

UNIVERSIDAD DE PANAMÁ
FACULTAD DE INGENIERÍA EN INFORMÁTICA, ELECTRÓNICA Y COMUNICACIÓN
ESCUELA DE ING. ELECTRÓNICA Y COMUNICACIÓN
ARQUITECTURA DE COMPUTADORAS

Proyecto Final

Implementación del RS-485 con 8085 (Red Microcontroladora)

Desarrollado por los estudiantes:

Irina Ruiz F. PE-9-1266
Luis Carlos Wong 8-742-2404

Ingrid De Obaldía 4-721-1101
Gerardo Castillo 8-735-93

Gina Villafañe 9-709-856
Jennifer Rego 8-741-458
Johnny Ng 8-724-1424

Yao Lao Kong E-759-53
Ameth Batista 9-711-71
Álvaro Osorio 9-712-1002

Nadja Benson 8-735-560
Salvador Cornejo 8-736-1146

A Consideración del Profesor:

José Luis Palacios

Fecha de Entrega: 30 de julio de 2001

Objetivos

- ❑ Construir una red utilizando el standar RS-485 en la plataforma de desarrollo JLP8085.
- ❑ Implementar la Red como un sistema de microcontroladores.
- ❑ Elaboración de un Protocolo de Red que permita intercomunicar 2 o más microprocesadores.
- ❑ Escribir los programas de funcionamiento y subrutinas del Protocolo en lenguaje Assembler.

Introducción

La creación de una red entre computadoras surge de la necesidad de intercambiar información, de enviar y recibir instrucciones, ejecutar acciones o simplemente comunicarse.

De esta manera, las terminales en las redes actuan como controladores individuales de información, capaces de interactuar entre ellas y con el mundo exterior.

Nuestro proyecto se enfoca en la implementación del microprocesador INTEL 8085, el cual es montado sistemáticamente en una plataforma de desarrollo que permite su portabilidad y por lo tanto su aplicación en el diseño de redes microcontroladoras multipropósito.

Descripción General

En una red cada microcontrolador actúa independiente y puede ser utilizado para ejecutar programas que permitan intercambiar información entre ellas, interactuar con el usuario, monitorear condiciones externas (como temperatura, humedad, intensidad de la luz) u obtener datos específicos sobre el entorno.

A continuación, describimos algunos temas relacionados al proyecto a fin de facilitar la comprensión de este informe:

❑ **La Plataforma JLP8085**

El diseño de una red microcontroladora es uno de los proyectos más exigentes en el ámbito de las microcomputadoras. Utilizamos la JLP8085 como punto de partida para realizar este tipo de aplicaciones.

Algunas técnicas empleadas para su montaje fueron: Soldadura de precisión con estaño, ubicación adecuada de los componentes, confección del cable serial de datos y la puesta en marcha de la tarjeta.

La plataforma consta de un bus de Direcciones de 8 bits, una interfaz de datos, un microprocesador, una memoria RAM 62256 (32K x 8), una EPROM 2764 (8K x 8), y "chips de apoyo" de los que se sirve el procesador para delegar funciones.

Estos chips de apoyo (8251, 8254, 8255) son los responsables de procesos tales como el flujo de información a través de la circuitería interna y de controlar el flujo de información de uno a otro nodo en nuestra red.

❑ **El Microprocesador INTEL 8085:**

El 8085 es un microprocesador de 8 bits. Esto quiere decir entre otras cosas, que el procesador va a manipular en una sola operación datos de hasta 8 bits. Cuando se transfiera datos a la memoria o los traiga desde ella, lo podrá hacer de 8 bits en 8 bits. Aquí juega un papel decisivo el BUS de datos, ya que es por donde circulan los datos en las transferencias de la red de los microcontroladores.

El microprocesador cuenta con una serie de registros usados para realizar las operaciones de cálculo, y como almacenamiento de datos. Para hacer una idea más clara, un registro del 8085 es como una zona de memoria dentro del microprocesador donde se puede almacenar información. De esta forma el acceso es casi instantáneo.

□ Programación Periférica:

Se logra haciendo uso del lenguaje "*assembler*" y del software PROCOMM PLUS (Versión 1.1B). Para ello es necesario tener un conocimiento más o menos profundo acerca de las peculiaridades de la máquina que vamos a programar. Esto es obvio, ya que el ensamblador es un lenguaje de bajo nivel que nos permite un control total del sistema con que estamos trabajando.

Diseño de la Red

Para establecer una red entre dos o más computadoras, se necesitan de tres elementos básicos: Una *Guía física* por la cual viajarán los datos, *Interfaces con el hardware* para conectar los dispositivos o nodos que guíen las comunicaciones y por último pero no menos importante... el *Protocolo de Software* que habilita a los nodos para que comprendan la información que lleva la red.

La Guía física:

El microcontrolador que recibe la información captura los bits en la medida en que llegan y los reconvierte en data paralela para su procesamiento.

La transmisión se lleva a cabo enviando los bits uno a uno a través de un cable telefónico de 2 pares (3 de 4 alambres): uno para transmisión (Tx), otro para recepción (Rx) y un tercer alambre para el *ground* (*Véase esquemático*). El uso de este cable favorece la transmisión de señales analógicas o digitales.

Además, se utilizaron cajillas y conectores telefónicos de 2 a 1 para implementar la topología de red desde la placa de prueba (*protoboard* en que se armó la circuitería).

Interfáz RS-485 con el Hardware:

El RS-485 es una *interface* estándar ajustada especialmente para las redes de microcomputadoras. En este proyecto la misma:

- ❑ Permite, arriba de 32 receptores y transmisores pero sólo un transmisor a la vez
- ❑ Conecta cada nodo a la red por medio de esta interface serial a una velocidad de 2400bps.
- ❑ Utiliza la Transmisión Diferencial que no es más que la aplicación de dos señales de polaridad opuesta, para que el sistema sea más inmune a los ruidos y permita la transmisión a grandes distancias.
- ❑ Requiere de una fuente de voltaje de +5V
- ❑ Requiere del **75176** (*Transceiver Chip*) para cumplir con las especificaciones del diseño. Este chip de 8 pines, convierte entre 5 voltios lógicos y las señales diferenciales del RS-485, en una señal de habilitación de entrada para cada dirección de la transmisión (*Véase Hoja de Especificaciones del Fabricante*).
- ❑ Requiere que el **PCI 8251** sea conectado en la circuitería del nodo MASTER (*Véase esquemático*). Este chip funciona como registro de corrimiento. Toma los 8 bits en serie provenientes de la red y los manda en paralelo al microprocesador a través del bus de datos.

Protocolo de Software:

Una vez que los nodos se han interconectado, necesitan poder comprenderse entre sí. A este lenguaje común se le llama el Protocolo de una red.

Para que los paquetes de datos puedan viajar desde el origen hasta su destino a través de la red, el protocolo debe definirse como direcciones en los mapas de memoria de los microprocesadores (codificadas en lenguaje assembler).

Los puntos que definen un Protocolo son:

- La Sintaxis: formato de los datos y niveles de señal .
- La Semántica: incluye información de control para la coordinación y manejo de errores.
- La Temporización: incluye la sincronización de velocidades y secuenciación.

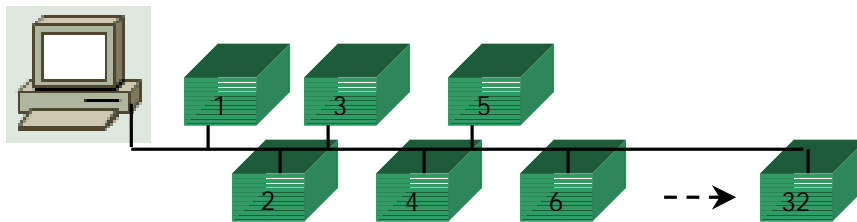
El nodo MASTER utiliza un protocolo para dirigir las terminales esclavas, así como los SLAVE utilizan un protocolo para poder entender al MASTER y responder.

Para crear un driver amigable al usuario, se puede adaptar el código del MASTER a el lenguaje C u otro más rápido y eficiente. Por su parte, los SLAVE pueden seguir utilizando el lenguaje *assembler* que se ha establecido por el Protocolo de la red.

Configuración de la Red

Se la utiliza **topología tipo BUS**. En la misma, todas las estaciones se encuentran conectadas directamente a través de interfaces físicas llamadas *tomas de conexión* a un medio de transmisión o BUS.

El diagrama de bloques que aparece a continuación muestra el "set" de nodos interconectados (uno al lado del otro) y cableados cada uno a la red.



Es necesario colocar terminales a cada extremo del BUS para que las señales no reboten y vuelvan al BUS. El último Slave que se coloque debe tener una resistencia de 120Ω (*Véase esquemático*).

Transmisión de Datos Digitales

En nuestro proyecto utilizamos la transmisión *half-duplex*. Logrando así, que las estaciones microcontroladoras se comuniquen por turnos. Esto es, los SLAVES sólo transmiten cuando se les pide (o sea cuando reconocen su *ID*).

Requerimientos

Para interconectar la red es necesario:

- Cable Telefónico de 2 pares, conectores RJ-11 y cajillas (de 1 a 1 y de 2 a 1).
- Placas de Prueba para montar la plataforma JLP8085 e incluir la circuitería adicional requerida por el estándar RS-485.
- Asegurarse que la red esté polarizada a un estado alto para prevenir señales no deseadas que puedan ser interpretadas como datos al inicio del proceso.
- Nuestra red puede albergar 32 nodos o terminales, controladas por un dispositivo MASTER, que puede ser una PC o bien una JLP8085.
- En caso deseable, si la red requiriese de más terminales, puede usarse un chip *Repetidor* como el 75177 o 75178.
- La distancia del MASTER a el nodo SLAVE más lejano puede ser tan grande como 4000 pies.

El Protocolo 8085/2001

El programa controlador almacenado en cada nodo es capaz de realizar funciones como:

-Configurar el puerto serial:

Cada nodo se ocupa con sus propias tareas, teniendo a la interface de red como línea de transmisión.

-Detectar y Procesar mensajes de entrada:

Se logra programando las interrupciones en los Slaves y por medio de rutinas en el Master.

-Asegurar que el nodo no trate de transmitir cuando la red esté ocupada:

A esta técnica se le llama *Control de Flujo*, y se usa para que el nodo emisor no sobrecargue al receptor al enviarle más datos de los que pueda procesar.

Cada nodo en la red tiene un buffer de una cierta capacidad para ir guardando los datos recibidos y tras procesarlos, enviarlos a *capas superiores*.

-Ocuparse de errores de transmisión y otros problemas:

Cada nodo detecta lo que un mensaje significa para él. Si es para otro nodo, el mensaje es ignorado, de otra manera el nodo interpreta y actúa sobre el mensaje.

Una forma eficiente de identificar el nodo objeto, es utilizando un *formato standar de mensajes*. Este formato incluye información sobre direcciones de memoria específicas como: el nodo que la envía, comprobación de errores en la longitud del mensaje, etc.

Framing / Entramado

La trama genérica para el Protocolo 8085/2001 es única. La misma contiene secciones denominadas campos, y cada campo está formado por bytes.

Los nombres de los *campos* son los siguientes:

- campo de inicio de trama
- campo de dirección
- campo de longitud/tipo/control
- campo de datos
- campo de fin de trama

Formato de Trama Genérica del MASTER:



Protocolo de Transmisión

Es una cadena de caracteres que debe ir en cada comando del Protocolo cuando el Master efectúa una transmisión. Comprende una cantidad específica de bytes que deben escribirse en mayúscula de acuerdo con las siguientes reglas:

- ❑ **Header:** se debe poner XT (2 bytes)
- ❑ **Slave ID:** es la identificación del Slave, puede ser A,B,C,D,E,...etc.
- ❑ **Comando de Lectura/Escritura:** se pone R para lectura y W para escritura.
- ❑ **Ubicación:** puede ser Puerto A, B, C, o M (para indicar memoria).
- ❑ **Parámetro:** dependiendo del comando y la ubicación puede tomar las siguientes formas:
 - ◆ Lectura a puerto: no necesita el parámetro.
 - ◆ Lectura a memoria: ubicación de memoria.
 - ◆ Escritura a puerto: es el dato en hexadecimal.
 - ◆ Escritura a memoria: memoria y dato a escribir.
- ❑ **Footer:** se debe poner XX para cerrar el *entramado*.

Ejemplos de instrucciones de el MASTER:

XTAWA40XX	Slave A <i>escriba</i> en el puerto A la dirección 40 hexadecimal
XTBWME0507FXX	Slave B <i>escriba</i> en la ubicación E050 el dato 7F
XTCRBXX	Slave C <i>lea</i> el puerto B
XTDRMC060XX	Slave D <i>lea</i> la ubicación de memoria C060

Formato de Trama Genérica de los SLAVE:



Protocolo de Recepción

Es la forma mediante la cual los SLAVE regresan un mensaje de *acknowledge* al MASTER transmitiéndole su respuesta a la trama recibida.

Esta información es decodificada siguiendo un entramado específico:

- ❑ **Header:** XS (2 bytes)
- ❑ **Slave ID:** la identificación del Slave, puede ser A,B,C,D,E,...etc.
- ❑ **Tipo de Respuesta:** dependiendo del comando del MASTER puede tomar las siguientes formas:
 - ◆ Error: detecta error en el entramado, se especifica con el número 1
 - ◆ Data: responde con un dato, se especifica con el número 2
 - ◆ O.K.: acción satisfactoria, se especifica con el número 3
- ❑ **Data:** especifica el dato en hexadecimal, aparece precedida del número 2
- ❑ **Footer:** XX cierra el *entramado*.

Ejemplos de respuesta de los SLAVE:

XSC1XX	Slave C detectó un error !!
XSD3XX	Slave D responde O.K.
XSA23FXX	Slave A responde con el dato 3F

Conclusiones

Una red se convierte en una solución cuando es necesario enviar y recibir información compleja a través de cierta distancia.

Los dispositivos microcontroladores de este proyecto pueden comunicarse entre sí e interactuar por medio de una red de topología tipo BUS que permite 32 terminales como máximo. Esta topología es aceptada favorablemente por el standar RS-485.

El protocolo de la red fue diseñado en un lenguaje de bajo nivel que permitiese la comunicación entre los circuitos. Las principales instrucciones ASM nos ayudaron a programar los microcontroladores y obtener los resultados esperados.

Un número de identificación o *ID* le fue asignado a cada nodo terminal. Este número funciona como una dirección global para identificar la terminal en el resto de la red.

Una vez establecido el protocolo, la red se comporta como si fuera una conexión directa entre las múltiples estaciones microcontroladoras, ahorrando así bastante lógica de control.

Entre nuestras recomendaciones está la implementación de esta red a un sistema controlador de parámetros como:

- ❑ Ejecución de acciones (activar mecanismos, controlar maquinarias, etc)
- ❑ Lectura y escritura en determinada dirección de memoria (transmisión de datos).

Todo esto es posible a largas distancias, y pueden desarrollarse futuros proyectos que implementen y mejoren el trabajo presentado.

Teoría Básica Utilizada

- ❑ Aplicación del curso *Arquitectura de Computadoras*
- ❑ Programación en Lenguaje "Assembler"
- ❑ Técnicas de programación de Modular
- ❑ Conocimientos básicos sobre Interfaces y Protocolo de Redes
- ❑ Confección de Diagrama Esquemático en ORCAD 9

Anexos

- ❑ Esquemático de la Interface RS-485
- ❑ Cronograma de Trabajo
- ❑ Hojas de Especificaciones
- ❑ Fotografías del Proyecto
- ❑ Código del Programa en Assembler

Citas Referenciales

- ❑ **Electrónica Práctica.** "Comprender los Microcontroladores". Artículo en pág 24. Revista Número 20. Madrid. 1998.
- ❑ **Página web de National Semiconductor.** www.national.com