solicitud a través del MDC.....	<b>49</b>
<b>Figura 5.7</b>	Transmisión de una trama Modbus TCP/IP para una respuesta a través del MDC .....	<b>50</b>
<b>Figura 5.8</b>	Esquema general de la solución final modificada .....	<b>51</b>
<b>Figura 5.9</b>	ChipKIT MAX32 .....	<b>53</b>
<b>Figura 6.1</b>	Ventana de visualización de dirección IP en el controlador híbrido programable #1 .....	<b>67</b>
<b>Figura 6.2</b>	Conexión de 2 controladores conmutados en una subred 10.1.0.X68	
<b>Figura 6.3</b>	Ventana de visualización de parámetros de red y formato Modbus configurado en los controladores HC900 .....	<b>70</b>
<b>Figura 6.4</b>	Configuración de bloques para la manipulación de registros dentro del controlador híbrido programable HC900 .....	<b>71</b>
<b>Figura 6.5</b>	Configuración de mapeo Modbus dentro del controlador híbrido programable HC900.....	<b>72</b>
<b>Figura 6.6</b>	Resultados de prueba realizada al MDC mediante un programa de simulación Modbus maestro y esclavo .....	<b>74</b>
<b>Figura 6.7</b>	Concentrador de datos Modbus TCP/IP conectado mediante Ethernet .....	<b>75</b>

**Figura 6.8** Diagrama de bloques del proceso de aceptación de escritura en una memoria EEPROM 24LC256. .... **76**

**INDICE DE TABLAS**

<b>Tabla 3.1</b> Códigos de excepción Modbus .....	25
<b>Tabla 3.2.</b> El tradicional modelo OSI, junto con el TCP/ IP estándar de 5 capas .....	27
<b>Tabla 3.3.</b> Código de referencia general para peticiones Modbus .....	33
<b>Tabla 4.1.</b> Presupuesto del proyecto a realizar .....	36

## Capítulo 1: Introducción

### 1.1 Entorno del proyecto

Los procesos de automatización industrial que se utilizan actualmente, ya sea en medianas o grandes empresas van generalmente de la mano con el control automatizado de estos mismos procesos. La empresa Control y Automatización S.A situada en la provincia de Cartago provee a industrias costarricenses y centroamericanas con la venta, instalación y puesta en marcha de equipos personalizados que ayudan a dichas industrias a modernizar y mejorar sus procesos de manufactura, ensamblaje o procesamiento de materia prima para ser convertida en producto final.

Uno de los sistemas más requeridos por las industrias y por consecuencia más solicitados para la empresa Control y Automatización S.A. es la automatización de terminales como silos de almacenamiento, cintas transportadoras, generadores, motores eléctricos, etc. Esto requiere de un lazo cerrado de acción y reacción que funcione sin intervención humana para el mantenimiento de un valor deseado dentro de una cantidad o condición , midiendo el valor existente , comparándolo con el valor deseado , y utilizando la diferencia para proceder a reducirla. [1]

Actualmente en cada industria se maneja una gran cantidad de componentes que deben ser controlados y esto requiere de dispositivos que obtengan las variables a ser procesadas constantemente y en tiempo real, a la vez se necesita que el sistema tome esta variable y la compare con el valor deseado. Así que utilizar un sistema de control para cada proceso no es rentable, es mucho mejor hacer que un solo sistema controle varios dispositivos a la vez.

Este sistema entonces lo podemos separar en dos etapas:

1. La adquisición en tiempo real de las variables de cada dispositivo.

2. La lógica y hardware de control que modifica la variable hasta el nivel deseado.

Actualmente hay dispositivos que realizan una conexión entre los dispositivos y el control, estos ayudan a disminuir la carga del proceso de control en específico, proporcionando un intermediario que se encarga de solicitar variables y enviarlas al dispositivo central que generalmente es un controlador lógico programable (PLC por sus siglas en ingles). [4]

Estos módulos mencionados son muy caros y se encuentran confeccionados para controlar un máximo de 32 terminales. Además que funcionan mediante la conexión Modbus RTU que se usa en comunicación serie. [2]

La empresa Control y Automatización S.A pretende construir un dispositivo que realice la interconexión de estos controles y terminales al igual que los descritos anteriormente para reducir así la carga de procesamiento del dispositivo de control central o Master en nuestro caso.

Se pretende utilizar un protocolo de uso común, estandarizado, de alta densidad de nodos por red y simplicidad de uso. Así que decidieron utilizar la versión de Modbus TCP-IP, este protocolo brinda la simplicidad de uso del protocolo Modbus RTU dentro del estandarizado TCP-IP (Ethernet) el cual abre las puertas a una gran cantidad de aplicaciones e implementación. [3]

El proyecto de investigación y desarrollo se llevará a cabo en las instalaciones de la empresa "Control y Automatización S.A" en la provincia de Cartago, y el diseño del dispositivo final al que de ahora en adelante llamaremos Concentrador de Datos Modbus (CDM), pretende ampliar la gama de productos que utiliza la empresa para la implementación de los sistemas de control en las industrias modernas.

## 1.2 Problema existente e importancia de su solución

Durante los últimos años las industrias que se dedican a la construcción masiva de productos, recipientes, materia prima, etc. Se han enfocado en automatizar procesos de producción específicos para aplicaciones diversas, ya sea desde el transporte de recipientes en bandas transportadoras, formación de tapas, moldeo de diseños específicos, procesado de caña o muchas otras más aplicaciones necesarias para completar un ciclo de producción.

Las maquinas que se encargan de hacer estas labores deben ser controladas para hacer las cosas en el tiempo, cantidad, tamaño adecuado, etc., esta tarea la cumplen los dispositivos de Control de Lógica Programable (PLC) los cuales se encargan de manipular las maquinas terminales automáticamente.

Un controlador de lógica programable es un equipo eléctrico programado en un lenguaje visual (diagrama de escalera) diseñado para controlar en tiempo real procesos secuenciales en base a información recibida por medio de sus entradas, la cual es procesada por un programa lógico interno diseñado según la aplicación y que actúa sobre las salidas o accionadores de otros sistemas que se quieren controlar, todo esto en un ambiente de tipo industrial. [7]



**Figura 1.1.** Controlador industrial HC900 [16]

La empresa CYASA actualmente distribuye equipos y realiza la puesta en marcha de estos sistemas en empresas de automatización industrial en general, ya sean dispositivos solicitados directamente por el cliente o asesoría de proyectos nuevos a la medida e instalaciones completas.

Dentro de las solicitudes se presentan varias configuraciones en donde es necesario plantear una comunicación entre el PLC y diferentes dispositivos terminales por lo cual se crea la necesidad de desarrollar el mencionado Concentrador de Datos Modbus que pretende dar solución al problema de intercomunicación de datos entre el PLC (Master) y muchos terminales (Esclavos).

El problema radica en que actualmente existen varias marcas y modelos de esclavos en una misma industria que deben ser consultados y controlados mediante un solo master, así entonces el dispositivo encargado de la dinámica de control necesita mucha cantidad de procesamiento para consultar valores, procesarlos y según estos, seguir la lógica de control planteada en su programación. Esto consume recursos de procesamiento, espacio en memoria estática y memoria dinámica los cuales son parámetros muy importante en el desempeño de sistemas de control ya que se requieren tiempos de acceso muy cortos y velocidad de procesamiento muy rápida, lo cual es directamente proporcional al hardware disponible como las memorias y a la eficiencia en el programa de control.

Además uno de los problemas que adicionalmente se pretende solucionar en un futuro es referente a la comunicación; Para lograr interconectar dispositivos de distintas marcas, estilos, diseños y tecnología mediante un protocolo universal y ampliamente estandarizado.

Se presenta ahora un reto para solucionar estos problemas y se plantean también posibles caminos a tomar en la solución del caso. La importancia de construir un sistema capaz de resolver lo mencionado anteriormente radica en que se hace una contribución

grande en el asunto de automatización de procesos, reducción de equipos y precios para las industrias, y especialmente en la empresa CYASA con la adición de un electrónico que puedan utilizar en el diseño de nuevos proyectos de automatizado.

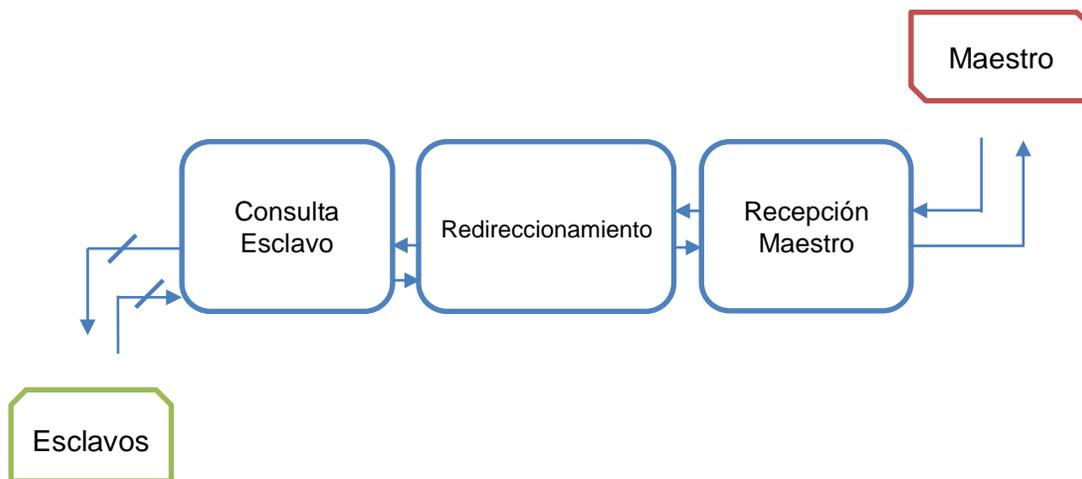
El desarrollo de un dispositivo como este que pretende concentrar destinos o direcciones red de terminales esclavos en un mismo punto y provee a la empresa una herramienta muy interesante ya que hablando de transmisión de datos existen varios formatos que serán admitidos y procesados por el MDC para así tener un punto de comunicación eficiente y practico a la vez, dando así la oportunidad de crear diseños de automatización más abiertos y compactos y complicados a la vez.

### **1.3 Solución seleccionada**

Teniendo en cuenta los siguientes requisitos, las cuales son fundamentales en los procesos de automatización he seleccionado una solución que cumple estos puntos con una eficiencia apropiada.

- Bajo tiempo de muestreo por petición.
- Estabilidad en el ciclo de trabajo.
- Eficiencia en el código de programación.
- Fácil configuración de direcciones.
- Economía de hardware.

Estos requisitos dieron lugar al diseño de un sistema modular tanto en *hardware* como en *software* que suple las necesidades mediante la separación de tareas y procesamiento, el esquema general se presenta a continuación:



**Figura 1.2.** Esquema de comunicación general del dispositivo concentrador de datos

El dispositivo maestro es el que envía las peticiones a los esclavos y procesa las respuestas; Los dispositivos esclavos son servidores Modbus TCP que esperan peticiones para enviar valores requeridos. Después de tener el emisor y el receptor definidos se planteó un diseño en el cual la recepción, redirección y envío de información se realice por partes diferentes en un dispositivo conjunto.

El Concentrador de Datos Modbus TCP-IP está conformado por tres módulos de *software* y dos módulos *hardware*. Dentro del software está el “Receptor Maestro”, el cual se encarga de recibir la trama de solicitud; Luego esta solicitud pasa por el módulo de “Redireccionamiento” donde es modificada para que el ultimo modulo “Consulta Esclavo” la envíe hacia el dispositivo terminal específico, espere su respuesta y la devuelva en dirección contraria (Consulta Esclavo -> Redireccionamiento -> Receptor Maestro).

En el caso del hardware los módulos son: “Procesador de datos” y “placa de red”, ambos nombres describen directamente su funcionamiento modular. Estos cinco diseños se analizarán detalladamente en el capítulo 5.

La escogencia del protocolo Modbus TCP-IP brinda un enlace muy útil y completo entre los sistemas de automatización industriales que utilizan registros accedidos mediante Modbus y la estandarización de comunicación por medio de Ethernet con el cual se tiene un amplio panorama de aplicación a redes grandes y de proporciones variables para ser modificadas o sustituidas en proyectos futuros.

## **Capítulo 2: Meta y objetivos**

### **2.1 Meta**

- Construir un dispositivo electrónico capaz de conectarse a un dispositivo máster y realizar peticiones de información por él a varias terminales, mediante el uso de protocolo Modbus TCP/IP.

*Indicador:* Funcionamiento de la red Ethernet entre el prototipo, el dispositivo Master y los terminales, 100% de peticiones exitosas desde el solicitante (master) hasta los solicitados (terminales).

### **2.2 Objetivo general**

- Diseñar un dispositivo concentrador de datos Modbus TCP/IP que trabaje en serie con un dispositivo master y 128 esclavos para alivianar la carga de procesamiento en sistemas de control automatizados industriales modernos.

*Indicador:* Verificar el funcionamiento del dispositivo mediante pruebas con peticiones Ethernet, que produzcan un porcentaje de error menor al 5%.

### **2.3 Objetivos específicos**

1. Construir un prototipo electrónico capaz de conectarse a una red Ethernet e intercambiar información entre sus conexiones.

Indicador: Que haya comunicación exitosa y correcta entre un terminal conectado en el puerto de entrada y otro terminal conectado al puerto salida del MDC.

2. Implementar un protocolo MODBUS TCP/IP en el MDC para ser utilizado mediante la conexión Ethernet.

Indicador: intercambio de información correcta utilizando tramas Modbus TCP/IP entre el dispositivo Master y los dispositivos Esclavos con un máximo de 25% de paquetes perdidos.

3. Implementar un sistema digital el cual permita al MDC recibir peticiones de variables específicas en tiempo real desde un dispositivo Master hacia varios Esclavos y devolver datos requeridos mediante MODBUS TCP/IP.

Indicador: Que las variables solicitadas sean devueltas correctamente, en tiempo real y desde el destino indicado.

4. Verificar que una consulta completa hacia los 128 dispositivos esclavos se realice en menos de 4 segundos.

Indicador: Que 128 solicitudes sean procesadas correctamente en un tiempo menor a 3 segundos.

#### **a. Objetivos de hardware**

1. Seleccionar dispositivos adecuados basándose en requerimientos mínimos de diseño propuestos y características de software.
2. Reducir lo mayor posible el tamaño y los costos de construcción.

3. Interconectar hardware y crear un diseño compacto que soporte las condiciones industriales y de uso continuo.

#### **b. Objetivos de software**

1. Implementar un módulo llamado “Receptor Maestro” que cumpla la función de esclavo Modbus TCP-IP dentro del concentrador de datos.
2. Crear módulo de “Redireccionamiento” capaz de modificar la trama de acuerdo a la información guardada en memoria.
3. Implementar modulo “Consulta Esclavo” encargado de funcionar como un servidor Modbus TCP-IP dentro del concentrador de datos.
4. Diseñar una página web para que funcione como entrada de datos en el servidor web del MDC.

#### **c. Objetivos de documentación**

1. Crear un esquema electrónico del dispositivo.
2. Documentar el modo de uso y características principales en un manual de usuario.

#### **d. Objetivos de implementación**

3. Seleccionar un armazón para montar y proteger el equipo ante las condiciones diarias de un ambiente industrial.

## **Capítulo 3: Marco teórico**

### **3.1 Descripción del sistema o proceso a mejorar**

Se debe incluir y explicar en esta sección el diagrama de flujo que muestre la secuencia de etapas del proceso.

Además, si procede, es conveniente presentar una descripción del sistema en el cual se inserta el proyecto a implementar

### **3.2 Antecedentes Bibliográficos**

El objetivo de este dispositivo es lograr la concentración de datos solicitados a los dispositivos esclavos en una sola dirección Ethernet, esto se logra mediante el encapsulado de la trama con el protocolo MODBUS TCP/IP, este protocolo define la forma y el significado de los datos enviados mediante un orden de bytes dentro de la trama.

Modbus es un protocolo de mensajes desarrollado a finales de la década de los 70's por la empresa Modicon para la aplicación en sistemas de automatización industrial y controladores programables propios de esa misma empresa. Este se ha convertido en un estándar en la industria del control y monitorización de dispositivos industriales mediante la transferencia de información Discreta y Analógica.

### **3.3 Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema.**

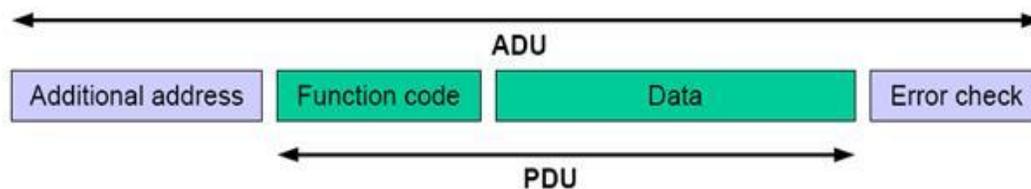
#### **3.3.1 Modbus**

Los dispositivos Modbus se comunican usando la técnica de Maestro-Eslavo (cliente-servidor) en donde solo uno de ellos puede iniciar una transacción (Master) y los demás (Esclavos) responden con los datos requeridos o acciones especificadas.

Un esclavo puede ser un transductor, un motor, dispositivo de medición, válvula o cualquier otro que procese la información y la envíe como una salida hacia el Maestro. Esclavos no pueden iniciar mensajes por ellos mismos, solo responden a solicitudes de un Maestro.

Un dispositivo maestro puede acceder a esclavos individuales o enviar un mensaje a todos los esclavos, pero estos últimos responderán solo a este master.

El mensaje de acceso de un maestro (Application Data Unit, ADU por sus siglas en inglés) consiste en una dirección hacia un determinado esclavo o hacia todos ("Broadcast"), un código de función, los datos requeridos, y un dato para la revisión de errores.



**Figura 3.1.** Estructura de datos de aplicación (ADU) [9]

Dentro de la estructura ADU se define otra estructura llamada unidad de datos de protocolo (Protocol Data Unit, PDU) por sus siglas en inglés y se refiere al código de función y datos incluidos en la trama completa. Así entonces tanto en la trama de solicitud como en la respuesta van incluidos los campos de dirección, código de función, datos y error de datos, Si no ocurren errores, el esclavo retorna los datos como fueron solicitados, en cambio si algún tipo de error ocurre retorna un mensaje de excepción según la falla (Figura 3.1).

**Tabla 3.1** Códigos de excepción Modbus [9]

<b>Código</b>	<b>Excepción</b>	<b>Descripción</b>
01	Función ilegal	El código de función recibido no es permitido o es inválido.
02	Dirección de datos ilegal	La dirección de datos recibida no es permitida para este esclavo o es inválida.
03	Valor de datos ilegal	Un valor contenido en el campo de datos de consulta no es permitido para el esclavo o no es válido.
04	Fala en dispositivo Maestro/Esclavo	El servidor fallo durante la ejecución. Ocurrió un error irrecuperable mientras que esclavo/servidor intentaba realizar la acción solicitada.
05	Conexión aceptada	El esclavo / servidor ha aceptado la solicitud y está procesando, se requiere algún tiempo para hacerlo. Esta respuesta es devuelta para evitar un error de tiempo de espera.
06	Dispositivo Maestro/Esclavo ocupado	El esclavo se dedica a la ejecución de un comando. El maestro debería transmitir el mensaje más tarde cuando el esclavo este libre.
07	Conexión rechazada	El esclavo no puede realizar la función de programa requerida en la consulta. El maestro debe solicitar información de diagnóstico del esclavo.
08	Error en paridad de memoria	El esclavo intentó leer la memoria extendida, pero detecta un error. El maestro puede volver a intentar la solicitud, pero el servicio puede estar ocupado en el dispositivo esclavo.
0A	Problema con puerta de enlace	Vías de puerta de enlace no disponible.
0B		El dispositivo de destino no respondió (la puerta de enlace genera esta excepción).
FF	Respuesta extendida de excepción	La respuesta de excepción PDU contiene información extendida. Posteriormente, un campo de 2 bytes de longitud indica el tamaño en bytes de la información de la excepción de función de código específico.

Modbus se clasifica como un protocolo de aplicación, lo que hace es definir reglas para organizar e interpretar datos conservando una estructura simple de mensaje, fácil de entender, de código libre y accesible para todos. Para entender mejor porque Modbus se clasifica en capa de aplicación y su estándar abierto, debemos entender el modelo de Interconexión de Sistemas Abiertos (OSI por sus siglas en inglés), El protocolo TCP/IP y el sistema Ethernet.

### **3.3.2 Ethernet**

Es la tecnología de redes de área local (LAN) más usada por computadores en el mundo. Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI. Una red LAN suele utilizar cable coaxial o calidades especiales de cables de par trenzado, los dispositivos están conectados al cable y compiten por el acceso utilizando un CSMA/CD (Acceso Múltiple por Detección de Portadora con Detección de Colisiones).

Ethernet fue nombrado en 1973 por Robert Metcalfe que trabajaba para Xerox PARC (Palo Alto Research Center), por la sustancia pasiva llamado "luminoso éter" que hacía referencia a la teoría de la física hoy ya abandonada según la cual las ondas electromagnéticas viajaban por un fluido invisible que se suponía llenaba todo el espacio. Ethernet fue llamado así para describir la forma en que el cableado, también un medio pasivo, de manera similar podría transportar datos en todas partes a través de la red. [11]

### **3.3.3 Modelo OSI**

El modelo de red OSI fue desarrollado en 1983 como una referencia para el desarrollo de estándares en comunicación de datos, el modelo tradicional está presente en la columna izquierda de la Figura xxxx33 la cual se compone de 7 capas, en las cuales cada una maneja tareas definidas específicamente. Existe otro estándar llamado TCP/IP (Protocolo de Control para Transmisión/Protocolo de Internet), el cual es una simplificación del modelo OSI pero con 5 capas (Capa 5 y 6 suprimidas), Este es el de

interés para definir el funcionamiento de Modbus a través de internet, el cual es de nuestro interés.

**Tabla 3.2.** El tradicional modelo OSI, junto con el TCP/ IP estándar de 5 capas [9]

Modelo OSI (7 Capas)			TCP/IP
7	Aplicación	Usado por las aplicaciones de software para preparar e interpretar los datos para su uso por las seis capas inferiores. Proporciona la aplicación de interfaz a la red.	Capa de aplicación.
6	Presentación	Representación de datos, tipo de codificación, y Define los caracteres utilizados. Realiza negociación de datos y protocolo además de la conversión para garantizar que los datos pueden ser intercambiados entre los dispositivos y puedan ser transportables a través de la red. También realiza de datos compresión y cifrado.	
5	Sesión	Control de llamada y sincronización de sesión de conexión. Responsable del establecimiento y gestión de sesiones / conexiones entre las aplicaciones y la red.	
4	Transporte	Secuenciación de datos de aplicación, controla inicio / final de la transmisión, proporciona detección error detección, corrección, recuperación de extremo a extremo y compensación. Proporciona software de control de flujo de datos entre las redes.	Capa de transporte.
3	Red	Controles de enrutamiento, priorización, configuración, liberación de conexiones, control de flujo. Establece / mantiene conexiones a través de una red y proporciona direccionamiento, enrutamiento y entrega de paquetes a los hosts. IP se encuentra aquí.	Capa de Internet y red.
2	Enlace	Responsable de asegurar una entrega confiable a los niveles más bajos, incluyendo trama de datos, detección y corrección de errores, control de secuencia y flujo. Ethernet (IEEE 802.2) y MAC se definen en este nivel.	Capa Host a red o de acceso a la red.
1	Física	Define los atributos eléctricos, mecánicos, funcionales y de procedimiento utilizados para acceder y enviar trama de datos binarios sobre los medios físicos (define el conector RJ-45 y cable CAT5 de Ethernet).	

### 3.3.4 Protocolo TCP / IP

TCP/IP fue introducido en 1974 para la World Wide Web, y está definido en la capa de transporte y red del internet que habitualmente enlaza las instalaciones Ethernet de todo el mundo, esto permite a los bloques de datos binarios ser intercambiados.

TCP tiene la función principal de asegurarse que los paquetes de datos sean recibidos correctamente, mientras que IP permite a los bloques ser intercambiados entre las computadoras mediante el correcto direccionamiento y enrutamiento.

Entonces para que 2 dispositivos seas interoperables en la misma red, ellos deben tener el mismo protocolo de capa aplicación. En el pasado cada dispositivo utilizaba su propio protocolo lo cual hacía imposible la comunicación entre ellos, es por esto que fue creado el Modbus TCP/IP.

### 3.3.5 Dirección IP

Una dirección IP es un dato de 32 bits comúnmente expresado por cuatro octetos (8bits) separados por un punto, esto quiere decir que podemos expresar un número de:

$$2^8 = 256$$

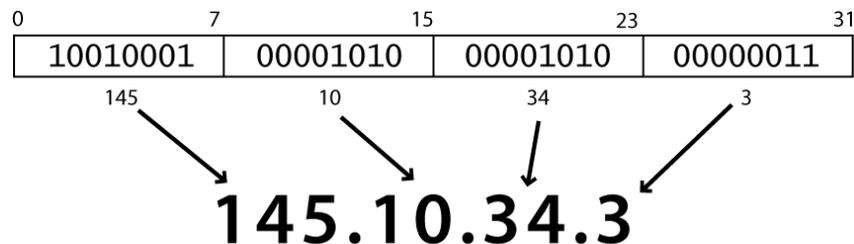


Figura 3.2. Dirección IP representada en forma decimal [12]

Como máximo por cada octeto, expresándolo en forma decimal tenemos un rango de 0-255 por octeto, con los cuatro entonces podemos representar un total de 4.3 billones de combinaciones. Esto es sumamente necesario, aunque parezca un número muy grande, ya que esta dirección es asignada a cada computadora y es única para cada dispositivo dentro de la misma red.

Las direcciones IP se obtienen mediante bloques y últimamente estos bloques los distribuye la Autoridad de Asignación de Números en Internet (IANA por sus siglas en ingles), las grandes compañías compran bloques muy largos de direcciones IP y las dividen entre su organización, a esto se les llama subredes las cuales son cadenas continuas de direcciones en donde la primera dirección está reservada para identificar la subred o el servidor igualmente; Mientras que la última dirección es utilizada cuando se envían mensajes "Broadcast" esto significa que va dirigido a todas las direcciones de la subred.

Una dirección IP está compuesta por dos partes, la primera se refiere a la dirección de red (Net ID) y la segunda a una maquina específica (Host ID) dentro de esa respectiva red.

Esta escalabilidad que proveen las direcciones IP es fundamental para el desarrollo del proyecto, ya que lo se busca es la concentración de datos en una sola dirección y la organización de estas direcciones nos abren un abanico de posibilidades para el diseño.

### **3.3.6 Mascara de Red**

Al igual que la dirección IP, la máscara de subred se compone de 32 bits, los cuales se separa en 4 octetos y su utiliza para determinar a qué subred pertenece determinado grupo de IP's, esto se realiza mediante una operación AND entre la máscara de subred y el IP en cuestión, el resultado presenta la subred y lo que sobra es la dirección del nodo.

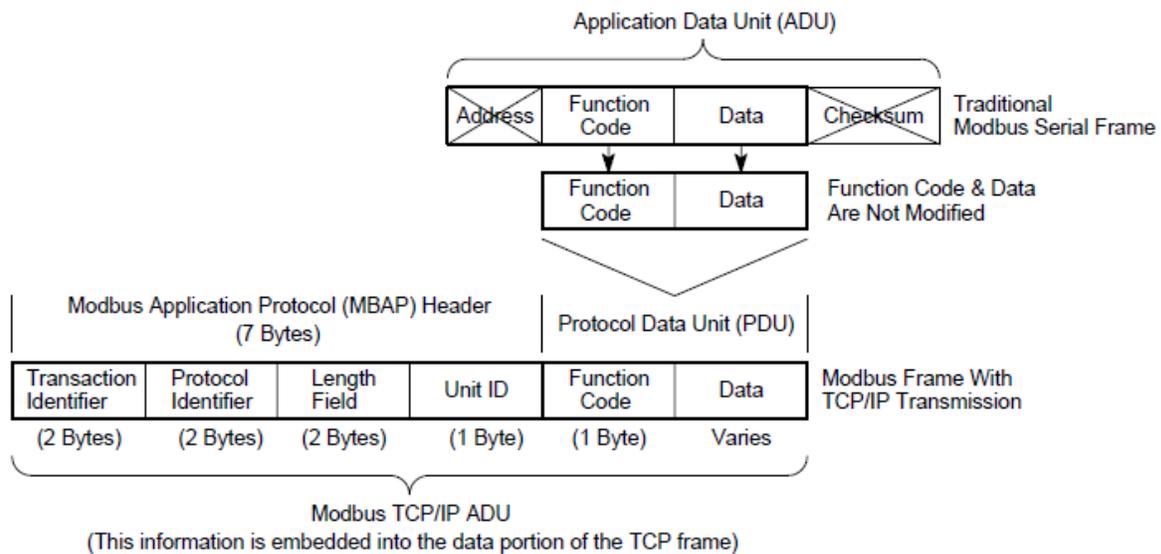
Esta característica de la máscara de subred amplia y subdivide la red en grupos de IP's o subredes de hasta 254 terminales, lo que plantea un gran interés de este diseño en el direccionamiento de nodos Modbus a través de Ethernet, ya que es posible conectar con dispositivos presentes en la subred que se especifique.

Y con 254 nodos por subred, lo que cubre con el objetivo de este diseño.

### **3.3.7 Modbus TCP/IP**

En resumen, Modbus TCP/IP utiliza el protocolo TCP/IP y Ethernet para transportar los datos de la estructura del mensaje Modbus entre dispositivos compatibles. Es decir, Modbus TCP/IP combina una red física (Ethernet), con un estándar de red (TCP/IP), y un método estándar de representación de datos (Modbus como el protocolo de aplicación). En esencia, el mensaje Modbus TCP/IP es simplemente una comunicación Modbus encapsulada sobre una red Ethernet TCP/IP. [9]

En la práctica, Modbus TCP incrusta una trama de datos Modbus en una trama TCP, sin la suma de control Modbus, como se muestra en el siguiente diagrama.



**Figura 3.3.** Trama de datos Modbus dentro de trama TCP/IP [9]

El campo de verificación de error es eliminado y a cambio de eso se utiliza el método estándar de verificación de errores de Ethernet TCP/IP de la capa de enlace además se sustituye el campo de dirección del ADU con un identificador de unidad (Unit ID) el cual forma parte de la cabecera del protocolo de aplicación Modbus (MBAP por sus siglas en ingles).

El MBAP esta conformada por los siguientes campos de bytes:

- **Identificador de transacción (2 Bytes):** Este campo se utiliza para la identificación de la transacción de sincronización.
- **Identificador de protocolo (2 bytes):** Este campo siempre es 0 para los servicios Modbus, otros valores están reservados para futuras ampliaciones.

- **Tamaño (2 bytes):** Es un contador de bytes de los campos restantes e incluye el byte identificador de la unidad, el código de función y el campo de datos.

- **Identificador de unidad (1 byte):** Este campo se utiliza para identificar un servidor remoto que no se encuentra en la red TCP/IP. Esto es con un caso de conexión serie el cual no se incluye en nuestro hardware.

La trama completa Modbus se encapsula en el campo de datos de la trama TCP y se envía hacia el puerto 502, el cual está especialmente reservado para aplicaciones Modbus en las cuales se reciben y envían datos.

El protocolo TCP/IP provee los recursos para que 2 terminales puedan comunicarse sobre una red Ethernet, pero esto no garantiza la correcta comunicación entre ellos, podemos hacer una analogía con las líneas telefónicas que representan las redes Ethernet y el lenguaje, que si dos personas no hablan el mismo lenguaje y aunque puedan oírse no podrán entenderse. Para esto es el protocolo Modbus, que trabaja en la capa de aplicación y organiza los mensajes para que sean entendidos por ambos terminales.

Modbus opera sobre registros en memoria, los monitorea, configura y escribe; y su modelo se basa en una simple estructura que diferencia 4 tipos de datos los cuales son:

1. Discrete Inputs (Entradas Discretas)
2. Coils (Salidas)
3. Input Registers (Datos de entrada)
4. Holding Registers (Datos de Salida)

Los registros de un dispositivo Modbus están organizados según los datos anteriores y estos datos se manejan identificándolos por un número antecesor en la dirección de referencia.

**Tabla 3.3.** Código de referencia general para peticiones Modbus TCP/IP

Leer/Escribir salidas discretas o Coils (0x)	Se utiliza para conducir los datos de salida a un canal de salida digital.
Leer entradas discretas (1x)	La dirección de referencia dada luego de 1x cambia su estado ON/OFF mediante el canal de entrada digital
Leer registros de entrada (3x)	Un registro de referencia 3x contiene un número de 16 bits recibido de una fuente externa por ejemplo, una señal analógica.
Lectura/escritura de salida o registros de retención. (4x)	Un registro 4x se utiliza para almacenar 16 bits de datos numéricos (binario o decimal), o para enviar los datos de la CPU a un canal de salida.

Así hay otras funciones más que definen la manipulación de los datos en el dispositivo y todo depende del número antecesor, esta característica del Modbus es esencial en la decodificación de la trama a recibir y enviar por parte del concentrador a diseñar y representa uno de los retos más grandes para desarrollar en el proyecto.

## **CAPITULO 4: Procedimiento Metodológico**

### **4.1 Reconocimiento y definición del problema**

El problema se define en base a demanda y experiencia en el campo de aplicación y ventas de la empresa para la cual se realiza el diseño. Control y Automatización S.A tiene alrededor de 10 años ofreciendo los servicios para empresas que necesitan dispositivos de control, mecanismos e instrumentación para controlar los procesos industriales que realizan, esta experiencia impulso a la gerencia para buscar una solución al problema que se presenta al tratar de integrar dispositivos de diferentes marcas en un mismo sistema de control.

Ya que existen diversas marcas proveedoras de instrumentación para el control industrial y en una empresa grande es poco probable que existan solo dispositivos de una sola marca, se presentó la idea de unificar conexiones en un solo dispositivo, independientemente del hardware propietario, por medio de un protocolo estándar es posible hacerlo.

Haciendo un estudio de los sistemas y dispositivos industriales, se deduce que es necesario un protocolo que reúna las características siguientes:

- Confiable
- Redundante
- Con capacidad de expansión.
- Conocido
- Rápido

De las recomendaciones del ingeniero a cargo en la empresa se plantearon posibles protocolos de comunicación para el desarrollo del software, entre ellos estaban Protocolo Serie, USB, Modbus RTU, Modbus TCP, Profibus.

Para evaluar y modificar soluciones fue necesario tomar referencia de documentación y datos teóricos de cada protocolo, así como también de cada dispositivo de hardware posible a utilizar. Esto seleccionaría solo los que cumplieran con requisitos planteados anteriormente.

Luego fue necesario aplicar pruebas de desempeño y simulaciones sobre el hardware para llegar a determinar la mejor solución para el problema. Los principales criterios para discriminación de alternativas fueron el tiempo total por lectura-escritura de envío por trama y el nivel de estandarización en la industria.

Otro problema que definió un punto de discriminación de soluciones fue la capacidad de almacenamiento, este problema se presentó al momento de plantear el objetivo de tiempos máximos de acceso, ya que una posibilidad que se tenía antes para el direccionamiento era pasar información completa de dispositivo a dispositivo, esto consumía mayor tiempo por proceso, así que una solución sería guardar estos datos en un dispositivo local para su uso secuencial.

Guardar datos implica conectar un dispositivo de almacenamiento al sistema y esto entonces presenta un nuevo problema, el de tiempos de acceso al nuevo dispositivo.

## **4.2 Obtención y análisis de información**

Parte de la información que se obtuvo del problema fue presentada por el personal a cargo de la empresa ya que estas personas fueron las primeras en plantear una solución para este problema. Así que las primeras soluciones plantearon incógnitas y nuevos problemas, para las cuales fue necesario obtener información desde otras fuentes.

Un medio de información fue la investigación bibliográfica en la literatura disponible dentro de la empresa, folletos de información de proveedores, catálogos de producto, libros y manuales.

Internet permitió el acceso a información reciente sobre los dispositivos existentes en la actualidad, los que se están desarrollando y los que antes se utilizaban. Gracias a esto, se replantearon objetivos y se mejoraron costos por hardware.

Para evaluar el desempeño con respecto a los tiempos de acceso se compararon especificaciones técnicas de dispositivos presentes en el mercado actual, se establecieron posibles protocolos que cumplieran con velocidades mínimas de comunicación, luego fue necesario realizar un presupuesto para especificar los gastos al desarrollar este diseño.

**Tabla 4.1.** Presupuesto del proyecto a realizar.

Recursos	Cantidad	Costo ¢
Mano de obra	1	
⤴ Estudiante		2 360 000
Servicios Básicos		
⤴ Electricidad	1	8 000
⤴ Agua	1	5 000
⤴ Internet	1	19 000
Componentes		
⤴ Dispositivos Esclavos	3	216 650
⤴ PLC	1	150 000
⤴ Componentes Electrónicos	1	200 000
Equipo		
⤴ PC	1	400 000
⤴ Software para Diseño de Circuitos	1	105 000
⤴ Software Para Programación	1	40 000
⤴ Software de simulación	1	80 000
⤴ Switch y cableado	1	500 000
<b>Total</b>	<b>1</b>	<b>4 203 700</b>

Este presupuesto es un estimado de todos los recursos a utilizar durante un periodo de tiempo específico, incluyendo la mano de obra por 6 meses de un estudiante en

condición de práctica profesional, estos datos fueron requeridos por el gerente general de la empresa para conocer la factibilidad de invertir en el proyecto.

### **4.3 Evaluación de las alternativas y síntesis de una solución.**

Dentro de las alternativas para el diseño de este hardware se encontraron soluciones que implicaban programación de software sobre una PC, ensamblado de componentes electrónicos en un sistema embebido, kit de desarrollo, FPGA y otros.

Se pueden describir ciertos criterios que se siguieron para la selección de alternativas, pero uno de los principales fue la estandarización de todo el hardware y software.

La idea de construir un dispositivo concentrador de datos es llegar a tener un hardware que se pueda conectar con dispositivos de conexiones físicas diferentes por un extremo y por otro extremo se tenga un solo tipo de conexión, en nuestro caso será el Ethernet.

Otra forma de implementar esta idea es interconectar dispositivos diferentes pero con un mismo protocolo y centralizarlos en una sola dirección IP, esto requeriría menos hardware que la idea anterior.

Así que inicialmente se recolectó información para cada una de las posibles soluciones, luego se planteó un esquema general del hardware-software necesario para hacer funcional cada dispositivo final y se hicieron estimaciones de los valores que obtenía cada solución con respecto a los objetivos planteados anteriormente, todo esto mediante datos técnicos.

### **4.4 Implementación de la solución**

Primeramente, se realiza una investigación sobre el protocolo de comunicación a utilizar para la realización del proyecto, además de la búsqueda de estándares en conexiones y tramas industriales a nivel nacional e internacional. Luego se procede con la investigación técnica de los componentes posibles para la construcción o adquisición de un dispositivo que tenga todas las características necesarias para implementar el protocolo de comunicación seleccionado en la sección anterior y además sea capaz de comunicarse con máquinas industriales ya existentes.

Luego de efectuar una selección apropiada de estos electrónicos ó posibles dispositivos en el mercado actual que cumplan con requisitos, se plantea un modelo teórico para cada solución.

Con el modelo teórico y características reunidas se procede a evaluar las características que se listan a continuación:

- ✓ Temperatura de operación
- ✓ Tamaño de componentes
- ✓ Material de construcción
- ✓ Interferencia electromagnética
- ✓ Capacidad de procesamiento
- ✓ Capacidad de mejorar
- ✓ Costos de operación
- ✓ Costos de mantenimiento
- ✓ Disponibilidad en el mercado

Hecha esta evaluación se procede a establecer el esquema de conexión entre los dispositivos, acoplando los módulos necesarios de hardware para la confección del circuito completo. Seguidamente se presenta un lapso en el cual se realizan pruebas de

simulación para los módulos, se programan rutinas de funcionamiento para los medios de comunicación en los módulos, puertos disponibles, ciclos para establecer tiempos de corrida y otras simulaciones necesarias para caracterizar el hardware escogido.

Esta fase marco una pausa en el diseño ya que la propuesta que se tenía hasta este momento debió ser modificada, esto se decidió gracias al análisis de las pruebas simuladas, las cuales revelaron que la opción que los ingenieros tenían pensada no era la mejor y se comprobó prácticamente mediante una nueva propuesta.

El protocolo de comunicación, en este caso Modbus TCP/IP debe ser programado en su forma servidor y en su forma terminal, esto con el objetivo de establecer la comunicación entre dispositivos maestros y esclavos, esta lógica debe ser muy eficiente al momento de manejar las tramas de información, ya que debe alternar estados en cada solicitud que hace el maestro. Por ende la implementación del protocolo debe ser lo más simplificada posible, para así ahorrar tiempo, espacio y capacidad de procesamiento.

Se deben comprar los componentes necesarios, para este paso es necesario estimar unas 2 semanas y media, ya que la disponibilidad del hardware en el mercado no es siempre la misma y podría retrasar todo el proyecto si no es posible conseguir todo lo planteado. Al tener disponible el hardware necesario se procederá a programar todo el código.

Lo siguiente en la línea es realizar pruebas necesarias para probar que el protocolo seleccionado e implementado funcionase correctamente en el hardware diseñado y así poder proceder con el resto del sistema. Esta es una de las etapas más importantes, ya que este acople entre software y hardware será el que determine la eficiencia, velocidad en las peticiones y procesamiento de datos.

Por último, se harán pruebas del sistema en completo, desde el reconocimiento de un dispositivo esclavo hasta la verificación del control automático a través de una red que utilice el nuevo prototipo MDC para interconectarla.

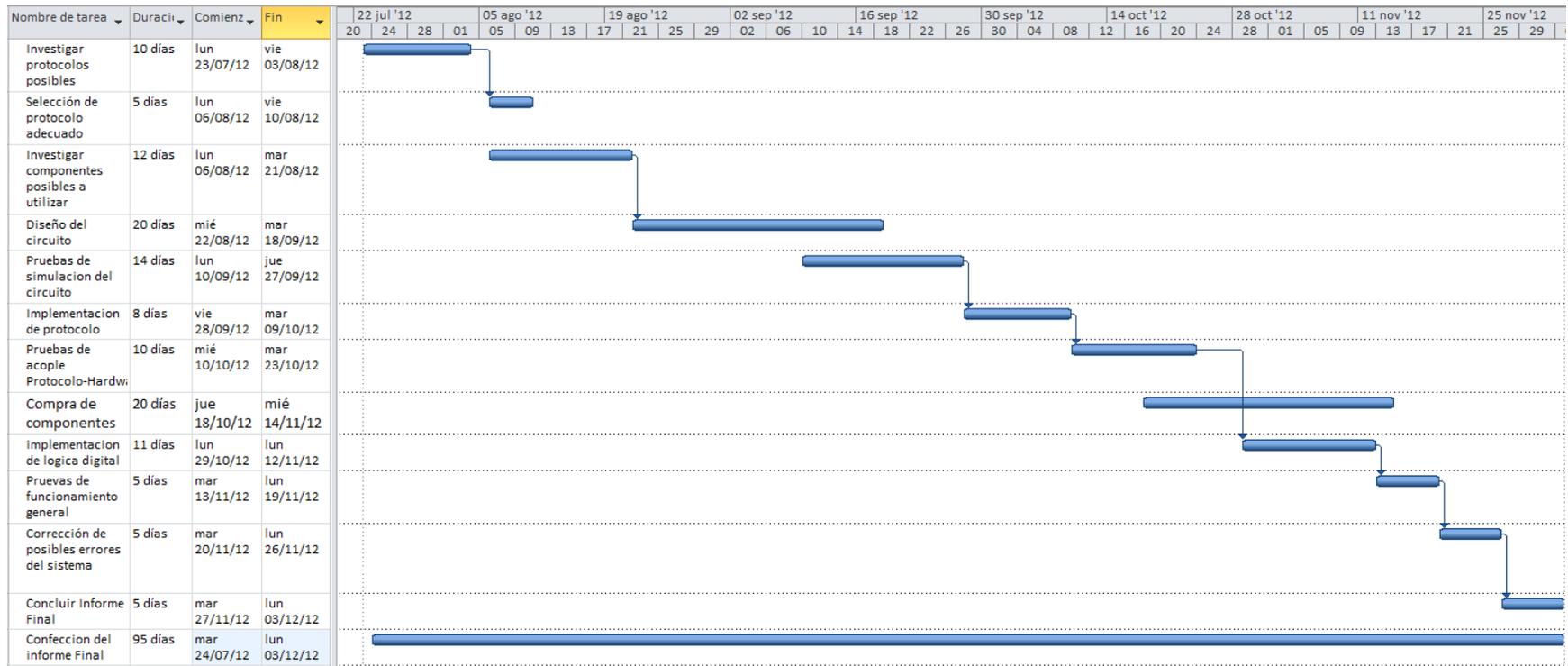


Figura 4.1. Diagrama de Gantt del proyecto.

## **4.5 Reevaluación y rediseño**

Una re-evaluación para el MDC podría realizarse luego de ponerlo en funcionamiento en redes más grandes, esto daría a conocer el verdadero potencial de la idea y revelaría si es necesario aumentar capacidad de procesamiento, conexiones extras o modificación de software.

La construcción física, resistencia a temperaturas, estados de operación forzados y toda clase problemas que se manifiestan en un ambiente industrial son criterios que se podrán evaluar como posibles mejoras al sistema, y la única forma de realizar estas pruebas es implementar el dispositivo en fábricas reales que presentan estas características. Estos serán los criterios a evaluar en futuras mejoras.

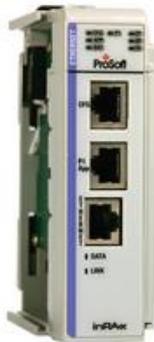
La capacidad de almacenamiento también es una posible mejora para el MDC, ya que los datos que se puedan re-direccionar dependen del número de direcciones de esclavos que se puedan almacenar internamente.

## CAPITULO 5: Descripción detallada de la solución (Explicación del diseño)

### 5.1 Análisis de soluciones y selección final

#### 5.1.1 Solución A.

Una de las primeras soluciones planteadas al inicio fue la idea de comprar un módulo de hardware basado en conexiones Ethernet totalmente programable en lenguaje C++ que soporta ser cliente o servidor web y el cual permite a un procesador comunicarse con otros dispositivos de protocolo Ethernet utilizando un código de software personalizado.



**Figura 5.1.** Módulo "C" Programable con aplicaciones Ethernet para interfaz de CompactLogix. [18]

Este módulo permite programar mediante software C, rutinas de comunicación y protocolos que uno desee en el, esto brinda un puente entre los procesadores Allen-Bradley de la serie CompactLogix que poseen protocolos propietarios con cualquier otro protocolo de comunicación personalizado que pueda ser programado en este módulo.



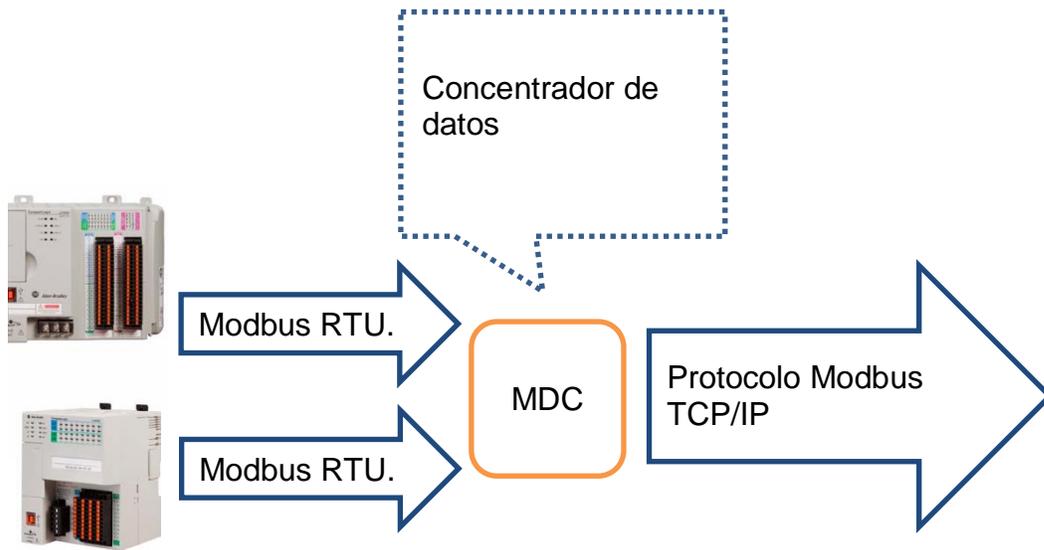
**Figura 5.2.** Esquema general de solución A mediante módulo MVI69-ADMNET de Prosoft Technology.

Esta propuesta se desechó ya que cumplía el objetivo de migrar de un protocolo propietario hacia otro protocolo basado en Ethernet (el cual es un estándar deseado en el diseño del proyecto) pero queda el diseño restringido a controladores de marca Allen-Bradley, además de presentar un alto costo, ya que aunque es un módulo de programación abierto, la interfaz hardware es propietaria de Prosoft y el precio de este queda dependiendo de dicha empresa.

### 5.1.2 Solución B.

La siguiente solución presenta una característica de estandarización más pronunciada, la idea principal es construir un concentrador que sea capaz de comunicar dispositivos por puertos seriales que hablan Modbus RTD con otro dispositivo que se comunique a través de Ethernet por Modbus TCP/IP.

En este caso los dispositivos Modbus RTU se comunican a través de las terminales serie RS-232 y cada uno necesita un puerto independiente para comunicarse.



**Figura 5.3.** Esquema general de solución B que interconecta Modbus RTU con Modbus TCP/IP.

Entonces el MDC debe tener varios puertos RS-232 para conectar los terminales esclavos a ser consultados, debe también poseer la lógica necesaria para transformar el mensaje desde Modbus TCP/IP a Modbus RTU y viceversa en cada transacción de solicitud.

Dentro de las consultas a ingenieros en plantas de proceso industriales, esta idea tiene mucha aceptación y es una de las más puntuadas debido al número de controladores que se manejan en las industrias y que utilizan el protocolo Modbus RTU para comunicarse.

El hecho de necesitar un puerto serie para cada esclavo, implica tener un hardware más grande, más costos de construcción y por ende mayor costo para el usuario final. En estos momentos la versión RTU de Modbus es bastante estándar pero no es tendencia para nuevos PLC's, sistemas distribuidos y demás, uno de los objetivos es apuntar el sistema hacia dispositivos modernos, así que esto disminuye mucho el criterio de estandarización que posee esta solución. Además que en términos de programación, el consumo por comunicarse con varios terminales por medio de puertos RS-232 demanda más capacidad de procesamiento a medida que aumenta cantidad de conexiones. Estas fueron algunas razones por lo que dicha solución no fue implementada.

### **5.1.3 Solución C.**

Otra posible solución se planteó basándose en conexiones Ethernet únicamente, que comuniquen por un extremo del MDC al maestro y en el otro extremo al dispositivo esclavo terminal que será consultado. Ambos dispositivos deben ser comunicados por Modbus TCI/IP. Ahora el problema de tener solamente un puerto para un terminal esclavo lo resuelve la característica de que estos terminales sean Ethernet también, ya que la necesidad de conectar más terminales se soluciona con la inclusión de un conmutador IP en la red local.

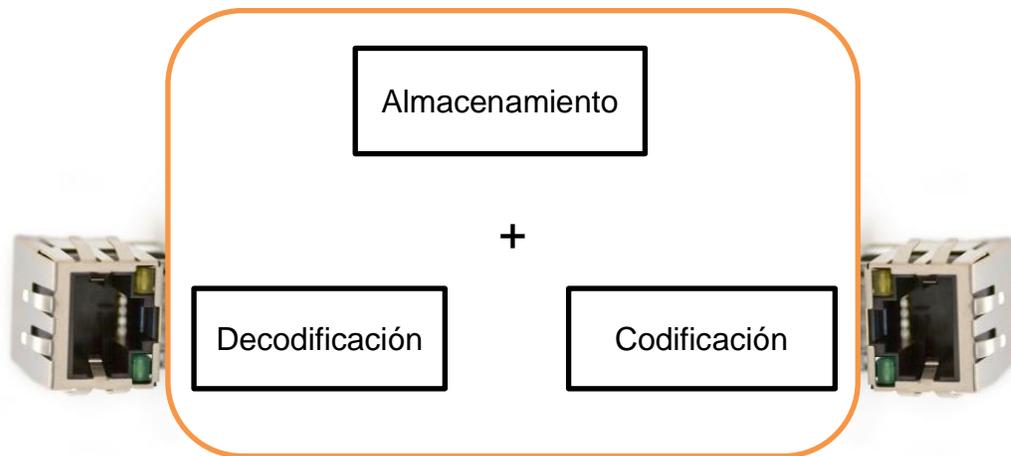


**Figura 5.4.** Esquema general de solución C que interconecta dispositivos Modbus TCP/IP.

La ventaja de conexiones Ethernet es el protocolo TCP/IP con lo cual es posible acceder a cualquier dirección de red que este dentro de la misma subred asignada, estas direcciones representan terminales posibles a conectar y por supuesto a consultar. Esta característica cubre los objetivos de estandarización y además de número de dispositivos posibles para conectar.

Dentro cada subred es posible asignar 255 direcciones IP, lo que representa ya sea PLC's con registros accesibles, Terminales de entradas analógicas u otros. Cualquiera de estos pueden poseer muchos registros que se seleccionan mediante el protocolo Modbus TCP/IP, este se encarga de interpretar las peticiones y ordenar la información enviada.

Dentro de la caja que representa el MDC se pueden presentar un poco más detallados los componentes y conexiones que forman la lógica de funcionamiento del concentrador en sí.



**Figura 5.5.** Diagrama interno general de funciones que realiza el MDC.

Este diagrama es un poco más específico en términos de funcionamiento interno del concentrador de datos, se presentan tres funciones principales del sistema las cuales son necesarias para que el hardware realice el direccionamiento de las tramas que envía el dispositivo maestro a los diferentes esclavos.

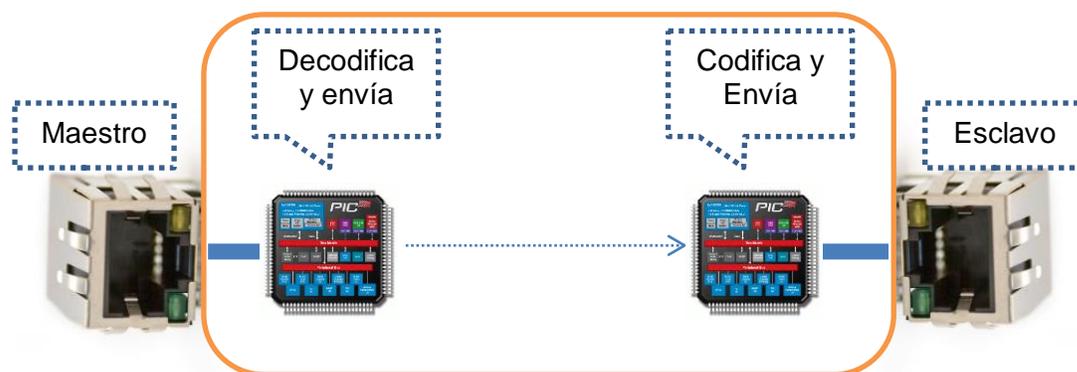
Estas funciones y todas las sub-funciones necesarias para la transmisión exitosa entre el emisor y el receptor deben repetirse en forma invertida al momento en que el receptor envíe la respuesta correspondiente.

Debido al carácter estándar de las conexiones Ethernet, Modbus TCP/IP, estructura del diseño y hardware propuesto, esta solución fue la selección final. Se desarrolló inicialmente siguiendo el diseño planteado anteriormente pero como se mencionó en el capítulo 4, sección 4, el diseño fue re-evaluado y pruebas simuladas

probaron que podía ser mejorado así que se rediseño tomando en cuenta estos nuevos aspectos que se detallan a continuación:

- Al tener dos puertos Ethernet externos, uno para el maestro y otro para la conexión centralizada de esclavos, debe haber una conexión interna por la cual los datos sean procesados, re direccionados y enviados de nuevo a los puertos externos. Esta transmisión interna se puede hacer mediante comunicación serie que posee una velocidad máxima de 0.23 Mbps o mediante I2C con una velocidad máxima de 3.4 Mbps[19].

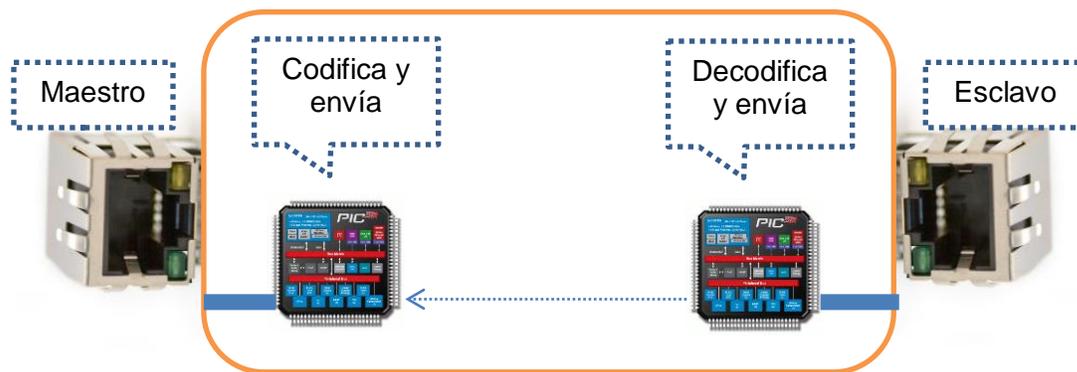
Ahora según podemos ver en el marco teórico una trama completa básica de Modbus TCP/IP tiene 8 Bytes que son fijos en cualquier trama más otros 2 Bytes mínimos que pertenecen a información, así tenemos un total de 10 Bytes por trama básica o lo que es lo mismo 0.08 Kb, esto lo tenemos que multiplicar por 2 ya que la trama hace el siguiente viaje:



**Figura 5.6.** Transmisión de una trama Modbus TCP/IP para una solicitud a través del MDC.

En esta solicitud, el microprocesador #1 recibe la solicitud desde el dispositivo maestro, decodifica, re-direcciona y la envía hasta el microprocesador #2 que la vuelve a codificar como trama Ethernet y la envía.

Esto es necesario ya que el kit de desarrollo escogido maneja solamente un puerto Ethernet a la vez de manera que hay que comunicar dos de ellos para plantear este diseño.



**Figura 5.7.** Transmisión de una trama Modbus TCP/IP para una respuesta a través del MDC.

Así que para una pasada completa de 128 esclavos, la comunicación interna adhiere:

$$2 * \left( \frac{0.00008Mb}{0.12Mb/s} \right) * 128 = 0.17s \quad (1)$$

De retraso suponiendo una solicitud básica de un registro por esclavo, utilizando velocidad máxima disponible en conexión serie (115200 bps), lo que es poco probable tratándose de sistemas de control industriales que manejan miles de variables en registros.

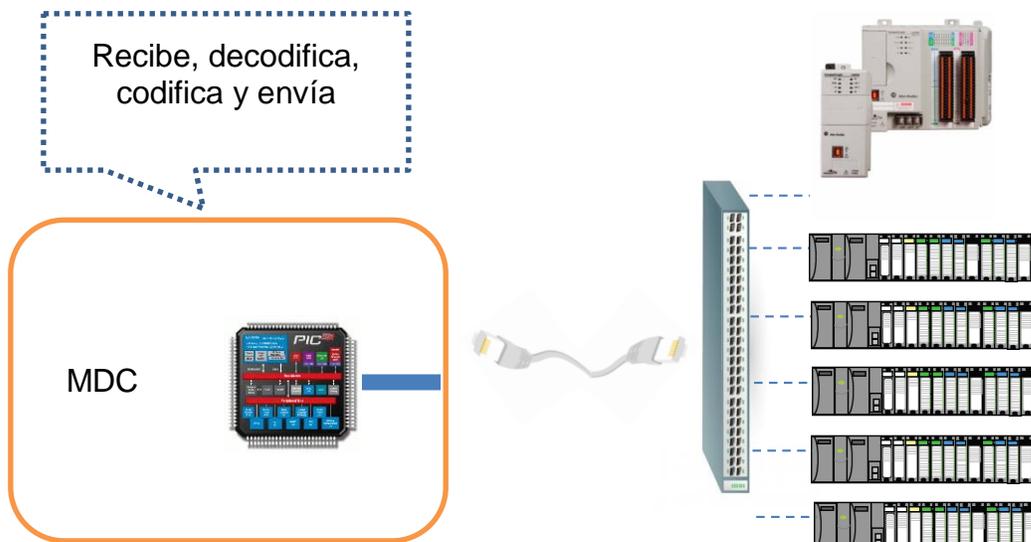
Según el objetivo específico #4 se planteó un tiempo máximo de 4 segundos por solicitud de 128 esclavos, es por ello que una comunicación interna haría que los

tiempos de proceso sobrepasen el tiempo planteado en el objetivo cuando las tramas transmitan cantidad de registros reales.

Es por esto que se rediseño el hardware necesario y se propuso eliminar un puerto Ethernet, ya que las transmisiones de solicitud y de respuesta pueden entrar por el mismo puerto mientras haya un programa bien estructurado que diferencia con la suficiente velocidad una trama de la otra, esto haría necesario una programación un poco más avanzada pero se ganarían nuevas características que acoplan el sistema mucho más a los objetivos y requisitos planteados. Se detallan a continuación:

- Menor tiempo en la transmisión de datos.
- Menos hardware necesario para la construcción del dispositivo final.
- Costos de producción más bajos.
- Prototipo de menor tamaño.

El nuevo diseño posee un solo procesador y un solo puerto Ethernet el cual permite la conexión de un dispositivo master y hasta 128 terminales que se comuniquen mediante Modbus TCP/IP.



**Figura 5.8.** Esquema general de la solución final modificada.

## **5.2 Descripción del hardware**

Todo el desarrollo del proyecto se realizó pesando en un entorno industrial, en el cual las condiciones de operación son extremas, la electrónica debe resistir a los cambios de temperatura, vibración, humedad y a la misma vez ser capaz de operar constantemente sin variaciones de desempeño.

Es por esto que fue seleccionado un kit de desarrollo pequeño pero robusto, de acople modular y con todos los requisitos de hardware necesarios.

### **5.2.1 Kit de desarrollo.**

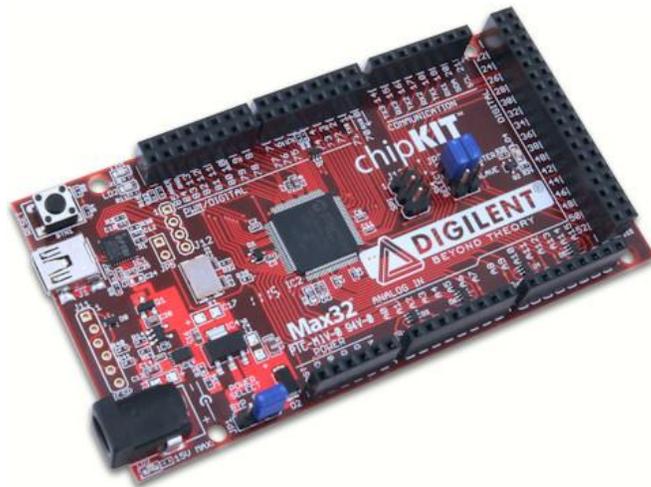
#### *5.2.1.1 chipKIT Max32*

Este es un módulo de desarrollo creado por Microchip el cual provee de un microprocesador bastante potente para sistemas embebidos de pequeño tamaño, así como una amplia gama de conexiones y protocolos de comunicación de manera que brindan una base para conectar otros módulos con diferente funcionalidad, como lo es el módulo de conexión Ethernet el cual nos permite desarrollar una aplicación con comunicación hacia la red.

La programación de este sistema se realiza mediante el entorno MPIDE o MPLAB, este último es un entorno basado en C++ muy completo y de amplias posibilidades de configuración, en cambio MPIDE está enfocado a la simplicidad, siempre basándose en C++ pero con rutinas resumidas y preestablecidas que facilitan la configuración de una manera impresionante y funcional.

Este dispositivo toma ventaja del procesador PIC32 de Microchip el cual presenta la característica de funcionar a 32-bits con un núcleo de instrucciones MIPS corriendo a 80Mhz, estas características fueron de suma importancia al escoger el hardware de desarrollo, contar con un bus de datos de 32-bit otorga una potencia de operación de datos mayor por unidad de tiempo lo que genera una transmisión y procesamiento de instrucciones mucho más rápida que un PIC similar.

De acuerdo a nuestro objetivo de hardware, debemos disponer de un dispositivo que procese los datos de forma muy rápida, esto debido a que se deben interrogar hasta una cantidad de 128 dispositivos con un tiempo de muestreo menor a 4 segundos.



**Figura 5.9.** ChipKIT MAX32 [13]

Este kit de desarrollo nos permite utilizar la potencia del microcontrolador de 32 bits con las facilidades de conexión del diseño modular, y así poder cumplir con el objetivo planteado.

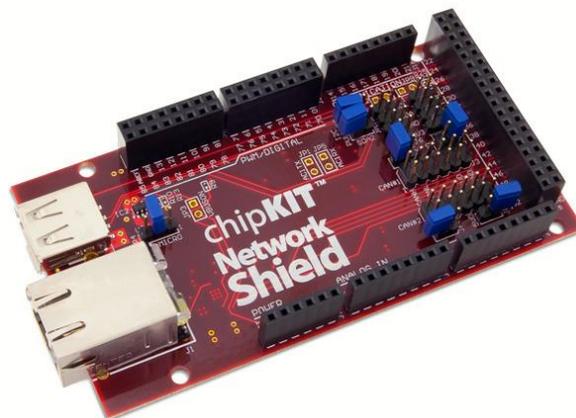
#### **Características Utilizables:**

- Procesador Microchip® PIC32MX795F512
- 80 MHz, 32-bit MIPS

- 512K Flash, 128K RAM
- Controlador USB 2.0
- 10/100 Ethernet MAC
- Factor de forma igual al Arduino Mega
- 83 I/O disponibles
- LED's de indicación

### 5.2.1.2 chipKIT Network Shield

El Network Shield es un módulo de entrada-salida construido para ser capaz de interactuar directamente con el Max32 y proveer comunicación Ethernet al sistema, contiene la circuitería necesaria para explotar las conexiones avanzadas del PIC32 así como del USB, CAN y MAC.



**Figura 5.10.** ChipKIT Network Shield [13]

Este módulo proporciona una Ethernet PHY 10/100 Mbps para permitir la conexión a una red Ethernet mediante el transmisor LAN8720. Además de las funciones de comunicación, el dispositivo también añade una EEPROM I2C 256Kbit para el

almacenamiento de datos no volátil y un oscilador 32.768Khz para permitir aplicaciones en tiempo real y secuencial muy rápidas.

### Características Utilizables:

- Módulo de acoplo con el Max32™
- Transmisor SMSC LAN8720 con velocidad de 10/100 Ethernet PHY
- Conector RJ45.
- Conectores USB Device and Host.
- Two I2C daisy chain connectors
- 256Kbit I2C EEPROM
- Oscilador de 32.768 KHz

#### 5.2.1.3 Transmisor Ethernet

El circuito integrado LAN8720 es un transmisor avanzado de conexión Ethernet 10BASE-T/100BASE-TX lo que provee a la aplicación con un recurso de transmisión de datos con velocidades de 10 y 100 Mb por segundo a través de un conector RJ-45 designado para proyectos de bajo consumo energético.

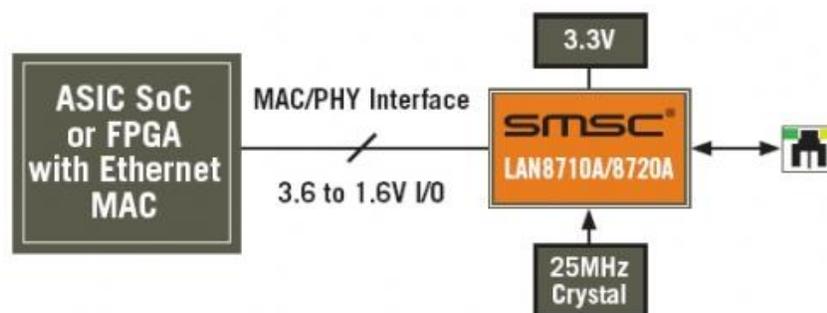


Figura 5.11. Diagrama de bloque de integrado LAN8720[14]

#### 5.2.1.4 Memoria EEPROM I2C 256Kbit

Este dispositivo de memoria eléctricamente borrable (EEPROM) es capaz de operar a diferentes voltajes en un rango de los 1.7v a los 5.5v y ha sido desarrollado para aplicaciones avanzadas de bajo consumo de energía como las comunicaciones en dispositivos embebidos, adquisición de datos y especialmente en este caso para almacenamiento de configuración y datos de escritura esporádica del servidor web embebido.

Tiene una capacidad de grabación de página de hasta 64 bytes de datos y es posible hacer lecturas al azar o secuenciales, al mismo tiempo que permite la interconexión de hasta 8 memorias sobre el mismo bus de datos para aumentar capacidad de almacenamiento.

Esta es una memoria que tiene un tamaño de 32 Kbytes, su estructura está organizada en palabras de 1 byte (8 bits) de longitud, por lo tanto disponemos en total de  $32 \times 8 = 256$  kbits para almacenar información. [15]

#### **Características:**

Organización de Memoria: 32K x 8 (256 Kbit)

Duración: > 1 millón de ciclos de borrado/escritura

Tiempo de escritura: 5 ms

Tipo de empaque: 8-pin DIP

Pueden ser interconectados hasta 8 dispositivos

#### **Especificaciones Clave:**

Requerimiento de voltaje: 2.5 - 5.5 VDC

Comunicación: I2C (CMOS); compatible con 100 kHz y 400 kHz

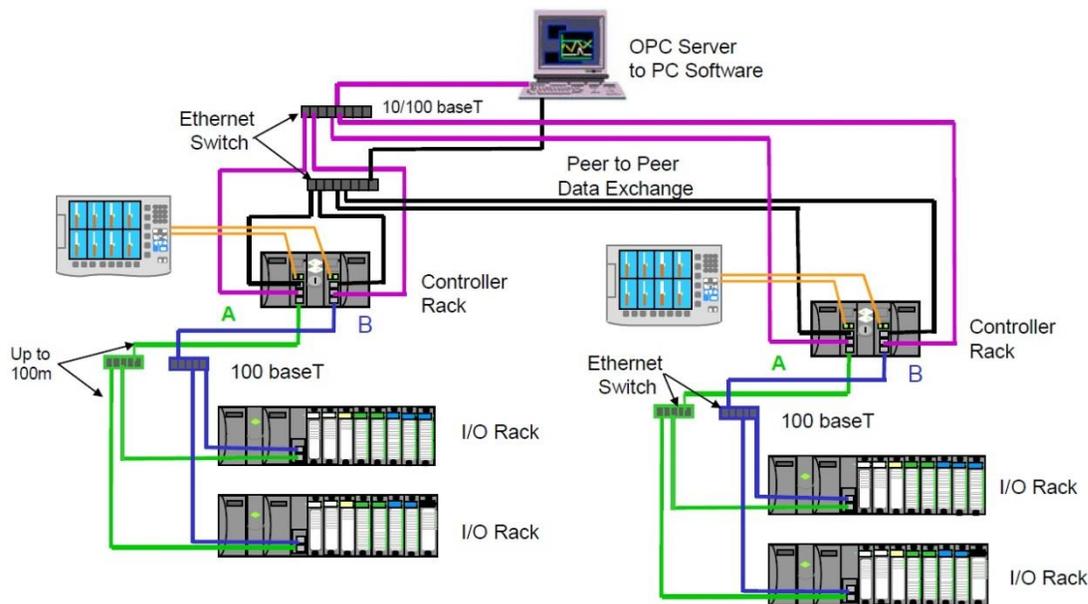
Dimensiones: 0.373 x 0.313 x 0.130 in (9.46 x 7.94 x 3.30 mm)

Temperatura de Operación: -40 to +185 °F (-40 to +85 °C)

### **5.2.2 Controlador HC900**

El controlador híbrido HC900 es un controlador lógico de lazo integrado Honeywell, posee un conjunto de hardware y software que se puede modificar para adaptarse a una amplia gama de aplicaciones en el control de procesos industriales. Tiene una gran ventaja en cuanto a organización, por ejemplo en términos de hardware puede estar incrustado en un solo rack junto con sus módulos de entrada y salida o puede estar conectado en red con otros controladores y módulos; esto da la posibilidad de expandir las dimensiones de control a un amplio rango de procesos.

Este controlador permite a los usuarios crear o modificar mediante el software visual cualquier configuración rápidamente y según las necesidades propias del proceso organizando componentes modulares según la aplicación y lugar, otro gran potencial de este sistema es la capacidad de redundancia.



**Figura 5.12.** Dos sistemas redundantes con HC900 consultados por una PC y 2 HMI [16]

En la figura anterior se presenta un sistema redundante con cuatro controladores HC900 y 2 extensiones de entradas/salidas por cada uno, estos controladores deben tener una dirección IP única si están en la misma red, por lo que se debe consultar uno por uno sus registros, y cambiar de dirección IP para consultar al siguiente.

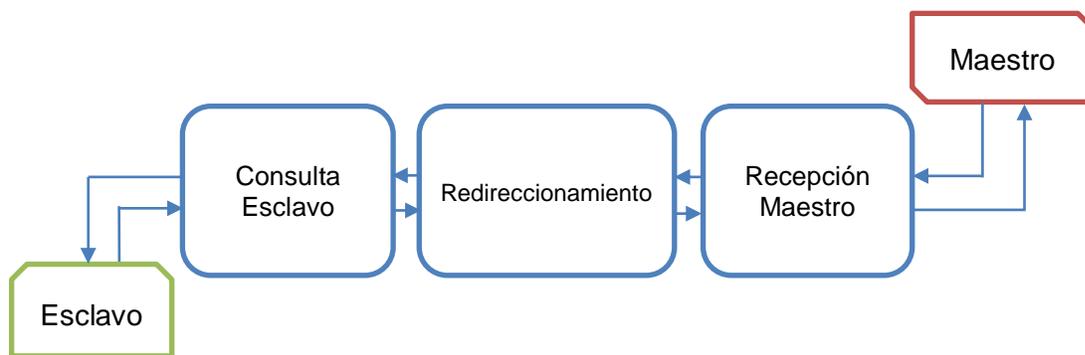
El HC900 está construido para operar comunicaciones mediante Ethernet con otros sistemas Host<sup>1</sup> que se comuniquen mediante el protocolo Modbus TCP, estos pueden ser dispositivos de interfaz humana como las pantallas táctiles industriales o programas de “software” compatibles.

### 5.3 Descripción del software

<sup>1</sup>El término host o anfitrión es usado en informática para referirse a un ordenador o máquina que funciona como punto de inicio o final de las transferencias de datos.[17]

En lo que respecta al diseño de este dispositivo, el software representa una de las partes más importantes de la construcción, ya que determina directamente el desempeño y los valores agregados que pueda tener el prototipo.

Se utilizó un modelo modular para la programación, el cual divide todo un ciclo de funcionamiento en tres grandes partes lógicas denominadas módulo de recepción maestro, módulo de redireccionamiento y módulo de consulta esclavo.



**Figura 5.13.** Diagrama de bloques general de la estructura de software para el MDC

Aplicar el modelo en forma modular tanto en el hardware como en el programa de funcionamiento es una forma de simplificar el problema, dividiendo procedimientos en el programa hacen posible la reutilización de los métodos en diferentes partes del sistema.

Podemos ver en la figura siguiente, la cual representa el diagrama de bloques del sistema completo, que hay pequeños pasos que se repiten, esto quiere decir que si dividimos la programación y creamos un código para solucionar cierto problema, podremos usarlo también para solucionar problemas que se repiten.

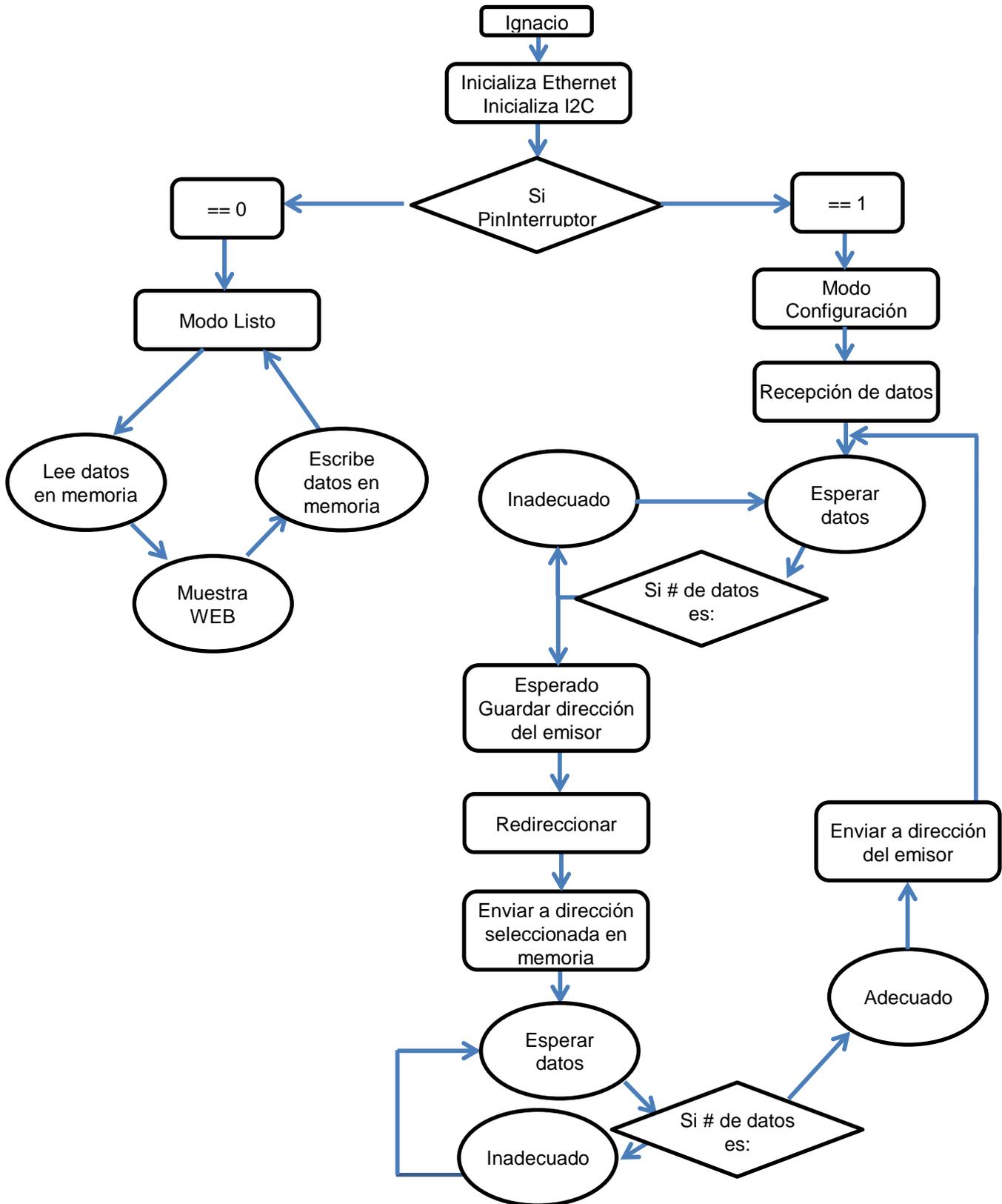


Figura 5.14. Diagrama de bloques del sistema de programación completo

Cuando el dispositivo se enciende ejecuta un código de configuración en el cual se inicializan todos los componentes necesarios para recibir una transmisión Ethernet que contenga una trama Modbus TCP/IP. Se leen de memoria interna direcciones IP configuradas y se establecen en la configuración del transmisor de red.

Inicialmente el dispositivo lee una entrada digital por medio del hardware y dependiente del estado de esta entrada entra a un estado de configuración o al estado normal de funcionamiento en el cual recibe solicitudes por parte del dispositivo maestro.

En el estado de configuración, se repite un procedimiento en el cual se leen datos de configuración desde la memoria interna, luego se despliegan los datos con un formato WEB<sup>2</sup> para actuar como interfaz de usuario. Estos datos están en continuo despliegue mediante un ciclo del programa y al momento de presionar un botón específico, estos datos se guardan en la memoria interna y el ciclo se repite nuevamente.

El sistema debe reiniciarse para ponerse de nuevo en estado de recepción de solicitudes, un interruptor externo de hardware determina la entrada digital de la cual depende el estado de operación en el cual inicie el MDC.

### **5.3.1 Modulo de recepción maestro.**

Este código presenta la primera etapa del proceso de recepción, el programa entra en un ciclo de espera controlada en el cual, cada vez que se lee algo nuevo desde el puerto de red se verifica la validez del dato y se comprueba si la cantidad de datos es correcta para una transmisión de solicitud tipo Modbus.

---

<sup>2</sup> Un grupo de páginas World Wide Web que por lo general contienen hipervínculos entre sí y se encuentran a disposición para ser visto en línea.[20]

Cada dato válido es extraído de la trama completa Ethernet y almacenado en un arreglo que conforme se llena va comprobando si posee la cantidad de datos correctos para formar una trama Modbus completa, mientras esta trama no esté llena, el ciclo de recepción de solicitud queda abierto.

Al tener lista la trama y verificada, la dirección que envió los datos es guardada en una variable IP temporal, esto para ser utilizada de nuevo al momento de que el esclavo envíe su respuesta y esta deba ser devuelta al módulo maestro. Finalmente estos datos son pasados al próximo módulo mediante el arreglo de variables, para ser re-direccionado.

### **5.3.2 Módulo de Redireccionamiento**

El módulo anterior (Recepción maestro) provee el arreglo de bits que contiene la trama que se quiere analizar con este módulo, se extraen ciertas variables que contienen datos importantes para determinar el tipo de trama que se está analizando. Primeramente se extrae la variable llamada "Unit ID", la cual pertenece al encabezado de protocolo de aplicación Modbus (MBAP por sus siglas en inglés) que será utilizada para seleccionar uno de los 128 dispositivos terminales posibles a consultar, así que esta es una de las principales variables para el direccionamiento, ya que en la memoria interna del sistema se guarda la información que direcciona un número de unidad (Unit ID) con una dirección IP configurada anteriormente por el usuario.

Luego de tener la dirección IP a la cual se debe enviar la consulta se extraen los demás datos necesarios, como el código de función para determinar el tipo de registro a ser consultado.

El código de función se evalúa en todos sus valores posibles y depende del valor identificado se determina el tipo de petición para el esclavo, esto también determina cómo será evaluado el resto de información en la trama.

El número de transacción es necesario para determinar si la secuencia que se recibe está correcta, y si no se han presentado errores de comunicación, cabe resaltar que el número de transacción que se recibe en una solicitud debe ser copiado exactamente igual al momento de crear una respuesta, ya que si esto no sucede entonces el dispositivo maestro no considera la trama como una respuesta válida y causa errores en la comunicación.

Así sucesivamente se van eliminando partes de la solicitud que representan instrucciones Modbus para determinar cómo tratar la última parte de la trama, la cual contiene los datos a ser solicitados. Más específicamente al momento de leer una solicitud, estos datos son solo dos bytes con valor de cero, pero el campo de tamaño (Length field) especifica cuántos registros deben ser leídos y retornados en la trama de respuesta.

Podemos ver entonces que el tamaño total de cada solicitud, es más pequeño que su respectiva respuesta.

Luego de extraer toda la trama Modbus, byte por byte se debe empaquetar de nuevo en un encabezado Ethernet que debe contener la dirección IP que se leyó de memoria en el primer módulo.

Al tener la trama reconstruida, con el número correcto de bytes, esta información se encuentra lista para ser pasada al próximo y último módulo.

### 5.3.3 Módulo de consulta esclavo

En este caso lo que debe ejecutar esta parte del código es un proceso invertido del primer módulo, el conjunto de bytes seleccionado en la parte anterior contiene la estructura Modbus básica completa, esta es encapsulada y enviada a la nueva dirección IP, esto requiere dar un tiempo de espera para interpretar la respuesta de parte del terminal esclavo.

Si no se recibe ninguna respuesta, se interpreta como un error de conexión o de terminal, esto retorna un código de error al dispositivo maestro, ahora en caso contrario la respuesta es la esperada y al igual que se recibió en el primer módulo, se extrae parte por parte analizando cada una para comprobar que no existan errores en la estructura.

Cada dato analizado se almacena en el arreglo de bytes temporal para luego ser utilizado.

Así entonces el proceso se invierte y los datos son enviados al módulo de Redireccionamiento nuevamente, esta vez utilizando la dirección IP temporal que pertenecía al maestro.

Como podemos ver el problema global se trata como 3 problemas secundarios, lo cual resulta muy útil y efectivo, reutilizando partes de código es posible simplificar las tareas a realizar en el entorno de programación.

### 5.3.4 Entorno de programación del controlador híbrido HC900.

La programación de la unidad central de procesamiento del HC900 se realiza por medio del software llamado “Hybrid Control Designer”, una muestra es expuesta mediante la figura x.

Este entorno se diferencia de los demás en su interfaz de usuario. La cual es muy fácil de configurar y posee un carácter muy abierto tanto que no es necesario conectarse mediante protocolos propietarios del sistema, lo cual es muy común en sistemas robustos y conocidos.

El programa permite seleccionar entre una variada gama de configuraciones para el protocolo de comunicación, ya sea Modbus estándar, Modbus personalizado, variables float 32, float 16, etc, esto permite hacer conexiones a dispositivos de terceros utilizando un mismo controlador.

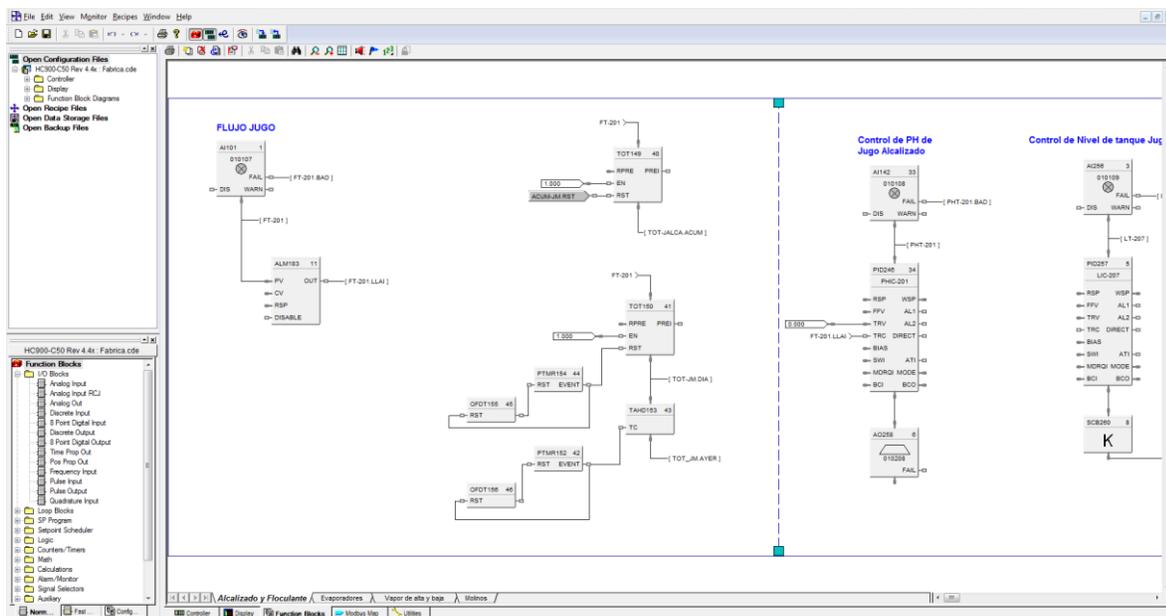


Figura 5.15. Entorno de programación gráfico para el HC900.

El Hybrid Control Designer es un programa de alto nivel desarrollado por Honeywell para la gama de controladores HC900 y tiene la característica de ser totalmente visual, una

ventaja sobre los demás controladores que se configuran mediante diagrama de escalera.

El entorno visual permite al usuario seleccionar de la lista una gran variedad de bloques que representan funciones, operaciones booleanas, matemáticas, secuenciales y principalmente de control de procesos. Estos bloques son arrastrados hasta la hoja de trabajo, donde son configurados mediante parámetros numéricos. La interconexión entre estos bloques se da en forma visual también, se dibujan líneas tipo cable entre cada bloque para definir una configuración en conjunto, este grupo de bloques controlan el modo de funcionamiento del controlador y al mismo tiempo que las salidas y entradas que son conectadas al mismo.

## Capítulo 6: Análisis de Resultados

El objetivo principal del proyecto es implementar un dispositivo concentrador que funcione sobre una red Ethernet con protocolo de aplicación Modbus TCP, para lograr configurar esta red es necesario configurar los terminales esclavos que en este caso serán controladores híbridos programables. Estos controladores de marca Honeywell, modelo HC900 como se especifica en la descripción del software, se configuraron mediante direcciones IP en el rango de 10.1.0.X para obtener una red pequeña de controladores, en donde cada uno de ellos posee una rango de registros internos que contienen información acerca de las variables de proceso que se están controlando o monitoreando.

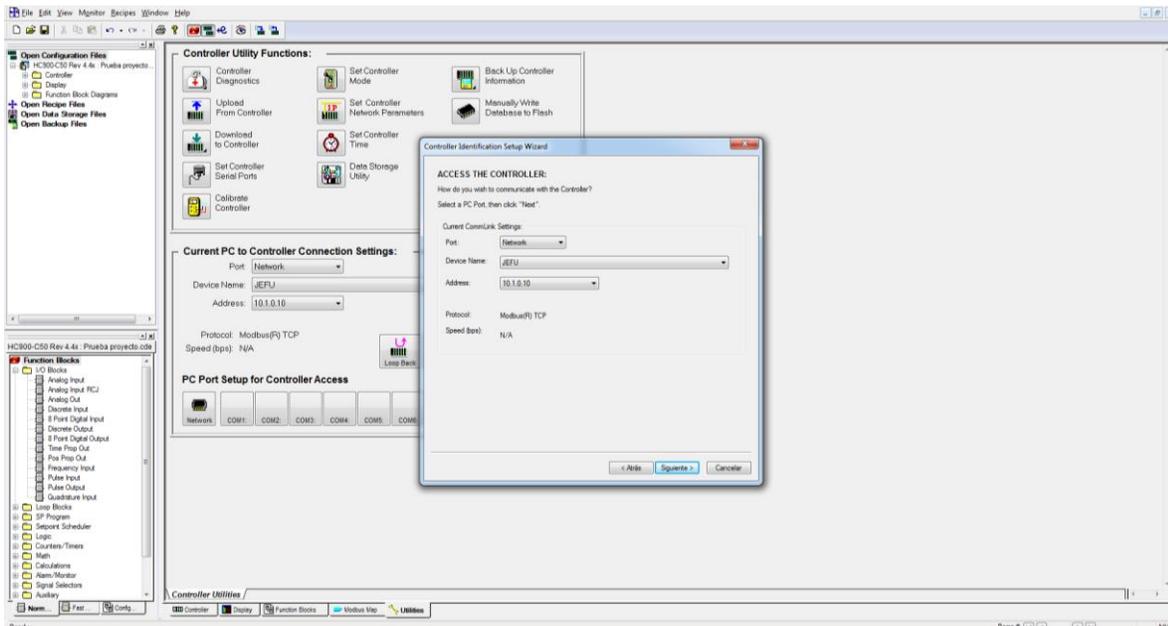
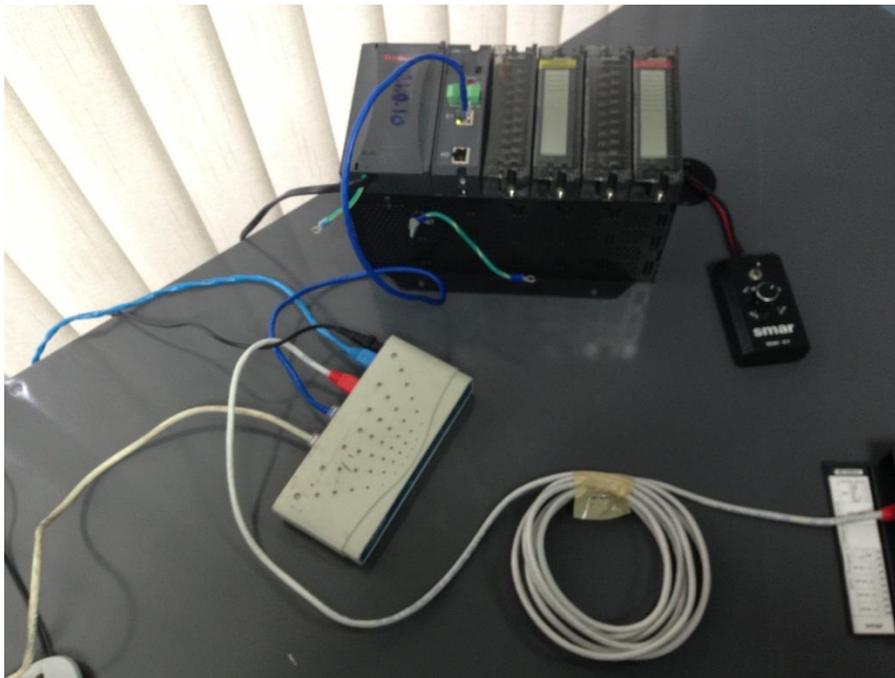


Figura 6.1. Ventana de visualización de dirección IP en el controlador híbrido programable #1.

Podemos apreciar en la figura X el resultado de la verificación que realiza el programa “*Hybrid Control Designer*” sobre la dirección IP 10.1.0.10 configurada en dicho dispositivo.

Igualmente se configuró un segundo controlador con una dirección IP 10.1.0.20, que representa un segundo terminal con variables propias en sus registros internos. Estos dos controladores son interconectados por un conmutador de direcciones IP para que luego se comunique mediante un solo cable Ethernet con el concentrador de datos Modbus como se muestra en la figura 5.17



**Figura 6.2.** Conexión de 2 controladores conmutados en una subred 10.1.0.X

Dicha conexión resulto ser una prueba contundente del objetivo de hardware número 3, en el cual se especifica conectar los controladores para uso industrial en una red funcional y eficiente.

Esta pequeña red permite analizar el desempeño máximo del MDC ya que uno de los mayores retrasos en la solicitud de datos por parte de este dispositivo es el hecho de inicializar una conexión con una nueva dirección IP, cada vez que se tiene que consultar a un controlador diferente es necesario cerrar la conexión actual, solicitar nueva conexión al nuevo dispositivo y esperar su mensaje de aceptación. No es hasta que este “Acknowledge” se reciba que el MDC puede empezar a transmitir datos hacia los terminales esclavos, por esta razón el trámite de solicitudes a diferentes direcciones IP tiene su mayor retardo en la negociación de una comunicación exitosa.

Al analizar resultados en las pruebas es evidente que fue uno de los códigos más importantes a depurar, y después de realizar suficientes modificaciones se lograron resultados menores a los 7ms en el tiempo de negociación, lo cual para un total de 128 esclavos consultados en una sola ronda, se produce un retraso máximo de 896ms, cantidad que se mantiene dentro de los márgenes del objetivo específico número 4.

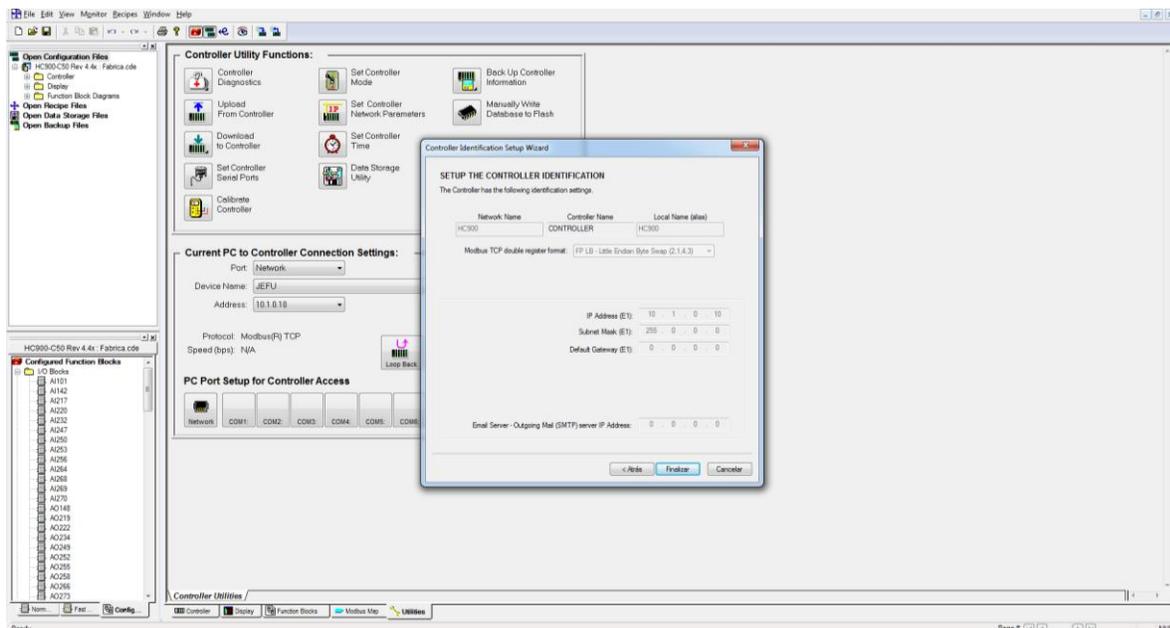
Internamente en cada controlador se configuró el protocolo Modbus sobre TCP/IP mediante la especificación “Little Endian Byte Swap<sup>3</sup>” la cual es un tipo de orden de bytes en las comunicaciones de este tipo, consecuentemente el controlador #2 se configura de igual manera. Esta característica fue esencial al momento de programar el MDC ya que el modo en que los bytes se reciben cambia totalmente la interpretación del dato, que en bajo nivel es un número binario ordenado de forma sumamente específica.

---

<sup>3</sup> Little Endian Byte Swap define el orden en el cual los bytes se transmiten, en este caso una palabra que contiene 4 bytes se envía: primero el byte #2, luego el #1, seguido del #4 y finalmente el #3.[21]

En un inicio esta situación dio problemas ya que no es la configuración que viene por defecto en los controladores HC900, esto causaba que el dato recibido en el MDC era diferente al dato dentro del registro que se estaba solicitando, se creyó en un principio que el error provenía de un mapeo erróneo en el controlador pero luego de realizar análisis profundos mediante *software*<sup>4</sup> analizador de protocolos se verificó el problema provenía de un orden diferente al momento de interpretar los datos en el concentrador de datos y esto quería decir que uno de los dos componentes estaba ya sea enviando o recibiendo los bytes en un orden distinto.

Según información adquirida y gama de dispositivos en el mercado actual, la opción de “Little Endian Byte Swap” es la más estandarizada de las posibles, esto determino un criterio para su elección.

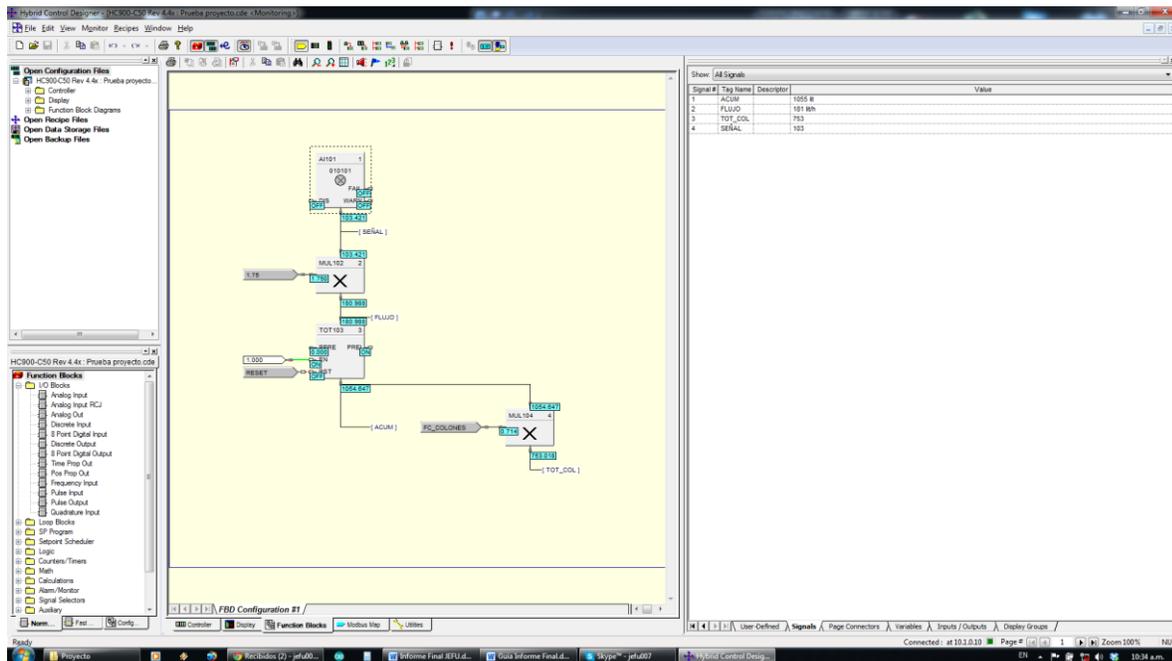


**Figura 6.3.** Ventana de visualización de parámetros de red y formato Modbus configurado en los controladores HC900.

<sup>4</sup> Término genérico que se aplica a los componentes no físicos de un sistema informático[22]

En términos de logica de control, se diseño un pequeño sistema que realiza lectura de entradas analogicas, las cuales son simuladas mediante un potenciómetro de precisión, luego son multiplicadas por un factor numerico para simular un un flujo variable, luego se utiliza un bloque totalizador que realiza una sumatoria continua del flujo(Lt/h).

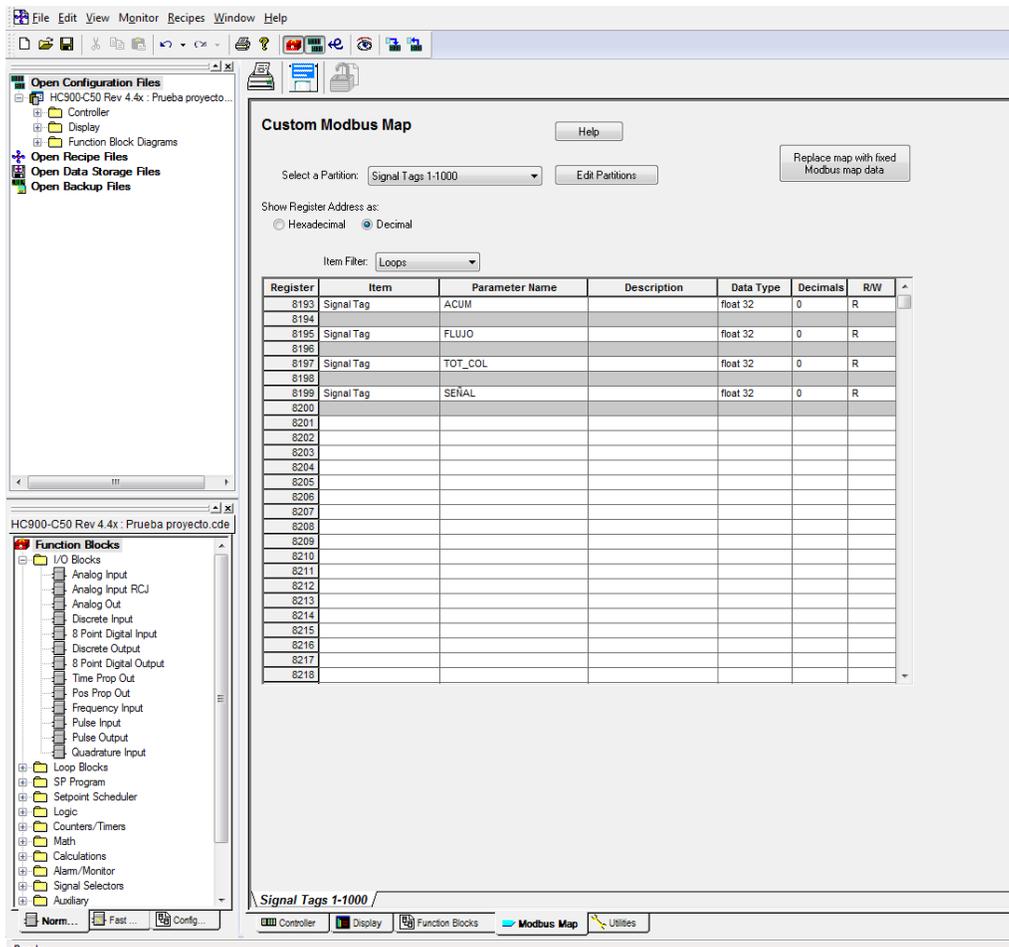
El bloque final toma la variable de salida del totalizador para calcular el costo total que se ha utilizado, según un factor de conversion a colones. Esto es un simulador de gastos según el fluido medido y funciona muy bien para realizar las pruebas requeridas en la red de controladores. Se aprecian los bloques de control en la siguiente figura.



**Figura 6.4.** Configuración de bloques para la manipulación de registros dentro del controlador híbrido programable HC900.

Como se aprecia en la figura anterior se programaron cuatro variables analógicas dentro de la estructura del controlador, cada una se mapeo en los registros del dispositivo, los cuales son consultados mediante el protocolo Modbus.

Dentro de la configuración Modbus (Figura 6.5) se mapearon las variables desde la dirección de registro 8193, hasta la dirección 8198, todas las variables análogas tienen un tamaño de 32bits y cada dirección contiene espacio de 16bits, así que estas cuatro variables se deben mapear en dos registros cada una.



**Figura 6.5.** Configuración de mapeo Modbus dentro del controlador híbrido programable HC900.

La primera prueba se realizó configurando una red virtual entre la computadora y el MDC, para la cual se utilizaron programas de simulación de protocolos Modbus TCP, un programa en la computadora simula el dispositivo maestro que envía las solicitudes al

concentrador y otro programa en una computadora diferente simula un dispositivo esclavo que responde a las solicitudes.

Se obtiene resultados muy prometedores con tiempos muy rápidos de acceso, se detalla en la figura 6.6 a continuación las tres ventanas que presentan la información de la transferencia.

Al lado izquierdo se detallan las transacciones reportadas por el simulador de esclavo Modbus, el cual presenta un historial de solicitudes concretadas exitosamente. Se puede resaltar que es consultado el esclavo # 1, mediante lectura de "HoldingRegisters" y luego en el lado derecho comprobamos que efectivamente el programa simulador de maestro está solicitando un registro de 16 bits, "HoldingRegister" a la dirección #1.

Luego también se verifica que ambos simuladores están comunicándose a través de un dispositivo con dirección IP igual a 10.1.0.5, el cual es el MDC, que se encuentra conectado entre ambas computadoras.

El tiempo de procesamiento reflejado en la ventana del lado derecho en la parte baja es mostrado por el MDC, el cual cada vez que termina de transmitir despliega el tiempo transcurrido desde que recibió la solicitud de parte del maestro. El tiempo promedio por transacción total es de 9ms, esto quiere decir que se estiman 1152ms + 896 de retraso considerando que todos necesitan negociación por IP diferente, producen 2.048 minutos para consultar 128 terminales esclavos.

Esto cumple con los cuatro objetivos específicos planteados, obteniendo resultados con mucha eficiencia, utilizando procedimientos de programación por módulos depurados y reutilizados que brindan el nivel que buscado para el ambiente de control industrial sobre procesos.



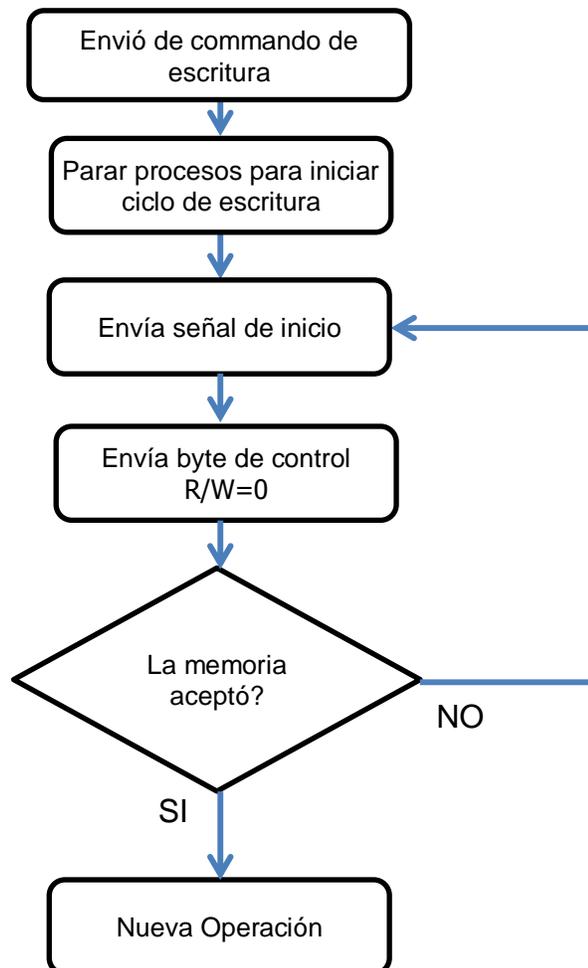


**Figura 6.7.** Concentrador de datos Modbus TCP/IP conectado mediante Ethernet.

Así también se presentaron problemas de tipo Software-Hardware muy difíciles de ver si no se tiene un conocimiento sobre el hardware que se utiliza, como por ejemplo, al crear una función para escritura de bytes en la memoria EEPROM 24LC256 que se comunica mediante el protocolo I<sup>2</sup>C la compilación del código no producía ningún error o advertencia, toda lógica parecía estar correctamente planteada, sin embargo al momento de quemar el programa en el microcontrolador y poner en marcha el sistema no funcionaba y no hacía ninguna escritura correcta como se esperaba.

Al tener conocimientos de hardware y sus principios básicos, podemos difamar que el las memorias, o registros en general tienen tiempos de acceso mínimos, dependientes del tipo de dispositivo y conexión

Entonces en este caso el retardo de escritura de esta memoria depende de las características de construcción y el tiempo de ACKNOWLEDGE<sup>5</sup>; se puede analizar la estructura del proceso de escritura en una memoria mediante el siguiente diagrama de bloques de la figura 6.8.



**Figura 6.8.** Diagrama de bloques del proceso de aceptación de escritura en una memoria EEPROM 24LC256.

<sup>5</sup> ACK una señal enviada por la estación receptora (destino) de vuelta a la estación emisora (fuente) después que la recepción de un bloque reconocible de datos de tamaño específico sea correcta. [23]

Según estos datos y mediante pruebas de acceso, se determinó un tiempo de espera de 5 mili segundos para que la memoria complete correctamente un ciclo de escritura, así que esto se refleja en el código como un retardo (“delay”) presente luego de las instrucciones de escritura. Esto permitió solucionar un problema que se presentó durante la confección del programa que controla la memoria.

## Capítulo 7: Conclusiones y recomendaciones

### 7.1 Conclusiones.

- Se cumplieron los objetivos específicos con rango de efectividad mejor al esperado.
- El objetivo específico número cuatro, que establecía un tiempo máximo de acceso de 4s fue superado y se logró un tiempo promedio de menos de 2.5s.
- La configuración de una red de dispositivos se estableció exitosamente mediante la escogencia de controladores que utilizan la conexión Ethernet para intercambiar datos.
- Fue necesario utilizar programas analizadores de paquetes de datos sobre conexiones Ethernet para solucionar problemas durante el proyecto.
- Se logró implementar una red de dispositivos que utilizan Modbus TCP/IP como su protocolo de aplicación.
- Para resolver el problema, tanto de “Hardware” como de “Software” se utilizó un sistema modular que divide los problemas antes de solucionarlos.
- Las memorias EEPROM son dispositivos que necesitan tiempos de acceso de escritura considerables al momento de ser incluidas en un diseño electrónico.

- El tiempo que dura un dispositivo Ethernet en establecer una conexión exitosa con otro dispositivo representa el retardo más significativo en este sistema.
- Un controlador híbrido programable posee un banco de registros en el cual se guardan variables de control continuamente.
- Las variables analógicas en el sistema de control desarrollado pueden ser de 16 bits o de 32 bits.
- Se logró un porcentaje de peticiones exitosas del 100% entre dispositivo master y esclavos.
- La comunicación Modbus TCP/IP aplicada a este sistema brinda una propiedad de estandarización que se buscaba para este sistema en el mercado actual y futuro.
- El orden en que se envían los bytes en este sistema se estableció según configuración “Little Endian Byte Swap”.

## 7.2 Recomendaciones

- Es fundamental tener en cuenta que el dispositivo diseñado puede tener acceso hasta un máximo de 128 esclavos, si se quiere realizar una mejora en cuanto a número de esclavos accesibles se debe considerar la memoria disponible del sistema para guardar direcciones adicionales.
- El método de despliegue de datos mediante página WEB está ajustado a las necesidades de este proyecto y las cumple a cabalidad. Es necesario estudiar si es capaz de aumentar la cantidad de datos a desplegar.

-

## Bibliografía

- [1] A. A. Rincon Charris, "INTRODUCCION AL CONTROL AUTOMATICO DE PROCESOS," vol. 2012, pp. 5, Junio 2012, 2012.
- [2] I. Modbus Organization, "MODBUS PROTOCOL," vol. 2012, pp. 3.
- [3] Modbus-IDA, "MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDEV1.0b," *Modbus IDA*, vol. 1, pp. 46, October 24, 2006, October 24, 2006.
- [4] S. B. Morriss. Programmable Logic Controllers 1999.
- [5] Kevin Bothamley and Jim Rodgerson, "Emerging Ethernet Protocols", USA, 2001.
- [6] B. Dutertre. Formal modeling and analysis of the Modbus protocol. Technical report, Computer Science Laboratory, SRI International, October 2006.
- [7] Iban Loiola Zubia, Autómatas Programables, el PLC, Diciembre 2001. Internet, Consultado 2 de Octubre. <http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/plc.htm>
- [8] Modbus Organization, "MODBUS PROTOCOL, specifications. Disponible en: <http://www.modbus.org/specs.php>
- [9] Acromag Incorporated, Introduction to Modbus TCP/IP, 2005, Disponible: [http://www.dee.hcmut.edu.vn/vn/ptn/sch/download/Network\\_Architecture/intro\\_modbusTCP.pdf](http://www.dee.hcmut.edu.vn/vn/ptn/sch/download/Network_Architecture/intro_modbusTCP.pdf)
- [10] Margaret Rouse-TechTarget, "Ethernet definition", 2006, Disponible: <http://searchnetworking.techtarget.com/definition/Ethernet>

[11] Scott Kirsner, "The Legend of Bob Metcalfe", 1998, disponible en línea: <http://www.wired.com/wired/archive/6.11/metcalfe.html>

[12] TecShout, JP. "What is an IP Address ?", 31 Diciembre del 2011, Disponible en línea: <http://tecshout.com/what-is-ip-address/>

[13] Digilent Inc. " chipKIT Max32™ Prototyping Platform", Disponible en línea: <http://www.digilentinc.com/Products/Detail.cfm?Prod=CHIPKIT-MAX32>

[14] SMSC. Small Footprint, Low Power Consumption, Full-Featured 10/100 Ethernet Transceivers. Disponible en línea: [http://www.smsc.com/Products/Ethernet\\_and\\_Embedded\\_Networking/Ethernet\\_Transceivers/LAN8710A\\_LAN8720A](http://www.smsc.com/Products/Ethernet_and_Embedded_Networking/Ethernet_Transceivers/LAN8710A_LAN8720A)

[15] Microchip. "256K I2C CMOS Serial EEPROM Datasheet", Disponible en línea: <http://ww1.microchip.com/downloads/en/devicedoc/21203m.pdf>

[16] Honeywell. Controlador Híbrido HC900, Guía de usuario", Disponible en línea: <https://www.honeywellprocess.com/library/support/Public/Documents/51-52-25-107-SP.pdf>

[17] Computer Desktop Encyclopedia, "host - technical definition"; Disponible en línea: <http://www.science.yourdictionary.com>

[18] Prosoft Technology, "C' Programmable Application Development Module with Ethernet for CompactLogix"; Disponible en línea: <http://www.prosoft-technology.com>

[19] I2C Bus, "I2C Bus / High Speed Mode"; Disponible en línea: <http://www.i2c-bus.org/highspeed/>

[20] Merriam Webster, "Web Site definition"; Disponible en línea: <http://www.merriam-webster.com/dictionary/web%20site>

[21] University of Maryland, "Big and Little Endian"; Disponible en línea: <http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Data/endian.html>

[22] WordReference, "Software definition"; Disponible en línea: <http://www.wordreference.com/definicion/software>

[23] Margaret Rouse; SearchNetworking, "ACK DEFINITION"; Disponible en línea: <http://searchnetworking.techtarget.com/definition/ACK>

## Apéndices y anexos

### A.1 Glosario y Abreviaturas

100BaseT: Normativa estándar para los cables de conexión Fast Ethernet, doble par trenzado categoría 3, 4 o 5.

10BaseT: Normativa estándar para los cables de conexión Ethernet, doble par trenzado de 0.4mm a 0.6mm de diámetro.

Ethernet: Estándar de la capa física del modelo OSI, que brinda servicios de transmisión de datos en redes LAN.

IP: Protocolo encargado del direccionamiento y enrutamiento de mensajes recibidos y enviados.

TCP: Protocolo de la capa de transporte que basa su funcionamiento en la transmisión de datagramas libres de error.

Modbus: Protocolo de la capa de aplicación que proporciona comunicación cliente/servidor entre dispositivos conectados en una red basada en común.

Modbus TCP/IP: Protocolo de la capa de aplicación que proporciona comunicación cliente/servidor entre dispositivos conectados a través de una conexión Ethernet sobre TCP/IP.

Modelo OSI: Modelo basado en 7 capas para la estandarización de los diseños de dispositivos de comunicación.

UDP: Protocolo de la capa de transporte que basa su funcionamiento en la transmisión de gran cantidad de paquetes sin revisión de errores.

## A.2. Abreviaciones

PLC: Programmable Logic Controller.

RTU: Remote Terminal Unit.

SCADA: Supervisory Control and Data Acquisition.

TCP: Transmission Control Protocol.

UDP: User Datagram Protocol.

APDU: Application Protocol Data Unit.

DNS: Domain name system.

DTE: Data Terminal Equipment.

HTTP: Hypertext Transport Protocol.

IP: Internet Protocol.

LAN: Local Area Network.

OSI: Open System Interconnection.

PDU: Protocol of Data Unit

PLC: Programmable Logic Controller.

RTU: Remote Terminal Unit.

SCADA: Supervisory Control and Data Acquisition.

TCP: Transmisión Control Protocol.

UDP: Usar Datagrama Protocol.

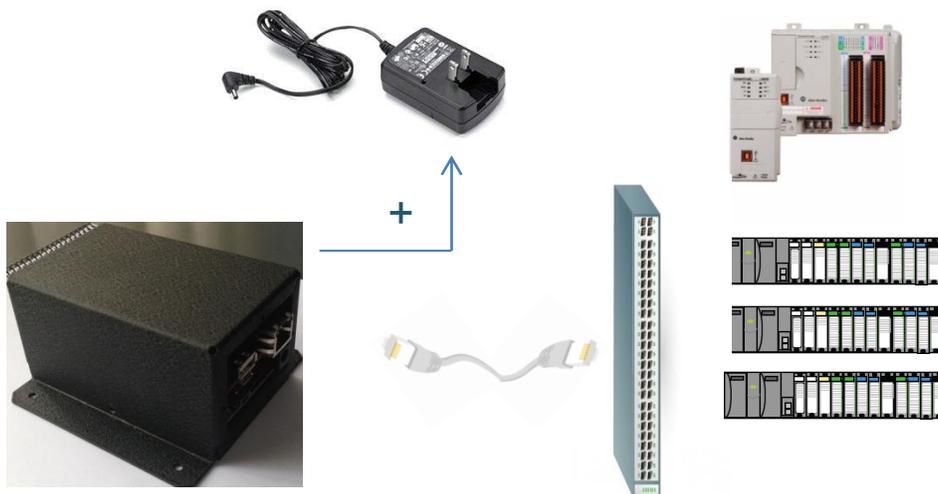
# Manual de Usuario

**Concentrador de datos Modbus TCP/IP.**

**128 Esclavos re-direccionables**

## Conexión del dispositivo.

- 1- Conecte el dispositivo a una fuente de poder de 5v DC o mediante la conexión autoalimentada USB.
- 2- Conecte el MDC a la red Ethernet en donde se desea operar, vea la siguiente figura.



- 3- Antes de encender el interruptor de poder, coloque el interruptor de modo en estado configuración.
- 4- Active el MDC mediante el interruptor de poder.

## Configuración del dispositivo.

- 1- Acceda mediante un navegador WEB a la dirección IP que viene por defecto en el MDC 10.1.0.10
- 2- Configure si lo desea dirección IP, Puerta de enlace y mascara de subred necesarios para conectarse a su red.

Configuración de Parametros

Parametros de este dispositivo

Dirección IPv4:

Máscara de subred IPv4:

Puerta de enlace:

Parametros de direccionamiento

Slave	IP	Registro
00 1	010.001.000.010	08199
00 2	000.000.000.000	00000
00 3	000.000.000.000	00000
00 4	000.000.000.000	00000
00 5	000.000.000.000	00000
00 6	000.000.000.000	00000
00 7	000.000.000.000	00000
00 8	000.000.000.000	00000
00 9	000.000.000.000	00000
010	000.000.000.000	00000
011	000.000.000.000	00000
012	000.000.000.000	00000
013	000.000.000.000	00000
014	000.000.000.000	00000
015	000.000.000.000	00000
016	000.000.000.000	00000
017	000.000.000.000	00000
018	000.000.000.000	00000
019	000.000.000.000	00000
020	000.000.000.000	00000
021	000.000.000.000	00000
022	000.000.000.000	00000
023	000.000.000.000	00000

- 3- Presione el botón guardar que esta el final de la configuración.
- 5- Luego de que la página web termine de guardar datos, puede empezar a asignar direcciones IP de terminales esclavos en el campo IP asignado para estas.

**6- Puede incluir hasta 128 dispositivos**

**7- Presione el botón “Guardar Datos”**

**8- Apague el dispositivo.**

**9- Mueva el interruptor de configuración al modo Modbus.**

**10- Encienda el dispositivo.**

**11- El concentrador de datos está listo para ser usado en su red.**

# Anexos

## ChipKit Max32 Reference Design

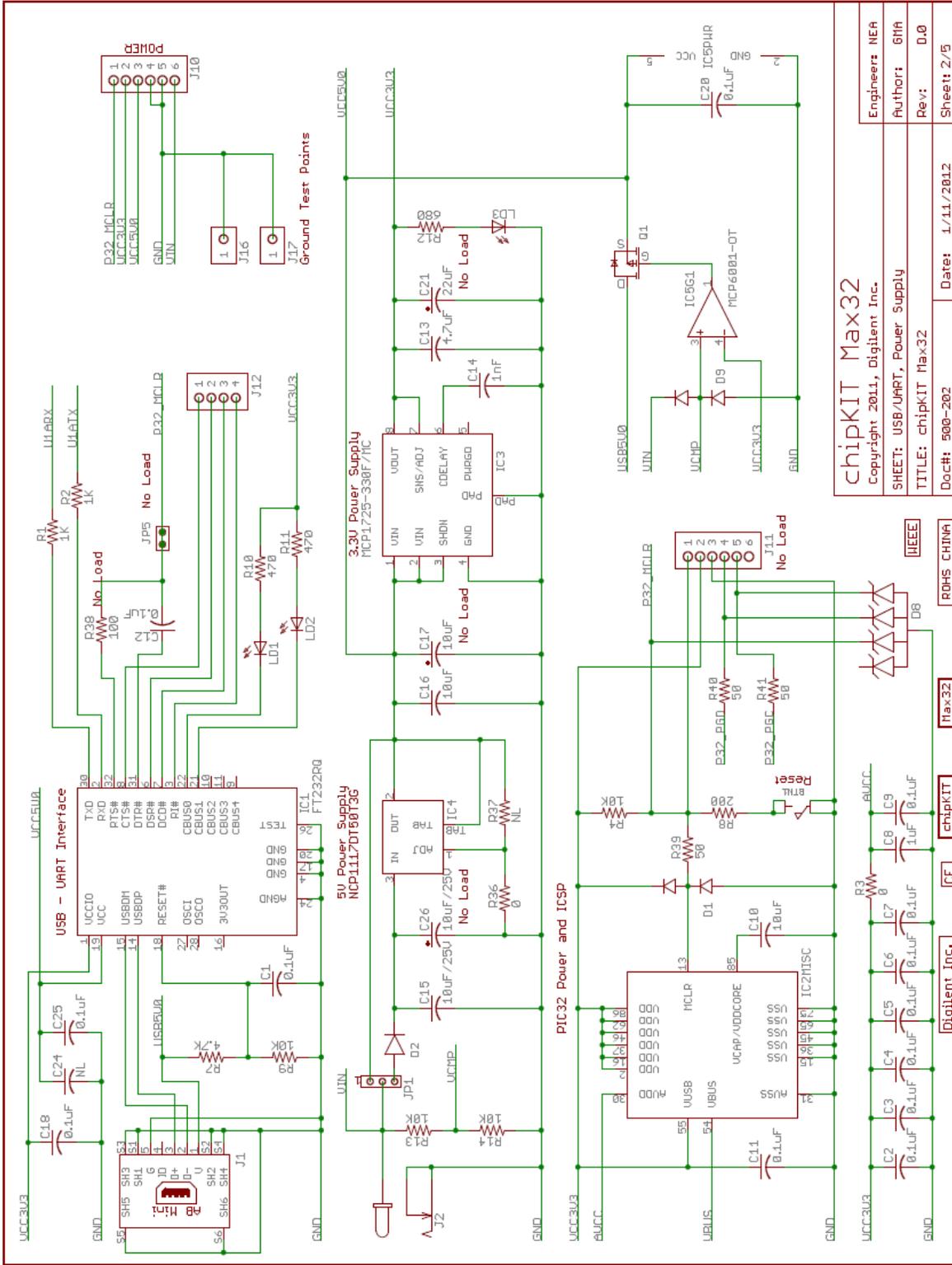
### chipKIT™ Max32™ Reference Design

Digilent chipKIT hardware files are licensed under the Creative Commons Attribution-ShareAlike license. This license allows for both personal and commercial derivative works, as long as proper credit is given to Digilent and the designs are released under the same license.

[www.creativecommons.org/licenses/by-sa/3.0/](http://www.creativecommons.org/licenses/by-sa/3.0/)

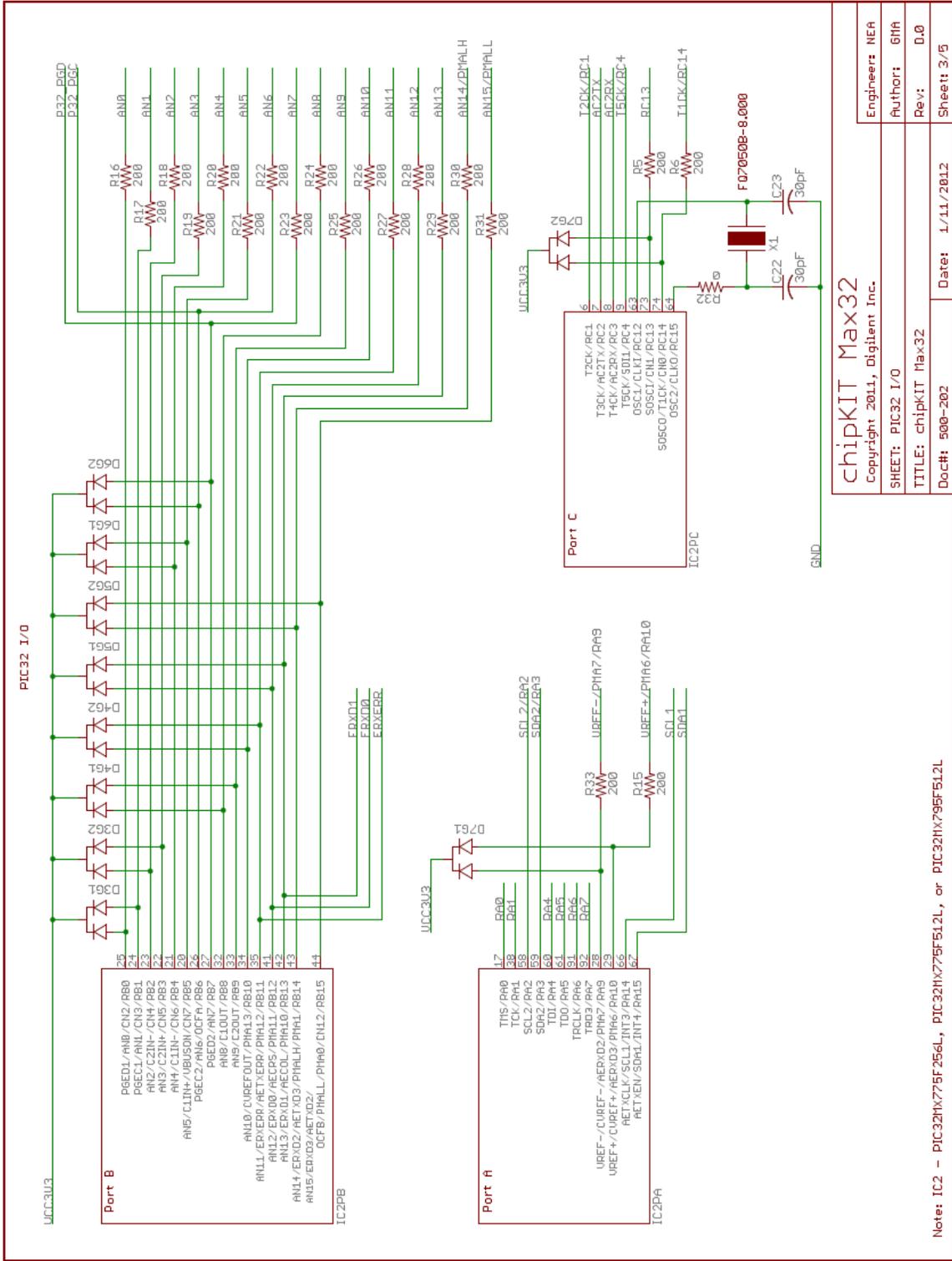
Reference designs are provided "AS IS" and "WITH ALL FAULTS". Digilent disclaims all other warranties express or implied, regarding this product, including but not limited to, any implied warranties of merchantability or fitness for any particular application. Digilent may make changes to specifications and product descriptions at any time, without notice.

chipKIT Max32		Engineer: NEA
Copyright 2011, Digilent Inc.		Author: GH
SHEET: License Agreement		Rev: D.0
TITLE: chipKIT Max32		Doc#s: 500-202
Date: 1/11/2012		Sheets: 1/5



Engineers:	NEA
Copyright:	2011, Digilent Inc.
SHEET:	USB/UART, Power Supply
TITLE:	chipKIT Max32
Doc#:	500-202
Date:	1/11/2012
Sheet:	2/5

chipKIT Max32	ROHS CHINA
Digilent Inc.	chipKIT



<b>chipKIT Max32</b> Copyright 2011, Digilent Inc.	
SHEET: PIC32 I/O	Engineers: NEA
TITLE: chipKIT Max32	Author: GHA
Doc#: 900-202	Rev: D.0
Date: 1/11/2012	Sheet: 3/5

**Port D**

72	SD01/OC1/INT0/RD0
73	DC2/RD1
74	OC2/RD1
75	OC3/RD2
76	OC4/RD3
77	OC5/PHM0/CN13/RD4
78	PHM0/RD5
79	PHM0/RD5
80	ETXEN/PHM4/CN15/RD5
81	RTCC/END0/PHM0/CN15/RD5
82	ETXEN/PHM4/CN15/RD5
83	ETXEN/PHM4/CN15/RD5
84	RD7
85	END0Q
86	RD9
87	SS1/IC2/RD9
88	SS1/IC2/RD9
89	SS1/IC2/RD9
90	SS1/IC2/RD9
91	SS1/IC2/RD9
92	SS1/IC2/RD9
93	SS1/IC2/RD9
94	SS1/IC2/RD9
95	SS1/IC2/RD9
96	SS1/IC2/RD9
97	SS1/IC2/RD9
98	SS1/IC2/RD9
99	SS1/IC2/RD9
100	SS1/IC2/RD9
101	SS1/IC2/RD9
102	SS1/IC2/RD9
103	SS1/IC2/RD9
104	SS1/IC2/RD9
105	SS1/IC2/RD9
106	SS1/IC2/RD9
107	SS1/IC2/RD9
108	SS1/IC2/RD9
109	SS1/IC2/RD9
110	SS1/IC2/RD9
111	SS1/IC2/RD9
112	SS1/IC2/RD9
113	SS1/IC2/RD9
114	SS1/IC2/RD9
115	SS1/IC2/RD9
116	SS1/IC2/RD9
117	SS1/IC2/RD9
118	SS1/IC2/RD9
119	SS1/IC2/RD9
120	SS1/IC2/RD9
121	SS1/IC2/RD9
122	SS1/IC2/RD9
123	SS1/IC2/RD9
124	SS1/IC2/RD9
125	SS1/IC2/RD9
126	SS1/IC2/RD9
127	SS1/IC2/RD9
128	SS1/IC2/RD9
129	SS1/IC2/RD9
130	SS1/IC2/RD9
131	SS1/IC2/RD9
132	SS1/IC2/RD9
133	SS1/IC2/RD9
134	SS1/IC2/RD9
135	SS1/IC2/RD9
136	SS1/IC2/RD9
137	SS1/IC2/RD9
138	SS1/IC2/RD9
139	SS1/IC2/RD9
140	SS1/IC2/RD9
141	SS1/IC2/RD9
142	SS1/IC2/RD9
143	SS1/IC2/RD9
144	SS1/IC2/RD9
145	SS1/IC2/RD9
146	SS1/IC2/RD9
147	SS1/IC2/RD9
148	SS1/IC2/RD9
149	SS1/IC2/RD9
150	SS1/IC2/RD9
151	SS1/IC2/RD9
152	SS1/IC2/RD9
153	SS1/IC2/RD9
154	SS1/IC2/RD9
155	SS1/IC2/RD9
156	SS1/IC2/RD9
157	SS1/IC2/RD9
158	SS1/IC2/RD9
159	SS1/IC2/RD9
160	SS1/IC2/RD9
161	SS1/IC2/RD9
162	SS1/IC2/RD9
163	SS1/IC2/RD9
164	SS1/IC2/RD9
165	SS1/IC2/RD9
166	SS1/IC2/RD9
167	SS1/IC2/RD9
168	SS1/IC2/RD9
169	SS1/IC2/RD9
170	SS1/IC2/RD9
171	SS1/IC2/RD9
172	SS1/IC2/RD9
173	SS1/IC2/RD9
174	SS1/IC2/RD9
175	SS1/IC2/RD9
176	SS1/IC2/RD9
177	SS1/IC2/RD9
178	SS1/IC2/RD9
179	SS1/IC2/RD9
180	SS1/IC2/RD9
181	SS1/IC2/RD9
182	SS1/IC2/RD9
183	SS1/IC2/RD9
184	SS1/IC2/RD9
185	SS1/IC2/RD9
186	SS1/IC2/RD9
187	SS1/IC2/RD9
188	SS1/IC2/RD9
189	SS1/IC2/RD9
190	SS1/IC2/RD9
191	SS1/IC2/RD9
192	SS1/IC2/RD9
193	SS1/IC2/RD9
194	SS1/IC2/RD9
195	SS1/IC2/RD9
196	SS1/IC2/RD9
197	SS1/IC2/RD9
198	SS1/IC2/RD9
199	SS1/IC2/RD9
200	SS1/IC2/RD9
201	SS1/IC2/RD9
202	SS1/IC2/RD9
203	SS1/IC2/RD9
204	SS1/IC2/RD9
205	SS1/IC2/RD9
206	SS1/IC2/RD9
207	SS1/IC2/RD9
208	SS1/IC2/RD9
209	SS1/IC2/RD9
210	SS1/IC2/RD9
211	SS1/IC2/RD9
212	SS1/IC2/RD9
213	SS1/IC2/RD9
214	SS1/IC2/RD9
215	SS1/IC2/RD9
216	SS1/IC2/RD9
217	SS1/IC2/RD9
218	SS1/IC2/RD9
219	SS1/IC2/RD9
220	SS1/IC2/RD9
221	SS1/IC2/RD9
222	SS1/IC2/RD9
223	SS1/IC2/RD9
224	SS1/IC2/RD9
225	SS1/IC2/RD9
226	SS1/IC2/RD9
227	SS1/IC2/RD9
228	SS1/IC2/RD9
229	SS1/IC2/RD9
230	SS1/IC2/RD9
231	SS1/IC2/RD9
232	SS1/IC2/RD9
233	SS1/IC2/RD9
234	SS1/IC2/RD9
235	SS1/IC2/RD9
236	SS1/IC2/RD9
237	SS1/IC2/RD9
238	SS1/IC2/RD9
239	SS1/IC2/RD9
240	SS1/IC2/RD9
241	SS1/IC2/RD9
242	SS1/IC2/RD9
243	SS1/IC2/RD9
244	SS1/IC2/RD9
245	SS1/IC2/RD9
246	SS1/IC2/RD9
247	SS1/IC2/RD9
248	SS1/IC2/RD9
249	SS1/IC2/RD9
250	SS1/IC2/RD9
251	SS1/IC2/RD9
252	SS1/IC2/RD9
253	SS1/IC2/RD9
254	SS1/IC2/RD9
255	SS1/IC2/RD9
256	SS1/IC2/RD9
257	SS1/IC2/RD9
258	SS1/IC2/RD9
259	SS1/IC2/RD9
260	SS1/IC2/RD9
261	SS1/IC2/RD9
262	SS1/IC2/RD9
263	SS1/IC2/RD9
264	SS1/IC2/RD9
265	SS1/IC2/RD9
266	SS1/IC2/RD9
267	SS1/IC2/RD9
268	SS1/IC2/RD9
269	SS1/IC2/RD9
270	SS1/IC2/RD9
271	SS1/IC2/RD9
272	SS1/IC2/RD9
273	SS1/IC2/RD9
274	SS1/IC2/RD9
275	SS1/IC2/RD9
276	SS1/IC2/RD9
277	SS1/IC2/RD9
278	SS1/IC2/RD9
279	SS1/IC2/RD9
280	SS1/IC2/RD9
281	SS1/IC2/RD9
282	SS1/IC2/RD9
283	SS1/IC2/RD9
284	SS1/IC2/RD9
285	SS1/IC2/RD9
286	SS1/IC2/RD9
287	SS1/IC2/RD9
288	SS1/IC2/RD9
289	SS1/IC2/RD9
290	SS1/IC2/RD9
291	SS1/IC2/RD9
292	SS1/IC2/RD9
293	SS1/IC2/RD9
294	SS1/IC2/RD9
295	SS1/IC2/RD9
296	SS1/IC2/RD9
297	SS1/IC2/RD9
298	SS1/IC2/RD9
299	SS1/IC2/RD9
300	SS1/IC2/RD9

IC2PD

**Port F**

87	ETXD1
88	ETXD1
89	ETXD1
90	ETXD1
91	ETXD1
92	ETXD1
93	ETXD1
94	ETXD1
95	ETXD1
96	ETXD1
97	ETXD1
98	ETXD1
99	ETXD1
100	ETXD1
101	ETXD1
102	ETXD1
103	ETXD1
104	ETXD1
105	ETXD1
106	ETXD1
107	ETXD1
108	ETXD1
109	ETXD1
110	ETXD1
111	ETXD1
112	ETXD1
113	ETXD1
114	ETXD1
115	ETXD1
116	ETXD1
117	ETXD1
118	ETXD1
119	ETXD1
120	ETXD1
121	ETXD1
122	ETXD1
123	ETXD1
124	ETXD1
125	ETXD1
126	ETXD1
127	ETXD1
128	ETXD1
129	ETXD1
130	ETXD1
131	ETXD1
132	ETXD1
133	ETXD1
134	ETXD1
135	ETXD1
136	ETXD1
137	ETXD1
138	ETXD1
139	ETXD1
140	ETXD1
141	ETXD1
142	ETXD1
143	ETXD1
144	ETXD1
145	ETXD1
146	ETXD1
147	ETXD1
148	ETXD1
149	ETXD1
150	ETXD1
151	ETXD1
152	ETXD1
153	ETXD1
154	ETXD1
155	ETXD1
156	ETXD1
157	ETXD1
158	ETXD1
159	ETXD1
160	ETXD1
161	ETXD1
162	ETXD1
163	ETXD1
164	ETXD1
165	ETXD1
166	ETXD1
167	ETXD1
168	ETXD1
169	ETXD1
170	ETXD1
171	ETXD1
172	ETXD1
173	ETXD1
174	ETXD1
175	ETXD1
176	ETXD1
177	ETXD1
178	ETXD1
179	ETXD1
180	ETXD1
181	ETXD1
182	ETXD1
183	ETXD1
184	ETXD1
185	ETXD1
186	ETXD1
187	ETXD1
188	ETXD1
189	ETXD1
190	ETXD1
191	ETXD1
192	ETXD1
193	ETXD1
194	ETXD1
195	ETXD1
196	ETXD1
197	ETXD1
198	ETXD1
199	ETXD1
200	ETXD1

IC2PF

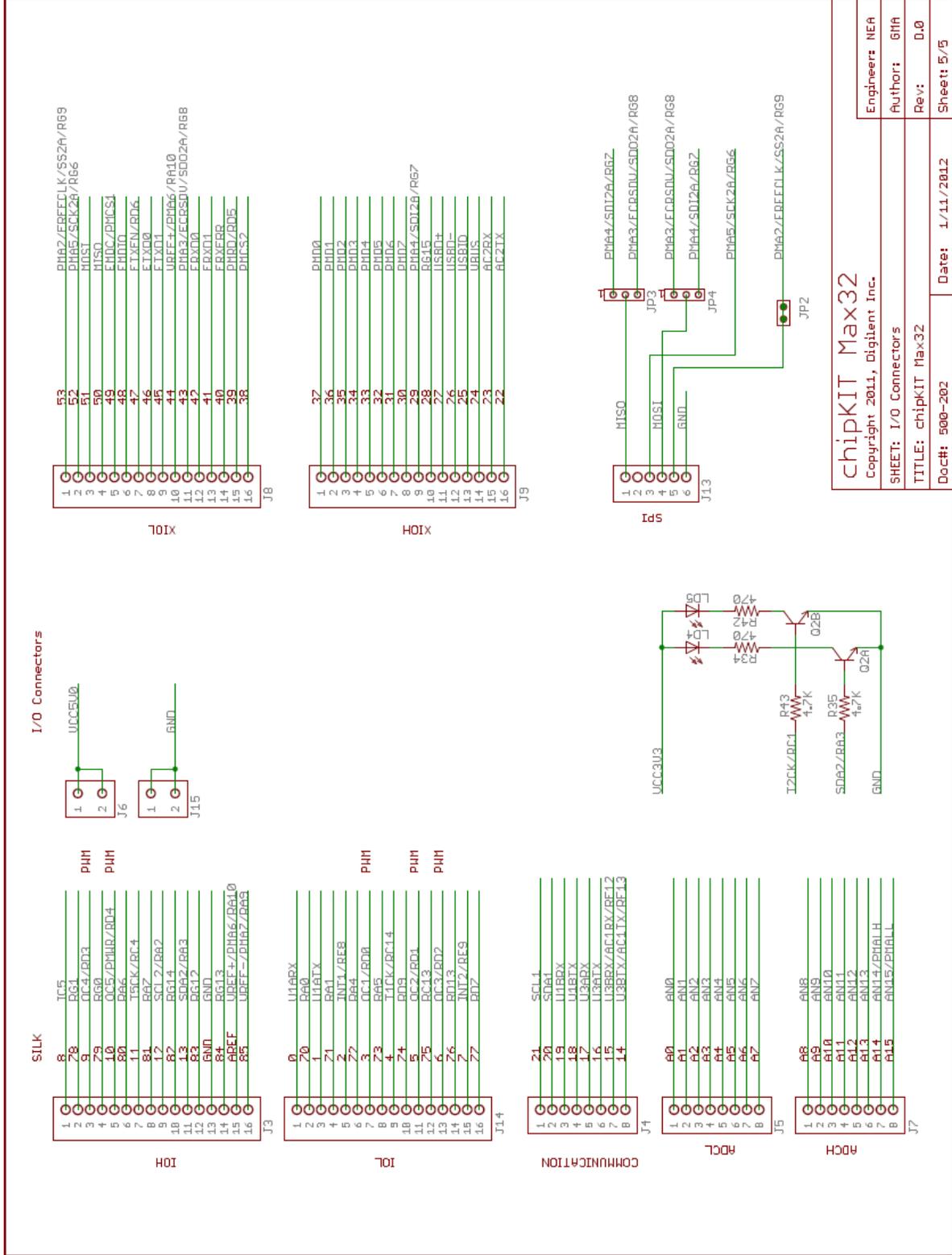
**Port E**

83	PHD0
84	PHD1
85	PHD2
86	PHD3
87	PHD4
88	PHD5
89	PHD6
90	PHD7
91	PHD8
92	PHD9
93	PHD10
94	PHD11
95	PHD12
96	PHD13
97	PHD14
98	PHD15
99	PHD16
100	PHD17
101	PHD18
102	PHD19
103	PHD20
104	PHD21
105	PHD22
106	PHD23
107	PHD24
108	PHD25
109	PHD26
110	PHD27
111	PHD28
112	PHD29
113	PHD30
114	PHD31
115	PHD32
116	PHD33
117	PHD34
118	PHD35
119	PHD36
120	PHD37
121	PHD38
122	PHD39
123	PHD40
124	PHD41
125	PHD42
126	PHD43
127	PHD44
128	PHD45
129	PHD46
130	PHD47
131	PHD48
132	PHD49
133	PHD50
134	PHD51
135	PHD52
136	PHD53
137	PHD54
138	PHD55
139	PHD56
140	PHD57
141	PHD58
142	PHD59
143	PHD60
144	PHD61
145	PHD62
146	PHD63
147	PHD64
148	PHD65
149	PHD66
150	PHD67
151	PHD68
152	PHD69
153	PHD70
154	PHD71
155	PHD72
156	PHD73
157	PHD74
158	PHD75
159	PHD76
160	PHD77
161	PHD78
162	PHD79
163	PHD80
164	PHD81
165	PHD82
166	PHD83
167	PHD84
168	PHD85
169	PHD86
170	PHD87
171	PHD88
172	PHD89
173	PHD90
174	PHD91
175	PHD92
176	PHD93
177	PHD94
178	PHD95
179	PHD96
180	PHD97
181	PHD98
182	PHD99
183	PHD100
184	PHD101
185	PHD102
186	PHD103
187	PHD104
188	PHD105
189	PHD106
190	PHD107
191	PHD108
192	PHD109
193	PHD110
194	PHD111
195	PHD112
196	PHD113
197	PHD114
198	PHD115
199	PHD116
200	PHD117

IC2PE

**Port 6**

89	C2PX/PHD8/R60
90	C2PX/PHD8/R60
91	C2PX/PHD8/R60
92	C2PX/PHD8/R60
93	C2PX/PHD8/R60
94	C2PX/PHD8/R60
95	C2PX/PHD8/R60
96	C2PX/PHD8/R60
97	C2PX/PHD8/R60
98	C2PX/PHD8/R60
99	C2PX/PHD8/R60
100	C2PX/PHD8/R60
101	C2PX/PHD8/R60
102	C2PX/PHD8/R60
103	C2PX/PHD8/R60
104	C2PX/PHD8/R60
105	C2PX/PHD8/R60
106	C2PX/PHD8/R60
107	C2PX/PHD8/R60
108	C2PX/PHD8/R60
109	C2PX/PHD8/R60
110	C2PX/PHD8/R60
111	C2PX/PHD8/R60
112	C2PX/PHD8/R60
113	C2PX/PHD8/R60
114	C2PX/PHD8/R60
115	C2PX/PHD8/R60
116	C2PX/PHD8/R60
117	C2PX/PHD8/R60
118	C2PX/PHD8/R60
119	C2PX/PHD8/R60
120	C2PX/PHD8/R60
121	C2PX/PHD8/R60
122	C2PX/PHD8/R60
123	C2PX/PHD8/R60
124	C2PX/PHD8/R60
125	C2PX/PHD8/R60
126	C2PX/PHD8/R60
127	C2PX/PHD8/R60
128	C2PX/PHD8/R60
129	C2PX/PHD8/R60
130	C2PX/PHD8/R60
131	C2PX/PHD8/R60
132	C2PX/PHD8/R60
133	C2PX/PHD8/R60



<b>chipKIT Max32</b>	
Copyright 2011, Digilent Inc.	
SHEET: I/O Connectors	Engineers: NEA
TITLE: chipKIT Max32	Author: GHA
Doc#: 500-202	Rev: D.0
Date: 1/11/2012	Sheet 5/5

# HC900 Specs

## Controller

<b>Function Blocks</b>	C50 CPU - 2000; C30 CPU - 400		
<b>Analog Inputs</b>	Up to 256 universal analog inputs		
<b>Accuracy</b>	±0.1% of span (field calibration to ±0.05% of span)		
<b>Analog Outputs</b>	Up to 64; user-specified span from 0 to 20 mA maximum, 12 bits, 0.1% Accuracy		
<b>Digital Inputs/Outputs</b>	Up to 960, contact DI, 24 Vdc DI/DO, 120 Vac DI/DO, 240 Vac DI/DO, relay DO		
<b>Total I/O</b>	Up to 960 combined analog and digital		
<b>I/O Racks per System</b>	One controller and up to 4 remote I/O racks		
<b>Control Loops</b>	C50 CPU (Up to 32); C30 CPU (Up to 8); cascade, ratio, %C, RH, dew point, three position step		
<b>Control Output Types</b>	Current, time-proportioning, position proportioning, three-position step		
<b>Setpoint Programmers</b>	Up to 8, 50 segments each, 16 event outputs, 99 profiles		
<b>Setpoint Scheduler</b>	Two: 50 segments, 8 ramp/soak outputs, 8 auxiliary outputs, 16 events, 10 schedules		
<b>Recipes</b>	Up to 50, 50 variables each		
<b>Comm.</b>	Ethernet 10baseT; Modbus/TCP protocol; up to 5 Ethernet hosts; up to 9 peer controllers; Serial Modbus RTU, RS485 or RS232, slave or master operation (up to 16 slaves)		
<b>Power Supply</b>	120 VAC to 240 VAC		
<b>Operating Temp.</b>	Rated: 0° to 140°F (0° to 60°C)		
<b>Humidity</b>	Rated: 10% RH to 90% RH, non-condensing		
<b>Rack Size (W)</b>	<b>4 Slot</b>	<b>8 Slot</b>	<b>12 Slot</b>
Inches:	10.5	16.5	22.5
Millimeters:	266.7	419.1	571.5

## Operator Interface

<b>Display</b>	Model 1042: 10.4 in. (264 mm); TFT Active Matrix Color LCD Model 559: 5.5 in. (140 mm); Color LCD	
<b>Distance from Controller</b>	Up to 2000 ft. (600 m)	
<b>Disk Drive or Zip Drive (1042)</b>	Data archiving and configuration, setpoint profile, recipe file transfer	
<b>Power Supply</b>	24 VDC	
<b>Size (WxHxD)</b>	<b>559</b>	<b>1042</b>
Inches:	9.40 x 6.15 x 5.40	15.80 x 9.80 x 7.20
Millimeters:	240 x 159 x 136	400 x 248 x 183
<b>Operating Temperature</b>	32° to 122°F (0° to 50°C)	32° to 113°F (0° to 45°C)
<b>Humidity (Non-condensing)</b>	Rated: 10 to 90% Extreme: 5 to 95%	
<b>Panel Rating:</b>	Type 12 or 4X	Type 4X

## Hybrid Control Designer Software

<b>Configuration</b>	Off-line, with run mode editing
<b>Operating Environment</b>	Windows Me, NT, 2000 or Xp
<b>PC</b>	Pentium, 200MHz with 64 MB RAM minimum SVGA or greater screen resolution
<b>Cable</b>	9-pin RS232 null modem cable to configuration port or Ethernet 10Base T (crossover)
<b>Modem Support</b>	Monitor, upload, download configuration

Honeywell offers a complete portfolio of products and solutions for process and machine control applications, including controllers, recorders, transmitters, actuators, smart sensors, and analytical instruments. To learn more about these offerings and how they can help your organization achieve breakthrough results, contact your local Honeywell representative, or contact us at the following phone numbers:



<b>U.S.A.:</b> 1-800-784-3011	<b>France:</b> 33 1 60 19 80 75
<b>Canada:</b> 1-800-461-0013	<b>Italy:</b> 39 02 9214 6503
<b>UK:</b> 44 1698 481730	<b>Spain:</b> 34 91313.61.00
<b>Germany:</b> 49 69 8064-336	<b>Latin America:</b> 1-305-805-8188
<b>Asia/Pacific:</b> 65 6355 2828	

For other locations, visit our website:  
<http://www.acs.honeywell.com>

## Warranty/Remedy

Honeywell warrants goods of its manufacture as being free of defective materials and faulty workmanship. Contact your local sales office for warranty information. If warranted goods are returned to Honeywell during the period of coverage, Honeywell will repair or replace without charge those items it finds defective.

**The foregoing is Buyer's sole remedy and is in lieu of all other warranties, expressed or implied, including those of merchantability and fitness for a particular purpose.**

Specifications may change without notice. The information we supply is believed to be accurate and reliable as of this printing, however we assume no responsibility for its use. While we provide application assistance personally, through our literature and the Honeywell web site, it is up to the customer to determine the suitability of the product in the application.

Copyright© by Honeywell International Inc., March 2004.



**Honeywell Process Solutions**  
www.acs.honeywell.com  
Honeywell Inc.  
2500 West Union Hills Drive  
Phoenix, AZ 85027  
51-52-58-23 06/04 KP10000