



115-20

Instituto Tecnológico y de Estudios Superiores de Monterrey



TECNOLÓGICO
DE MONTERREY



TECNOLÓGICO
DE MONTERREY

BIBLIOTECA
Campus Ciudad de México

Campus Ciudad de México

División de Ingeniería y Arquitectura

Ingeniería en Electrónica y Comunicaciones

Departamento de Ingeniería Eléctrica y Electrónica

“Sistema de control de acceso con comunicación Ethernet.”

Autores:	José Antonio Llanos Torres	969220
	Leonardo Ramírez Torres	969789
Asesor:	M.C. Israel Macías Hidalgo	
Profesor:	Dr. Raúl Crespo Saucedo	

México D.F. a 9 de mayo de 2005.

EGIA. PROYE
TISIOS. 8. E83
253
2005

ymb10759736

I. INTRODUCCIÓN	1
I.1. Antecedentes	1
I.2. Definición del problema	3
I.3. Objetivos.....	5
<i>I.3.1. General.....</i>	<i>5</i>
<i>I.3.2. Específicos.....</i>	<i>6</i>
I.4. Justificación.....	7
I.5. Alcances y limitaciones.....	8
I.6. Metodología a seguir.....	11
II. MARCO TEÓRICO.....	11
II.1. Rabbit Semiconductor RCM3200.....	11
II.2. Redes de Datos y Ethernet.....	15
<i>II.2.1. El modelo OSI.....</i>	<i>15</i>
<i>II.2.2. Ethernet.....</i>	<i>17</i>
<i>II.2.3. Estructuras y Topologías Ethernet</i>	<i>18</i>
<i>II.2.4. Protocolos Ethernet</i>	<i>21</i>
<i>II.2.5. Ventajas de Ethernet (justificación de su elección).....</i>	<i>22</i>
II.3. Lectores de Tarjetas.....	23
II.4. Tarjetas y Lectoras de Código de Barras.....	24
<i>II.4.1. Códigos de Barras.....</i>	<i>24</i>
<i>II.4.2. Impresión del código de barras.....</i>	<i>25</i>
<i>II.4.3. Características de los códigos de barras.....</i>	<i>25</i>
<i>II.4.4. Tipos de Códigos de Barras.....</i>	<i>26</i>
<i>II.4.5. Lectoras de código de barras.....</i>	<i>27</i>
<i>II.4.6. Orientación a la identificación de personal.....</i>	<i>28</i>
<i>II.4.7. Decodificador de teclado.....</i>	<i>29</i>
<i>II.4.8. Láser de pistola.....</i>	<i>30</i>
II.5. Sistemas de Administración de Bases de Datos.....	30
II.6. Edificios Inteligentes.....	33
<i>II.6.1. Grados de inteligencia.....</i>	<i>34</i>

III. DESARROLLO DEL PROYECTO.....	35
III.1. Diseño.....	35
III.2. Construcción.....	37
<i>III.2.1. Adaptación de la señal proveniente de la lectora de código de barras al módulo Ethernet.....</i>	<i>37</i>
<i>III.2.2. Servidor de Base de Datos.....</i>	<i>41</i>
III.3. Pruebas.....	42
<i>III.3.1. Servidor de base de datos.....</i>	<i>42</i>
<i>III.3.2. Interfaz Ethernet</i>	<i>44</i>
<i>III.2.3. Adquisición de Datos.....</i>	<i>46</i>
III.4. Resultados.....	49
<i>III.4.1. Programa "Servidor" del sistema en una terminal remota.....</i>	<i>49</i>
<i>III.4.2. Programa "Cliente" en la Interfaz RCM3200</i>	<i>50</i>
<i>III.4.3. Programa de adaptación de la señal de la lectora de código de barras</i>	<i>50</i>
<i>III.4.4. Elaboración de una puerta y una base de datos.....</i>	<i>50</i>
<i>III.4.5. Análisis Financiero.....</i>	<i>52</i>
IV. CONCLUSIONES.....	56
IV.1. Perspectivas y trabajo a futuro.....	56
IV.2. Comentarios.....	58
V. REFERENCIAS.....	62
VI. ANEXOS.....	64
VI.1. Programa de lectura de datos del lector de código de barras.....	65
VI.2 Programa Cliente de la Interfaz Ethernet.....	67
VI.3. Programa Terminal Remota – Interfaz.....	71
VI.4. Programa Central del Servidor de BDD.....	74
VI.5. Especificaciones Lector de Códigos de Barras MS6720 – KW.....	77
VI.6. Diagrama Esquemático del Módulo RCM3200.....	78
VII. PÓSTER.....	79

I. INTRODUCCIÓN

I.1. Antecedentes

Basta con mirar a nuestro alrededor para ver como la tecnología forma parte integra de nuestra vida cotidiana, desde simples aparatos en el hogar, como una lavadora que identifica que tipo de ropa se le introdujo y ella selecciona la temperatura del agua y el tiempo de lavado que tiene que realizar, hasta refrigeradores que nos dan la facilidad de conectarnos a Internet teniendo una pantalla donde podemos ver recetas y checar el clima a nivel mundial, más aún vemos con que facilidad podemos enviar un documento desde México hasta Japón por ejemplo en fracciones de segundos gracias al e-mail y al Internet.

Y qué decir de los nuevos edificios que están surgiendo gracias a los adelantos de la tecnología moderna. Esta tendencia se marcará aún más en el futuro. Los dispositivos electrónicos hacen el trabajo rutinario con más rapidez y facilidad, y a un menor costo que cualquier ser humano.

Existe una gran necesidad de ahorrar energía en nuestros días; la importancia de contar con una comunicación efectiva, clara y rápida; la seguridad, comodidad y confort de los trabajadores; la modularidad de los espacios y equipos, y la posibilidad de dar un mayor ciclo de vida a un edificio, han dado lugar al concepto de "edificios inteligentes".

Con estos adelantos tecnológicos, resulta imposible cerrar los ojos ante el futuro inmediato al que nos enfrentamos y mucho menos nosotros los profesionales de la ingeniería electrónica, que en cierta manera tenemos la responsabilidad de diseñar dispositivos que satisfagan las necesidades mencionadas anteriormente.

Una de las partes más importantes que conforman un edificio "inteligente" es aquella que se encarga de la apertura y cierre de puertas de los usuarios que por ahí transitan. El control de acceso con interfaz Ethernet, desarrollado en este proyecto, es un sistema de comunicación

entre un servidor de bases de datos y una estación remota que controla directamente el acceso a un área de trabajo previamente definida.

El Proyecto está basado en el sistema embebido RCM3200 RabbitCore® que presenta grandes ventajas en el manejo de datos, programación, capacidad y flexibilidad para lograr el alcance de nuestros objetivos.

Es seguro que muchos sistemas parecidos al nuestro han sido desarrollados ya por empresas para llevar un control del personal que labora en un lugar. Se encontraron diversas aplicaciones que van desde casos simples hasta sistemas muy robustos que son totalmente eficientes y flexibles. En un caso particular encontramos la siguiente tecnología *Moebius Identidad* que fue desarrollado por *Avant Technologies Argentina*™ y que es un sistema no muy complejo. Por medio de este software, se obtiene información precisa de los horarios de ingresos y egresos del personal ya sea en forma general o bien individual. Es compatible con lectores de huella dactilar y lectores de tarjetas con banda magnética. Pero este software está diseñado para ser instalado en cada punto de acceso y así verificar en una computadora central la base de datos de la nómina de la empresa [3].

Otro ejemplo que contrasta con el anterior es el desarrollado por AMTEL®. El Sistema de Control de Acceso de AMTEL® puede controlar desde una sola puerta hasta múltiples edificios en una ciudad o instalación. El mismo integra prácticamente todo tipo de enlaces de comunicación, siendo una plataforma versátil a la cual se le pueden adicionar múltiples módulos de administración. Es compatible con todas las tecnologías de lectoras, cuenta con la posibilidad

de interconectar tres tipos de paneles de control de acceso (1, 2 y 4 puntos de acceso) en el mismo lazo o red. También ofrece paneles de Control de Acceso totalmente funcionales e inteligentes que incluyen manejo de Zonas de tiempo, Niveles de Acceso, Control de Débito, Controles de Personal, Control de Dispositivos Externos a través de Entradas / salidas del panel, Control de AntiPassback, Lazos anidados y muchas cosas más [2]. Este sistema como se pudo ver es ampliamente robusto y flexible. Cabe mencionar que nuestro proyecto no presentará tantas opciones como el sistema anterior.

En nuestro caso, la innovación será transmitir vía Ethernet, a través de una interfaz, los datos obtenidos después de la validación de la tarjeta de usuario y recibir la autorización desde un programa de control montado sobre esta plataforma. Lo que omite el uso de una computadora en cada zona de acceso.

Se encontraron algunos sistemas muy parecidos a los antes mencionados con algunas características más y otras menos, pero en general ninguno opera con la interfaz que utilizamos, en este caso: RCM3200 RabbitCore™.

1.2. Definición del problema

En la actualidad los sistemas de control de acceso son usados en compañías y corporaciones vanguardistas. Debido a sus altos costos y su difícil implementación estos sistemas no fueron tan populares como se esperaba en un principio, pero con el paso del tiempo su uso se hace más y más necesario.

La creciente inseguridad, el ahorro en equipo y personal así como el manejo de información estadística son sólo algunas de las causas que dieron pie a la realización de este proyecto. El problema radica en mantener un control seguro del ingreso de usuarios o personal a un inmueble, edificio o zona de trabajo. En cada punto de acceso, generalmente, se cuenta con una computadora y un encargado que valida la identidad de la persona que desea ingresar, lo cual puede resultar engorroso para la empresa.

Otro de los problemas que se pretenden solucionar es: uso ineficiente de la energía, ya que los recursos que la generan en la mayoría de los casos no son renovables y resulta vital administrarlos adecuadamente, por lo que es importante que el dispositivo a diseñar funcione con un nivel energético menor al de sistemas similares.

La automatización se entiende como la capacidad de las máquinas para llevar a cabo determinadas tareas anteriormente efectuadas por seres humanos, y para controlar la secuencia de las operaciones sin intervención humana, lo cual además de contar con un buen nivel de sofisticación, beneficia a una empresa en la reducción de su nómina y a los mismos empleados que se encargaban de realizar tareas monótonas.

Supongamos que en una institución de investigación nuclear los alumnos sólo tienen acceso a ciertos laboratorios y salones, mientras que algunos maestros, pero no todos, tienen acceso ilimitado los laboratorios y salas de investigación disponibles en la institución, el personal de intendencia no puede acceder a ningún laboratorio y los directivos de la institución son los

únicos que puede entrar en la sala de sustancias radiactivas. El problema de mantener en cada acceso una computadora y un encargado resulta complejo.

Se busca, como solución, que en cada punto de ingreso exista un sistema automático que valide, registre y permita el ingreso a dicho sector a las personas previamente autorizadas, eliminando la necesidad de una computadora y su respectivo encargado.

I.3. Objetivos

I.3.1. General

Nuestra meta principal es desarrollar un sistema de control de acceso para validación, captura y registro de información de usuarios en un edificio, empresa o industria. Se pretende implementar un dispositivo electrónico que reciba un paquete de datos de una lectora de tarjetas de código de barras para la identificación de una persona que intenta ingresar a cierta área del edificio, el dispositivo cuenta con una interfaz Ethernet que evita la necesidad de tener una computadora en cada punto de acceso del edificio, pudiendo así, controlar el ingreso a todas las áreas de un edificio con una sola computadora.

El dispositivo procesa el paquete de datos que recibe de la lectora de tarjetas de identificación personal, la información es enviada vía Ethernet a una computadora que cuenta con una base de datos de los usuarios que tienen acceso al área a la que se intenta ingresar, se comparan dichos datos y en caso de que la persona esté autorizada se envía una respuesta a través de

Ethernet, la cual permite abrir o no el acceso; en este caso una puerta. Si el acceso ha sido negado se advertirá mediante un foco indicando alto. También se creará una base de datos que registre la hora de entrada y el nombre de la persona que entró a esa área, todos estos datos podrán consultarse desde el servidor central del sistema, si es que se cuenta con la autorización para ello.

1.3.2. Específicos

Nos hemos propuesto lograr que la tarjeta RCM3200 reciba e interprete los datos que le envía la lectora de códigos de barras, para esto será necesario acondicionar la señal de la lectora y elaborar un programa para el microprocesador del modulo RCM3200 y pueda almacenar dichos datos en forma de número en alguna locación de la memoria.

También el sistema deberá ser capaz de enviar mediante Ethernet esta información a un servidor de bases de datos, también robustecer esta base de datos y adicionar funciones de red que le permitan recibir un número, buscarlo entre sus registros y enviar una respuesta mediante Ethernet de autorización o rechazo de acceso.

Investigar y obtener datos sobre el beneficio económico que traería a una empresa implementar nuestro proyecto en sus instalaciones. Otros de nuestros objetivos son: que el sistema sea seguro y flexible, que permita llevar un control sobre los empleados de una empresa registrando de manera eficiente las horas de entrada, área accedida así como brindar seguridad a las diferentes áreas de un edificio.

Así mismo se espera que los beneficios del proyecto para la empresa que lo implemente sean: reducción de costos mediante la eliminación de equipo innecesario y relevación de personal de vigilancia, y que se garantice que sólo ingresa el personal autorizado en las distintas áreas que conforman el edificio.

I.4. Justificación

Este proyecto incursiona en las áreas de redes y comunicaciones, haciendo que los sistemas de control de acceso en edificios inteligentes sean mucho más sofisticados mediante el uso de la tecnología Ethernet, la capa física más popular de la tecnología LAN usada actualmente. Otros tipos de LAN incluyen Token Ring, Fast Ethernet, FDDI, ATM y LocalTalk. Ethernet es popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación. Estos puntos fuertes, combinados con la amplia aceptación en el mercado y la habilidad de soportar virtualmente todos los protocolos de red populares, hacen a Ethernet la tecnología ideal para la red de la mayoría los usuarios de la informática actual. La norma de Ethernet fue definida por el Instituto para los Ingenieros Eléctricos y Electrónicos (IEEE) como IEEE Standard 802.3. Adhiriéndose a la norma de IEEE, los equipos y protocolos de red pueden operar en conjunto eficazmente [1]. Al controlar el acceso a las áreas de un edificio mediante una interfaz Ethernet en cada puerta y un solo servidor de base de datos se ahorran costos en la adquisición de equipos y contratación de personal ya que evita la adquisición de una computadora por puerta o de un vigilante por puerta. Logrando un control de personal y de niveles de acceso eficiente y permitiendo a cualquier usuario autorizado consultar una base de

datos montada en Internet en la que pueda ver registros de entrada de personal en las diversas áreas, así como diferentes estadísticas de cada uno de los empleados.

Los beneficios del Sistema de Control de Acceso con Comunicación Ethernet son numerosos, entre ellos podemos mencionar los siguientes:

- Aprovechamiento de las redes existentes de las empresas.- Ya que las tecnologías de la información tienden al uso de redes de datos para la gestión de muchos de sus procesos, por lo que es natural esperar que ya cuenten con una infraestructura de red LAN adecuada. Debido a la existencia previa de estas instalaciones en la mayoría de las empresas o instituciones, la implementación del proyecto resulta muy sencilla ya que únicamente se requiere de un nodo de red por dispositivo.
- El proyecto constituye la base de un sistema completo de un edificio inteligente que permita la regularización, supervisión y el control del conjunto de las instalaciones eléctricas, de seguridad, informática y transporte, entre otras, en forma integrada y automatizada, con la finalidad de lograr una mayor eficacia operativa y, al mismo tiempo, un mayor confort y seguridad para el usuario, al satisfacer sus requerimientos.

I.5. Alcances y limitaciones

El ideal de nuestro proyecto contando con el tiempo y los recursos necesarios sería un sistema que incluya la elaboración de una puerta adecuada que tenga un actuador que la abra y cierre cuando un usuario autorizado ingrese, integrar una lectora de tarjetas de identificación

personal de cualquier tipo (magnética, código de barras o de chip), o de cualquier dispositivo que resulte útil para el control de acceso en edificios (como dispositivos de biometría; analizador de retina, analizador de huellas dactilares, etc.), para lograr esto tendríamos que programar el microprocesador de la interfaz Ethernet de forma que sea flexible para recibir los diferentes formatos en los que se presentan los datos que envían dichos dispositivos, así como acondicionar la señal entregada a la salida de los mismos a los niveles adecuados de corrientes y voltajes para operar en conjunto con la tarjeta utilizada en el proyecto.

También deberían resolverse ciertas vicisitudes como podrían ser que un usuario olvidara su tarjeta, o el acceso de visitantes al edificio y que el software de acceso pudiera ser manipulado por un operador que identifique a la persona y conceda el acceso, existen numerosas soluciones para estos detalles, sin embargo limitaremos los alcances del proyecto debido a la escasez de tiempo con el que se cuenta para desarrollarlo a plenitud, con la opción de que futuras generaciones lo retomen para mejorarlo o hacerlo más completo.

Los alcances iniciales del proyecto eran: elaborar el dispositivo electrónico que reciba la señal de la lectora de tarjetas de identificación personal, adquirir la interfaz Ethernet que cuenta con un microprocesador, programarla usando Dynamic C de forma que logre establecerse una transmisión / recepción de datos satisfactoria hacia el servidor remoto por medio de Ethernet y también lograr la transmisión / recepción en el sentido contrario (del servidor hacia el dispositivo).

De acuerdo a la experiencia adquirida en la primera fase del desarrollo del proyecto y a las sugerencias del asesor y los sinodales hemos decidido redefinir los alcances del proyecto de la siguiente forma:

El sistema únicamente recibirá datos de la lectora de códigos de barras Metrologic® modelo MS6720-47 KW, el uso de otro tipo de lectoras no garantiza el correcto funcionamiento del sistema, se elaborará todo el software necesario para operarlo, conformado por: un programa que decodifica bits recibidos en forma serial síncrona a código ASCII, los almacene y envíe a un servidor remoto, un programa manejador de la base de datos de usuarios del edificio que cuente con un servidor capaz de comunicarse a través de Ethernet, un programa cliente que reciba las señales de acceso concedido o negado, la propia base de datos, y un programa que genere una segunda base de datos con las estadísticas de acceso a un edificio.

La puerta elaborada para el proyecto tiene fines únicamente de simulación, cuenta con un electroimán que es activado por un relevador cuando recibe la indicación correspondiente proveniente de una interfaz Ethernet, además usaremos LED's que indiquen si un usuario tiene acceso o no al área a la que desea ingresar. El acoplamiento de una puerta que garantice la seguridad y el ingreso de solo una persona autorizada a la vez, es fuertemente sugerida tomando en cuenta las necesidades y a la capacidad económica de la empresa donde se implemente el proyecto.

También se elaboró una investigación de un caso real de impacto económico en una empresa si llegara a implementar nuestro proyecto en sus instalaciones.

I.6. Metodología a seguir

Inicialmente se planteó el problema a resolver y se realizó una valoración cualitativa de los beneficios que traería implementar un proyecto de esta naturaleza. El segundo paso fue la definición de objetivos y metas que deberían ser alcanzadas para que el proyecto fuera confiable y factible. A continuación se realizó la investigación de la base teórica que debía respetarse para el debido funcionamiento del proyecto. Después se inicio la búsqueda de alternativas de hardware y software que pudieran ser de utilidad en la solución del problema, seleccionado así la interfaz RCM3200 RabbitCore®, la lectora de códigos de barras Metrologic® modelo MS6720-47 KW para la parte de hardware, Dynamic C® y Visual Basic 6.0® para la parte de software. El siguiente paso fue la familiarización con estos dispositivos y lenguajes de programación y el desarrollo formal del sistema. Finalmente se evaluarán los resultados obtenidos y se realizará la presentación final del proyecto.

II. MARCO TEÓRICO

II.1. Rabbit Semiconductor RCM3200

Hardware. El módulo RCM3200 programable en C, con Ethernet contiene el microprocesador Rabbit 3000 que es una mejora derivada del ZiLOG Z80. El microprocesador Rabbit 3000 tiene siete puertos I/O de 8 bits. Muchos de los bits pueden tener funciones especiales, incluyendo seis puertos seriales para comunicación sincrónica y asíncrona, un puerto paralelo bidireccional, dos canales de captura de entrada y cuatro salidas PWM.

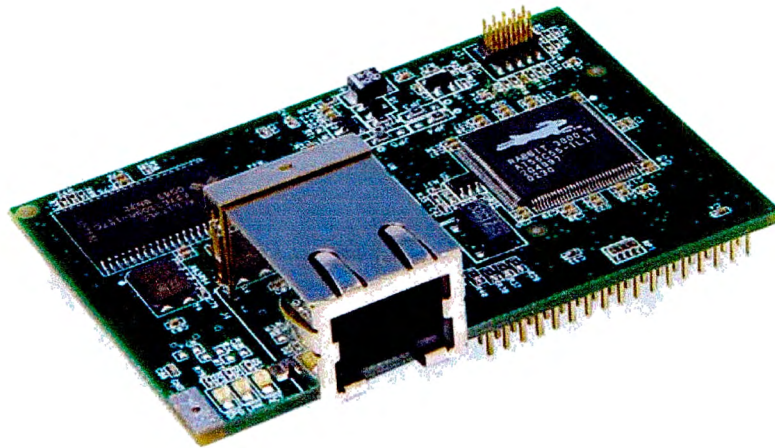


Figura 1. RCM3200 RabbitCore Modelo RCM3220 ®

Aparte de los puertos I/O, tiene un bus de memoria externo de 8 bits y 20 líneas de direcciones. La fuente de poder puede entregar de 3.6V a 1.8V. Un contador que funciona como un reloj de tiempo real tiene un pin de suministro de poder separado para proveerla de una reserva de batería.

Cuenta con 512 kilobytes de memoria Flash para almacenar programas, 512 kilobytes de RAM rápida para cargar códigos para ejecución, y 256 kilobytes de RAM para almacenar datos. Uno de los puertos seriales usa un cable especial de programación para cargar programas de una PC a la memoria Flash o RAM.

El controlador Ethernet del modulo es el "Local Bus Ethernet Controller ASIX AS88796 3-in-1", que interactúa con el bus de datos externo del CPU. El módulo tiene un conector RJ-45 para los sistemas Ethernet 10BASE-T y 100BASE-TX. Cuenta con dos headers para proporcionar acceso a los bits de I/O y otras señales.

El kit de desarrollo RCM3200 incluye un módulo RCM3200 y una tarjeta de diseño con un conector a la fuente de poder, un regulador de voltaje, un área tipo protoboard, switches y LED's para experimentación.

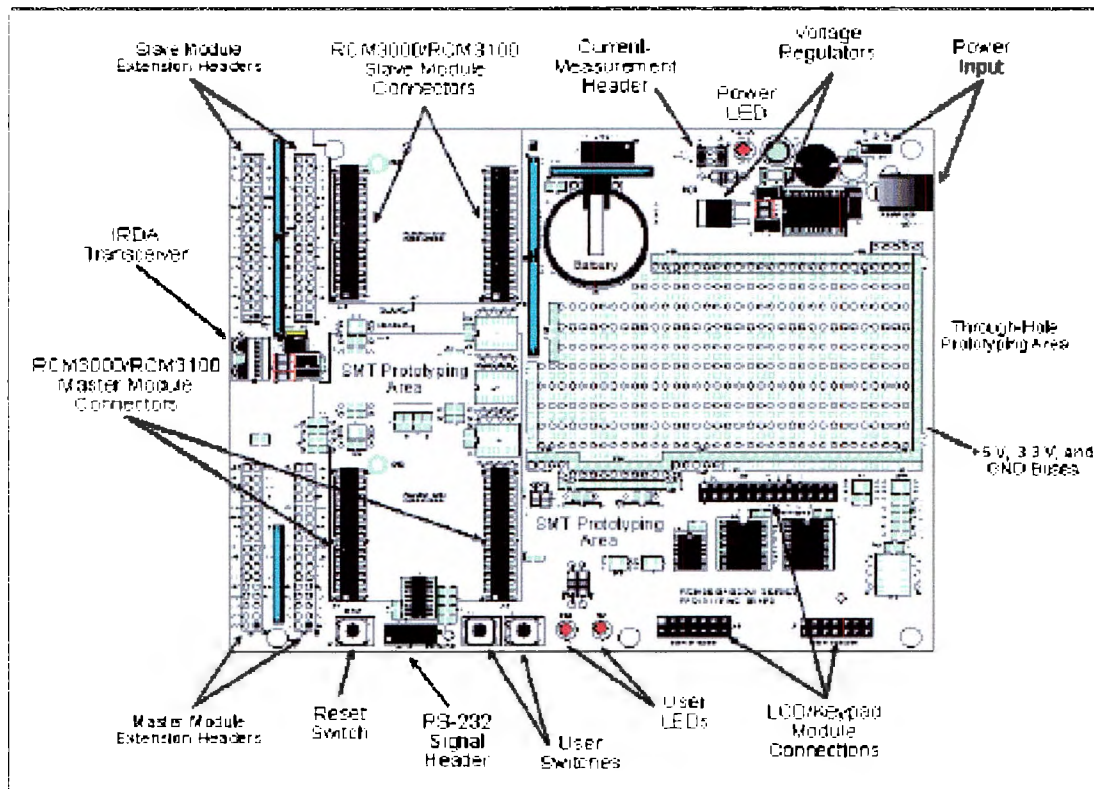
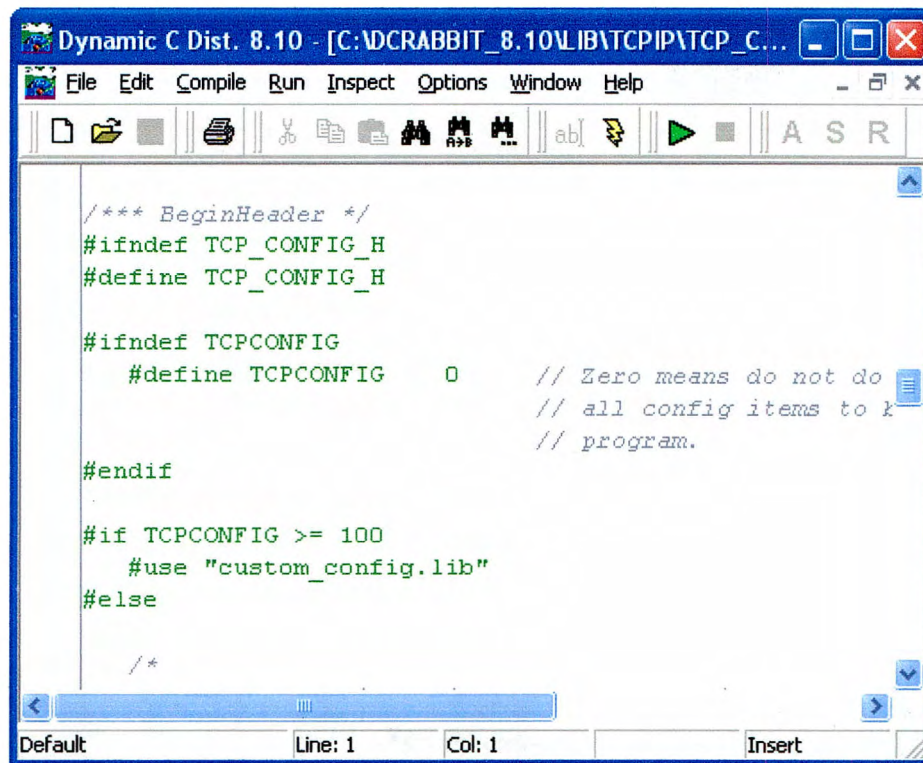


Figura 2. Área de la tarjeta de diseño del módulo RCM 3200

Software. Dynamic C de Rabbit Semiconductors® es un ambiente completo para escribir y editar código, compilar y unir, así como cargar el código ya compilado en la memoria RAM o Flash del módulo RCM3200.



```

Dynamic C Dist. 8.10 - [C:\DCRABBIT_8.10\LIB\TCP\TCP_C...
File Edit Compile Run Inspect Options Window Help
[Icons] [ab] [Run] [Compile] [Debug] [Find] [Save] [Print] [Copy] [Paste] [Undo] [Redo] [Home] [End] [A] [S] [R]

/**/ BeginHeader */
#ifndef TCP_CONFIG_H
#define TCP_CONFIG_H

#ifndef TCPCONFIG
#define TCPCONFIG 0 // Zero means do not do
// all config items to k
// program.
#endif

#if TCPCONFIG >= 100
#use "custom_config.lib"
#else

/*
Default Line: 1 Col: 1 Insert

```

Figura 3. Entorno del lenguaje de programación Dinamic C

Para trabajo en red el Dynamic C incluye drivers para el controlador Ethernet y librerías que soportan el protocolo de comunicación TCP/IP. Las librerías proveen soporte para un servidor HTTP, un cliente y un servidor FTP, y para mandar y recibir e-mail mediante SMTP y POP3. Un sistema de archivos soporta el almacenamiento de información en archivos en la memoria Flash.

La documentación para el manejo de Dynamic C y los módulos de Hardware incluye una serie de manuales detallados que pueden encontrarse en la página web de Rabbit Semiconductors®.[4]

II.2. Redes de Datos y Ethernet

II.2.1. El modelo OSI

Aunque en la actualidad nos parezca sencillo, conectar en red dos equipos, es un complicado problema de ingeniería. Cuando se abordan problemas de esta magnitud, la forma de solucionarlos suele ser dividir el problema grande en problemas pequeños. Esto es lo que propone el modelo de redes OSI (Open Systems Interconnection), publicado por la organización internacional ISO, con lo que se pretende que los nodos que conforman una red, incluso de diferentes fabricantes, sean capaces de establecer comunicación sin problema. El protocolo de comunicaciones se constituye en un estándar de software, que es el encargado de controlar la comunicación entre dos estaciones, el término "estándar" se refiere a que ese elemento debe cumplir determinados requerimientos, los cuales son impuestos por OSI. Este modelo divide el "gran problema" en 7 pequeños problemas a los que se conoce como los siete niveles de red OSI.

1. *Nivel Físico*: Conecta físicamente a dos transmisores. Define el medio de comunicación con el mecanismo de transmisión y el elemento de hardware. El nivel físico viene a ser básicamente el cable, que permite la comunicación, y transmisión de datos, y que define la transmisión de bits a través de un canal. En esta capa se tratan conceptos mecánicos eléctricos y procedimientos de interfase así como el medio de transmisión (par trenzado, cable coaxial y fibra óptica).

2. *Nivel de Datos*: Controla posibles errores entre dos puntos. Valida la integridad de los datos que pasan de un nodo a otro controlando su flujo. Trata de detectar y corregir los errores. Contiene información como: Número de caracteres, donde un campo de encabezamiento guarda el número, pero si el número es cambiado, será muy difícil recuperarlo. Caracteres de inicio y fin.
3. *Nivel de Red*: Encamina la información a través de la red. Define protocolos de enrutamiento de datos para asegurar que la información llegue al nodo correcto. Enruta los paquetes de la fuente al destino final a través de "routers" intermedios, tiene que saber la topología de la subred, evitar la congestión y maneja saltos de la fuente y el destino si se encuentran en redes distintas.
4. *Nivel de Transporte*: Propicia la comunicación entre dos puntos no adyacentes. Define protocolos para estructuración de mensajes, supervisa la transmisión y detecta errores. Provee un servicio eficiente y confiable. El hardware y el software dentro de este nivel de transporte se llaman "entidad de transporte", puede estar en el corazón del sistema operativo, en un programa, en una tarjeta, etc. Sus servicios son semejantes a los del nivel de red, al igual que las direcciones y el control de flujo.
5. *Nivel de Sesión*: Gestiona problemas ajenos a la comunicación. Coordina las comunicaciones y mantiene la sesión de comunicación el tiempo que sea necesario, controlando la seguridad, el ingreso de usuario y las tareas de administración.

6. *Nivel de Presentación*: Convierte la información. Define la manera como los datos se formatean, se presentan y se codifican.

7. *Nivel de Aplicación*: Proporciona servicios a la aplicaciones. Define la manera como interactúa la aplicación ejecutada con la red; incluye la administración de bases de datos, el correo electrónico y ciertos programas que emulan terminales. Este nivel es el más cercano al usuario. Es el programa o conjunto de programas que generan información para que esta viaje por la red, por ejemplo el correo electrónico, cuando lo procesamos y enviamos, este puede ir a cualquier lugar del mundo, y ser leído en cualquier tipo de ordenador. [9]

II.2.2. Ethernet

La idea original de Ethernet fue desarrollada experimentalmente cuando la corporación Xerox realizó trabajos de redes con cable coaxial, operando a una velocidad de 3 Mbps, en la década de los 70's. Para lograrlo fue usado el protocolo para LAN's CSMA/CD "Carrier Sense Multiple Access Collision Detect" que presentaba ocasionalmente grandes requerimientos de tráfico. El éxito de tal proyecto produjo que en 1980 se desarrollara la versión 10-Mbps Ethernet 1.0 por parte de tres compañías: Digital Equipment Corporation, Intel Corporation y Xerox Corporation.

El término Ethernet hace referencia a la familia de redes de área local "LAN" que son reguladas por el estándar 802.3 de la IEEE que define lo que comúnmente conocemos como protocolo

CSMA/CD. Son tres las velocidades de datos definidas por su desempeño sobre fibra óptica y cables de par trenzado.

1. 10 Mbps – 10Base-T Ethernet
2. 100 Mbps – Fast Ethernet
3. 1000 Mbps – Gigabit Ethernet
4. 10000 Mbps – 10-Gigabit Ethernet

II.2.3. Estructuras y Topologías Ethernet

Existen cuatro estructuras de interconexiones básicas principales, que determinan el tamaño y complejidad de una LAN.

1.- La primera estructura es la denominada *Punto a Punto*, donde dos unidades de la red se encuentran interconectadas y al cable que las conecta se le conoce como link de red. La longitud máxima del link depende del tipo de cable y del método de transmisión usado. La figura 4 muestra una conexión punto a punto.



Figura 4. Topología punto a punto

2.- La topología de bus se muestra en la figura 5 donde se usa un bus de estructura coaxial. Existen muchas limitaciones y restricciones en este tipo de red. La longitud de los segmentos está limitada a 500 metros y un poco más de 100 estaciones pueden ser conectadas a un segmento simple. Cada segmento también puede ser interconectado con repetidores siempre y cuando no existan múltiples caminos entre dos estaciones y que el número de equipos terminales de datos "DTE" no exceda de 1024. Un valor máximo predefinido también limita la distancia entre el par más distante de estaciones.

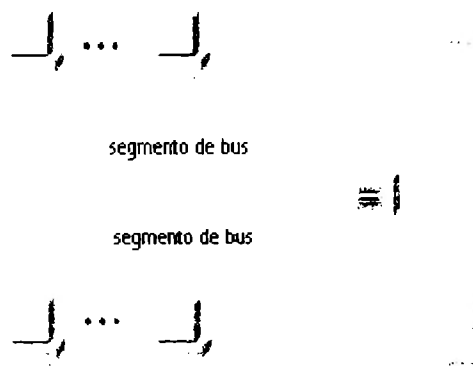


Figura 5. Topología de bus

3.- La tercera estructura de configuración de red es la topología estrella. Esta topología es la más popular y se muestra en la figura 6. La unidad central de la red actúa como un hub o un switch de red. Todas las conexiones en una topología de estrella son links punto a punto conectadas por fibra óptica o cable de par trenzado.

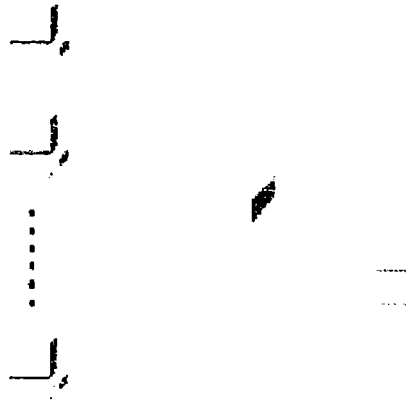


Figura 6. Topología de estrella

4.- Por último la topología "token ring" o de anillo. Las líneas de comunicación forman un camino cerrado. La información generalmente recorre el anillo en forma unidireccional, cada máquina recibe la información de la máquina previa, la analiza, y si no es para ella, la retransmite a la siguiente. Esta topología no es muy usada puesto que si una estación de trabajo falla, entonces la red se cae. En la figura 7 se puede ver una red de este tipo.

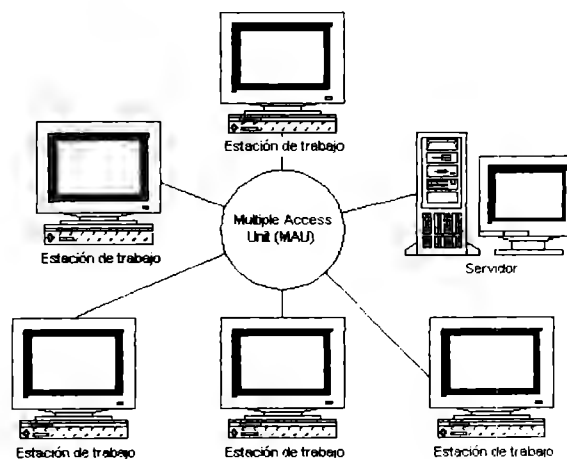


Figura 7. Topología "token ring" o de anillo

II.2.4. Protocolos Ethernet

Hablando de redes de datos, el término protocolo se refiere a un conjunto de reglas que rigen las comunicaciones, para que dos dispositivos de una red se comuniquen exitosamente. Es decir, los dos deben entender los mismos protocolos. A continuación se presenta una breve descripción de los protocolos más usados por esta tecnología.[5]

1.- TCP/IP: "Transmisión Control Protocol / Internet Protocol", es el protocolo principal que rige específicamente a Internet. TCP/IP permite la comunicación dentro de una red a través de los nodos de la misma. El protocolo IP es el encargado de realizar el movimiento de los datos entre computadoras o "host". TCP es responsable de que los comandos lleguen al otro extremo. Mantiene un control de la información transmitida y en caso de no llegar en buena forma a su destino, la retransmite. Si la información es muy grande para colocarse en un solo datagrama, TCP se encargará de colocarla en varios datagramas y se asegurará que lleguen correctamente a su destino. TCP e IP son dos protocolos que se usan frecuentemente en varias aplicaciones, por eso son colocados como uno solo. TCP/IP asigna un número único a cada equipo de trabajo en la red. Este número IP es un valor de cuatro bytes, que por conveniencia, es expresando en cuatro números decimales equivalentes que van de 0 a 255, cada byte es separado por un periodo o punto, por ejemplo 255.225.22.0; no deben existir dos terminales con la misma dirección IP en ningún momento.

2.- UDP "User Datagram Protocol", es otro protocolo de transporte donde Internet hace disponible su aplicación. No usa ninguna conexión y corre en la parte alta de redes IP. UDP/IP

proporciona pocos servicios de recuperación de errores y es por eso no es tan usado como TCP.

Pero algunas aplicaciones se adaptan mejor a este protocolo por las siguientes razones:

- Mayor velocidad y menores retrasos
- No mantiene una conexión de estado
- Velocidad de transmisión dependiente de la fuente
- Tamaño de encabezado menor

II.2.5. Ventajas de Ethernet (justificación de su elección)

Ethernet ha ido ganando popularidad con el paso de los años. Casi todas las computadoras de la actualidad presentan compatibilidades con Ethernet. Desde el hogar hasta la oficina, Ethernet es usado en numerosas aplicaciones incluyendo el diseño y estructura de redes (aproximadamente 85% de las redes LAN del mundo). Este proyecto utiliza las ventajas del amplio uso de Ethernet y de que es un protocolo que tiene las siguientes características: [6]

- Fácil de entender
- Bajo costo en implementaciones de red
- Proporciona una extensa flexibilidad topológica para la instalación de redes
- Garantiza interconexión y operación exitosa de equipos de diferentes proveedores

II.3. Lectores de Tarjetas

La lectura de datos provenientes de un dispositivo externo, en este caso tarjetas, se ha vuelto una de las formas más comunes de realizar algún proceso. Los tipos de tarjetas y sus respectivos lectores han inundado muchos campos y realizan aplicaciones de forma sencilla y rápida. Muchas veces es necesario un software especial que se instalará en la computadora para que la lectora funcione adecuadamente, y en la mayoría de los casos estos dispositivos pueden leer cualquier tarjeta de su tipo siempre y cuando esté programada correctamente.

Hay varios tipos de tarjetas que son usadas en la actualidad, los más comunes son las tarjetas magnéticas, las tarjetas denominadas “inteligentes” o con chip y las tarjetas con código de barras. La programación de cada una de ellas varía de acuerdo al tipo y a la información que quedará registrada en ellas. Los dispositivos que leen tarjetas están por todos lados; un ejemplo sencillo son las lectoras de código de barras utilizadas en los supermercados para registrar el artículo comprado y desplegar el precio del mismo en la pantalla de la computadora. Otro ejemplo cotidiano son los cajeros automáticos utilizados a diario por centenares de personas para consultar su estado de cuenta y realizar disposiciones de efectivo, el tipo de lector utilizado en este ejemplo es de tarjetas magnéticas. Y que decir de los lectores de tarjetas con chip como los empleados en las cabinas telefónicas distribuidas a lo largo y ancho de nuestro país. Estos ejemplos básicos constituyen sólo algunas de las formas más comunes de su uso para así registrar o realizar alguna actividad.

II.4. Tarjetas y Lectoras de Código de Barras

II.4.1. Códigos de Barras

La definición puntual de código de barras es: Dibujo formado por barras y espacios paralelos, que codifica información mediante las anchuras relativas de estos elementos. Los códigos de barras representan datos en una forma legible por las máquinas, y son uno de los medios más eficientes para la captación automática de datos.[10]

La primera patente para un código de barras, que tenía forma circular, fue solicitada en 1949 en Estados Unidos por N. J. Woodland; los códigos de barras se emplearon por primera vez a principios de la década de 1960 para identificar material rodante ferroviario. Y desde ese entonces ha seguido avanzando su aplicación y desarrollo.

El código de barras almacena datos que pueden ser reunidos de manera rápida y con una gran precisión y ofrecen con un método simple y fácil la codificación de información de texto que puede ser leída por lectores electrónicos de bajo costo.

Los Códigos de barras han sido creados para identificar objetos y facilitar el ingreso de información eliminando la posibilidad de error en la captura. Su estructura básica consiste de un patrón de inicio, uno o más caracteres de datos, opcionalmente unos o dos caracteres de verificación y patrón de término.

II.4.2. Impresión del código de barras

Los códigos de barras se pueden imprimir de varias maneras diferentes, entre ellas:

1.- Película maestra: Este método se utiliza para imprimir códigos de barras en imprentas, principalmente en empaques de comerciales destinados al comercio detallista. Se crea un original en una impresora de buena resolución y se reproduce por medios fotomecánicos añadiéndolo al original de impresión del empaque.

2.- Láser: Se puede utilizar una impresora láser imprimir planillas de etiquetas en bajo volumen o en documentos serializados que se imprimen eventualmente.

3.- Impresión térmica: Es la mejor tecnología para imprimir altos volúmenes de etiquetas en demanda o por lotes. Se utilizan impresoras industriales de mediana o alta velocidad que pueden imprimir sobre papel térmico o normal.

II.4.3. Características de los códigos de barras

Un símbolo de código de barras puede tener varias características, entre las cuales podemos mencionar:

1.- *Densidad*: Es la anchura del elemento (barra o espacio) más angosto dentro del símbolo de código de barras. Está dado en miles (milésimas de pulgada). Un código de barras no se mide por su longitud física sino por su densidad.

2.- *WNR*: "Wide to Narrow Ratio" Es la razón del grosor del elemento más angosto contra el más ancho. Usualmente es 1:3 o 1:2.

3.- *Quiet Zone*: Es el área blanca al principio y al final de un símbolo de código de barras. Esta área es necesaria para una lectura conveniente del símbolo.

II.4.4. Tipos de Códigos de Barras

Uno de los códigos de barras más empleados es el UPC "Universal Product Code" que utiliza 12 dígitos para representar el código. Emparentado con el UPC, existe el código ISBN, usado en la cubierta de libros y revistas, así como el código 39 que codifica números y letras para usos generales, siendo muy popular. Otro es el código entrelazado 2 de 5 (ITF), puede ser de cualquier longitud, pero con un número par de dígitos, siendo que codifica dos dígitos por vez. Este es uno de los pocos códigos en que los espacios en blanco tienen significado. También existen códigos de barras en 2 dimensiones, que se deben leer mediante un escáner o una cámara fotográfica digital.

Una de las más utilizadas es el símbolo internacional de número de artículo, llamado símbolo EAN – 13 por las siglas en inglés de la Asociación Europea para la Numeración de Artículos. Este tipo de código está formado por un margen, un dibujo normalizado de separación, un dibujo que representa directamente, seis dígitos e indirectamente un séptimo, un dibujo central de separación, un dibujo de barras y espacios que representa directamente seis dígitos, un dibujo normalizado de separación, un margen como se ilustra en la figura 8.

Cada dígito se representa mediante dos barras y dos espacios que tienen una anchura total de siete unidades; cada barra y cada espacio pueden tener una anchura de una, dos, tres o cuatro unidades. En la simbología EAN pueden elegirse tres formas distintas, A, B y C, para representar cada dígito. Estas formas se conocen como conjuntos numéricos.

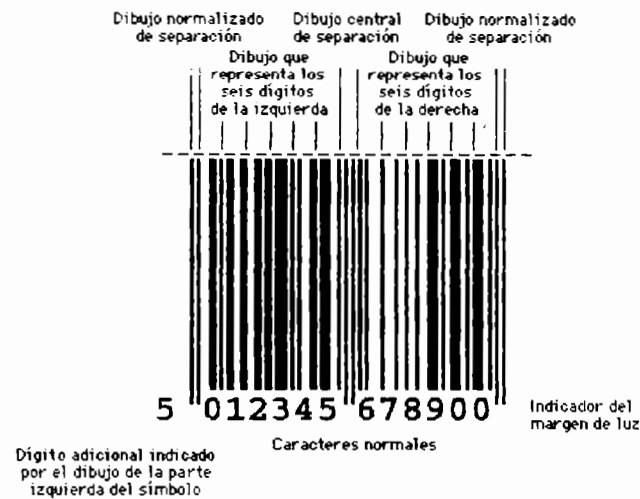


Figura 8. Estándar EAN-13 de código de barras

II.4.5. Lectoras de código de barras

El lector decodifica el código de barras a través de la digitalización proveniente de una fuente de luz que cruza el código y mide la intensidad de la luz reflejada por los espacios blancos. El patrón de la luz reflejada se detecta a través de una foto diodo el cual produce una señal eléctrica que coincide exactamente con el patrón impreso del código de barras. Luego esta señal es decodificada de regreso de acuerdo con la información original por circuitos electrónicos de bajo costo.[10]

Los lectores generan una señal digital pura de las barras y espacios. En el caso de los lápices ópticos ésta señal es de baja frecuencia, pues es generada por el barrido de las barras y espacios que hace el operador al deslizar el lápiz sobre el símbolo de código de barras. En el caso del láser, la señal es similar a la generada por el lápiz, sólo que a una frecuencia mucho mayor. Esta última señal es conocida como HHLC "Hand held laser compatible". En general las lectoras de tarjetas con código de barras son las de las más usadas en la actualidad debido a su versatilidad, fácil manejo y buen desempeño. La variada gama de aplicaciones permite que la mayoría de los artículos comercializados y muchas credenciales de uso institucional y corporativo tengan impreso un código de barras. Un lector de código de barras tiene tres etapas principales de funcionamiento:

- 1.- El dispositivo de entrada.- Lee los códigos de barras y transmite los datos al decodificador.
- 2.- El decodificador.- Convierte los datos en caracteres ASCII.
- 3.- La interfase.- Es la conexión entre el decodificador y la computadora.

II.4.6. Orientación a la identificación de personal

Es importante identificar de forma única a las personas, bien sea de una empresa, un club, una biblioteca etc., para controlar las actividades y operaciones que realizan; por ejemplo con control de activos fijos, control de acceso a áreas restringidas y manejo automático de nómina. Es de gran utilidad que dicha identificación sea estándar, a través del Código de Barras simbolizado e impreso en el carné o credencial de identificación. Así es

posible leer automáticamente la información de cada persona al realizar operaciones internas que requieran algún tipo de control.

Algunos otros beneficios otorgados por este tipo de aplicaciones son:

1. Seguridad en el registro de información
2. Agilización de procesos administrativos
3. Control de acceso en áreas restringidas
4. Agilización en la prestación de servicios

Muchas veces, el acceso a algún inmueble, área o edificio necesita ser monitoreado y controlado de forma sencilla pero rigurosa, es por eso que este tipo de aplicación brinda una seguridad muy completa.

II.4.7. Decodificador de teclado

Cuando se requiere que el decodificador sea de teclado, se utiliza lo que se conoce como keyboard wedge, el cual se conecta a la entrada del teclado de la PC o terminal. Existen diferentes tipos de wedges. Pueden tener una o dos entradas para lectores de código de barras y/o lector de cinta magnética, que son los más comunes. Estos decodificadores comúnmente se conectan a una PC, aunque hay modelos que pueden utilizarse en terminales tontas (WYSE, Link, IBM 5250). Obviamente se requerirá utilizar el cable apropiado y configurar el decodificador.

II.4.8. Láser de pistola

Realiza un barrido mediante una luz láser y que genera una señal similar a la del lápiz óptico, pero a una mayor frecuencia. Esta señal es conocida como HHLC (Hand Held Laser Compatible).

- Ventajas: es rápido, puede no requerir decodificador de teclado, puede leer a distancia (standard 5 a 30 cm, especial hasta 15m con etiquetas de papel retrorreflectivo), tiene un alto FRR.
- Desventajas: es relativamente caro (aunque existen modelos de 545 dls), puede presentar problemas de durabilidad debido a sus partes móviles (espejos giratorios), puede tener problemas para leer con demasiada luz ambiental.
- Precios: 500-1500 dls

II.5. Sistemas de Administración de Bases de Datos

Un sistema de administración de bases de datos o DBMS es una herramienta de software que permite que múltiples usuarios tengan acceso, almacenen y procesen los datos o hechos para convertirlos en información útil. Los DBMS son una de las razones principales por las que las personas usan computadoras. Muchas grandes organizaciones y compañías dependen en gran medida de los DBMS comerciales o personalizados para manejar grandes recursos de datos.

Del mismo modo los DBMS son herramientas vitales para personas y organizaciones que usan redes y computadoras personales independientes. Un DBMS hace posible realizar muchas tareas rutinarias que de otra manera serían tediosas, molestas y que consumen mucho tiempo de no ser por los servicios de una computadora.

Una base de datos contiene una colección de elementos o hechos relacionados, ordenados en una estructura específica. Por ejemplo, el directorio telefónico es una base de datos. En una base de datos computarizada, por lo general se introducen los datos y se ven en una tabla bidimensional que consiste de filas y columnas, similar a la estructura de una hoja de cálculo.

La colección entera de datos relacionados en la tabla se denomina archivo. Cada fila representa un registro, el cual es un conjunto de datos para cada entrada en la base de datos. Cada columna de la tabla representa un campo, el cual agrupa cada pieza o elemento de los datos entre los registros en categorías o tipos específicos de datos.

El orden de los campos en una tabla define en forma estricta la ubicación de los datos en cada registro. Un campo de número telefónico, por ejemplo, debe contener el número telefónico de un registro; es decir, en este caso no puede contener el nombre de una persona o un código postal. Del mismo modo, el conjunto de campos en cualquier tabla proporciona una definición sensible de la base de datos para aquellos que deben tener acceso a sus datos.

Un archivo de bases de datos que consta de una sola tabla de datos es una base de datos de archivo simple. Las bases de datos de archivo simple son útiles para ciertas situaciones de

usuario único o de grupos pequeños, en especial para mantener listas como directorios de direcciones o inventarios. Los datos que se almacenan, manejan y manipulan en una hoja de cálculo electrónica son otro ejemplo de una base de datos de archivo plano.

En una base de datos relacional, que es una base de datos formada por un conjunto de tablas, un campo común existente en cualquiera de las dos tablas crea una relación entre éstas. La estructura de base de datos relacional es la más frecuente en las organizaciones comerciales actuales.

Otros tipos de bases de datos son:

Bases de datos jerárquicas: En este tipo de base de datos los registros se organizan en una estructura tipo árbol. Se dice que la relación entre tipos de registros es una relación de padre a hijo, en la que cualquier tipo de hijo se relaciona sólo con un tipo único de padre.

Bases de datos en red: La base de datos en red es similar a la estructura jerárquica excepto que cualquier tipo de registro puede relacionarse con cualquier número de otros tipos de registros. Como la estructura jerárquica, la de red se usa en sistemas antiguos como mainframes.

Bases de datos orientadas a objetos: Una base de datos orientada a objetos es una estructura más nueva que recientemente ha generado un gran interés. Esta estructura agrupa elementos de datos y sus características, atributos y procedimientos asociados, en elementos complejos llamados objetos. Desde el punto de vista físico, un objeto puede ser cualquier cosa: un

producto, un evento, una sucursal, un empleado, etc. Un objeto se define por sus características, atributos y procedimientos. Las características de un objeto pueden ser texto, sonido, gráficos y video. Ejemplos de atributos podrían ser el color, el tamaño, el estilo, la cantidad, la nómina y el área de ingreso. Un procedimiento se refiere al procesamiento o manejo que puede estar asociado con un objeto.

II.6. Edificios Inteligentes

Un edificio inteligente es aquel que proporciona un ambiente de trabajo productivo y eficiente a través de la optimización de sus cuatro elementos básicos: estructura, sistemas, servicios y administración, con las interrelaciones entre ellos. Los edificios inteligentes ayudan a los propietarios, operadores y ocupantes, a realizar sus propósitos en términos de costo, confort, comodidad, seguridad, flexibilidad y comercialización.[12]



Figura 9. WTC Ciudad de México

II.6.1. Grados de inteligencia

Existen tres grados de inteligencia, catalogados en función de la automatización de las instalaciones o desde el punto de vista tecnológico:

Grado 1. Inteligencia mínima o básica. Un sistema básico de automatización del edificio, el cual no está integrado.

- Existe una automatización de la actividad y los servicios de telecomunicaciones, aunque no están integrados.

Grado 2. Inteligencia media. Tiene un sistema de automatización del edificio totalmente integrado.

- Sistemas de automatización de la actividad, sin una completa integración de las telecomunicaciones.

c) Grado 3. Inteligencia máxima o total. Los sistemas de automatización del edificio, la actividad y las telecomunicaciones, se encuentran totalmente integrados. El sistema de automatización del edificio se divide en: sistema básico de control, sistema de seguridad y sistema de ahorro de energía.

- El sistema básico de control es el que permite monitorear el estado de las instalaciones, como son: eléctricas, hidrosanitarias, elevadores y escaleras eléctricas, y suministros de gas y electricidad.

- El sistema de seguridad protege a las personas, los bienes materiales y la información. En la seguridad de las personas, destacan los sistemas de detección de humo y fuego, fugas de gas, suministro de agua, monitoreo de equipo para la extinción de fuego, red de rociadores, extracción automática de humo, señalización de salidas de emergencia y el voceo de

emergencia. Para la seguridad de bienes materiales o de información, tenemos el circuito cerrado de televisión, la vigilancia perimetral, el control de accesos, el control de rondas de vigilancia, la intercomunicación de emergencia, la seguridad informática, el detector de movimientos sísmicos y el de presencia.

- El sistema de ahorro de energía es el encargado de la zonificación de la climatización, el intercambio de calor entre zonas, incluyendo el exterior, el uso activo y pasivo de la energía solar, la identificación del consumo, el control automático y centralizado de la iluminación, el control de horarios para el funcionamiento de equipos, el control de ascensores y el programa emergente en puntos críticos de demanda

III. DESARROLLO DEL PROYECTO

III.1. Diseño

El diagrama a bloques esta conformado como se muestra a continuación, en la figura 9:

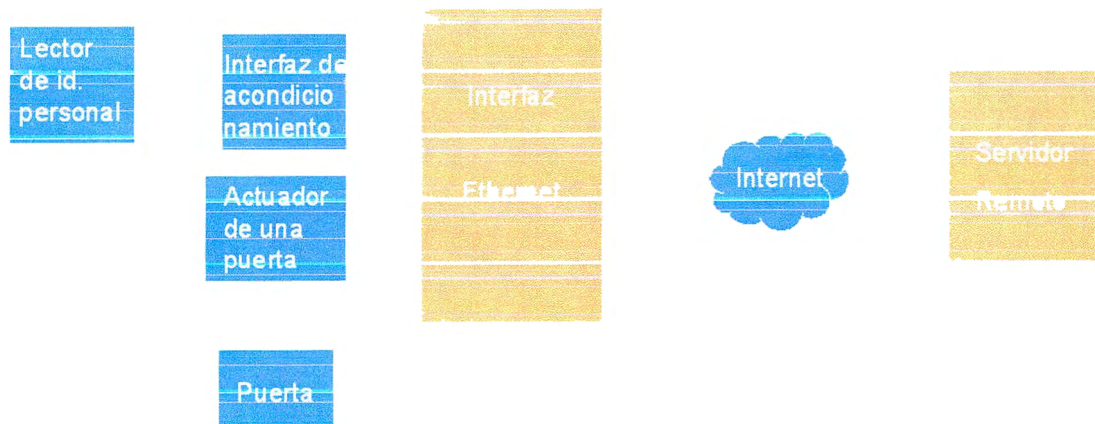


Figura 10. Diagrama a bloques del sistema a implementar

Consiste en un dispositivo electrónico que recibe un paquete de datos de una lectora de código de barras para la identificación de una persona que ingresa a cierta área de un edificio, la innovación del proyecto es que tal dispositivo contará con una interfaz Ethernet para evitar la necesidad de tener una computadora en cada área del edificio a la que se desee ingresar, pudiendo así, controlar el acceso a todas las áreas de un edificio con una sola computadora.

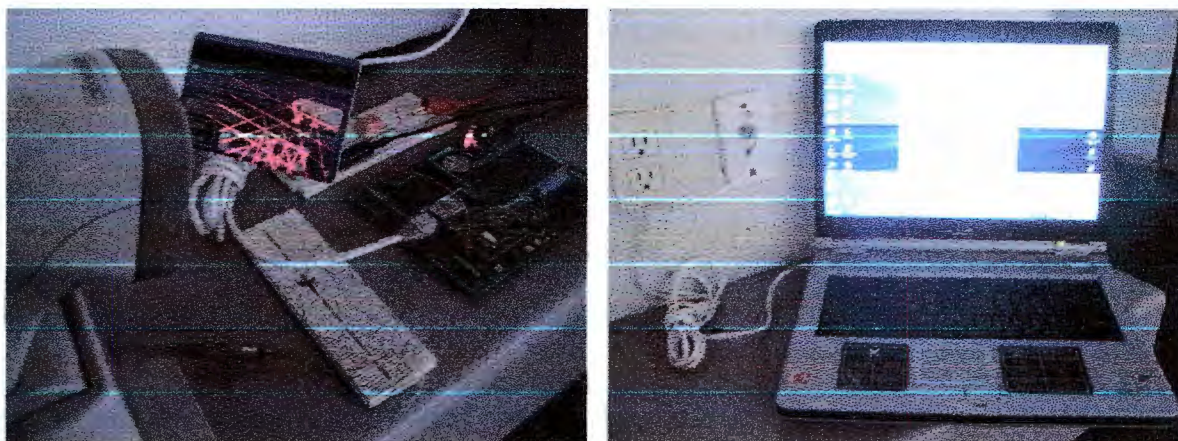


Figura 11. Sistema de Control de Acceso con comunicación Ethernet, a. Validación en el punto de acceso, b. Servidor de BDD

El dispositivo recibe el paquete de datos de la lectora de tarjetas de identificación personal, la manda por Ethernet a una computadora que cuenta con una base de datos de las personas que tienen acceso al área a la que se intenta ingresar (esta base de datos se creará en Microsoft Access®), compara los datos gracias a un programa realizado en Microsoft Visual Basic® y, en caso de que la persona este autorizada, manda una señal a través de Ethernet y dicha señal permite abrir una puerta, a través de un actuador, que puede ser tan sencillo como un relay que abra una chapa, también se generará automáticamente una base de datos que registre la hora de entrada y el nombre de la persona que entró a esa área, todos estos datos podrán

consultarse desde cualquier terminal remota que esté conectada a Internet, si es que se cuenta con la autorización para ello.

III.2. Construcción

III.2.1. Adaptación de la señal proveniente de la lectora de código de barras al módulo Ethernet

La salida de la lectora de códigos de barras cuenta con un conector mini DIN de 6 pines y envía datos en la misma forma que un teclado que cuente con este conector y que siga el protocolo PS2 desarrollado por IBM.

La configuración de pines es como sigue:

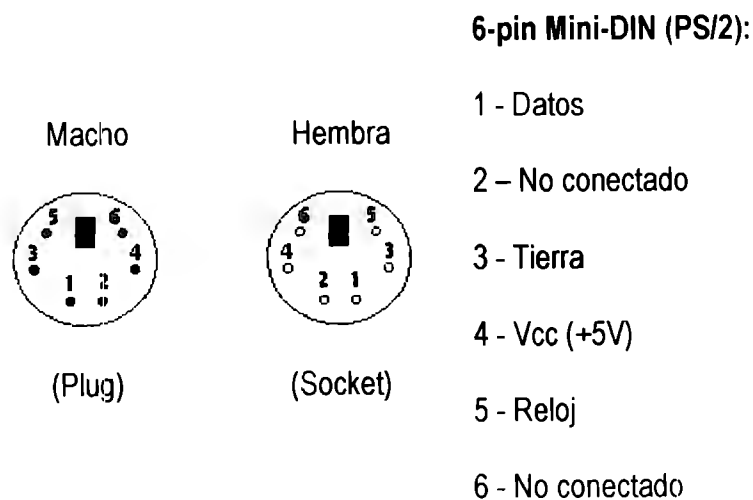


Fig.12. Configuración de pines de un conector mini DIN 6

El pin de VCC alimenta a la lectora de código de barras, las especificaciones de potencia sugieren valores de $VCC = +4.5\text{ V}$ a $+5.5\text{ V}$ y una corriente máxima de 275 mA.

Las líneas de datos y de reloj pueden tener dos estados: bajo o alta impedancia. Esto representa un problema si se desea desarrollar una aplicación y se requieren manejar valores de 0 y 5 volts. Sin embargo esto puede resolverse fácilmente colocando resistencias de pullup entre las líneas y VCC, en el estado bajo se vera tierra desde el microcontrolador o microprocesador y cuando se encuentre en alta impedancia el circuito se abre y se ven los 5 volts de VCC desde el microprocesador. El valor utilizado puede en las resistencias pueden variar entre 1 y 10 $K\Omega$, una resistencia grande produce menos consumo de potencia y una resistencia pequeña produce tiempos de subida rápidos, para el proyecto seleccionamos un valor intermedio, siendo este de 4.7 $K\Omega$.

La lectora de códigos de barras PS/2 se comunica con un host, que puede ser una computadora o en nuestro caso una tarjeta de desarrollo, mediante un protocolo síncrono serial bidireccional. El bus se encuentra esperando información cuando las líneas de reloj y datos se encuentran en alto. La lectora de código de barras siempre genera la señal de reloj, y los datos se transmiten un byte a la vez y cada byte consiste de una trama de 11 bits:

- Un bit de inicio (siempre es cero)
- 8 bits de datos (el menos significativo primero)
- 1 bit de paridad
- 1 bit de fin de trama (siempre es uno)

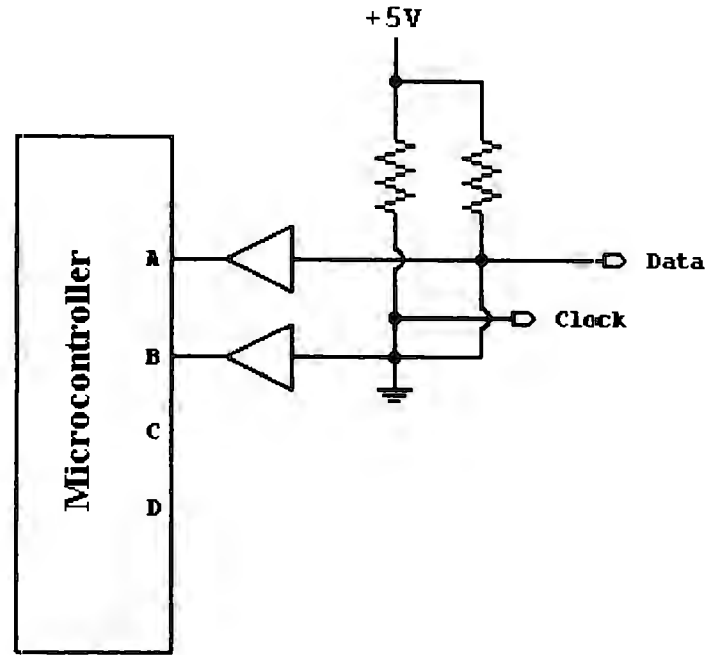


Fig.13. Configuración de colector abierto de las líneas "Data" y "Clock"

En el circuito que finalmente se implementó en este proyecto se utilizaron las líneas 1 y 3 del puerto G (PG1 y PG2) para la señal del reloj, tomada directamente de una extensión de los pines Mini-Din. De la misma forma se utilizó la línea 2 del puerto G (PG2) para recibir los datos provenientes del pin respectivo del lector de código de barras. Para ver el funcionamiento del programa que recibe los bits y los convierte en caracteres ASCII refiérase al anexo VI.1 en la página 65. Los 5 volts de alimentación se obtienen directamente de la tarjeta de diseño tipo protoboard disponible en la interfaz Ethernet, que cuenta con líneas de alimentación de 3.3 volts, 5 volts y tierra.

El host detecta el flanco de bajada de la línea de reloj para recibir datos provenientes de la lectora. La frecuencia del reloj es de 11.5 KHz, esto significa que el reloj está en alto 43 microsegundos y en bajo otros 43 microsegundos.

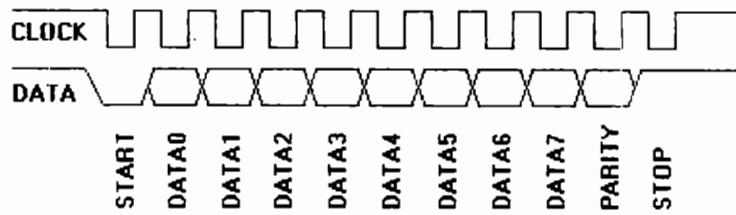


Fig.14 Comunicación de la lectora a un host

Cuando una tecla se presiona en un teclado PS2, o cuando la lectora de código de barras PS2 envía al host una letra o número a esta información se le denomina "scan code", cada letra o número tiene un "scan code" diferente para cada una de sus acciones, esto es tiene un scan code para cuando se oprime una tecla (make code) y un scan code para cuando esa tecla se suelta (break code), ya que la lectora envía una trama que emula a la de un teclado los códigos que envía son exactamente los mismos.

Existen tres juegos de códigos para teclados, mediante pruebas en el laboratorio y comparación con nuestras fuentes de información logramos identificar que en nuestro caso se utiliza el juego número dos. A continuación se muestra el ejemplo de los valores en hexadecimal que son enviados a un host al oprimir ciertas teclas, o al enviar una letra o un número:

Tecla	(Juego 2) Make Code	(Juego 2) Break Code
"A"	1C	F0,1C
"5"	2E	F0,2E

Tabla 1. Make y Break Codes para "A" y "5"

Los make codes están formados por un byte y los break codes están formados por dos bytes, el host es el encargado de relacionar el valor en hexadecimal que esta recibiendo con su letra correspondiente basándose, en este caso en el juego dos de scan codes. [11]

III.2.2. Servidor de Base de Datos

Elaboramos un programa en el entorno Visual Basic 6.0® el cual es la base de operación de nuestro sistema, este programa había sido elaborado de una forma más simple en el semestre anterior [7], y ahora cuenta con prácticamente todas las funciones requeridas en el proyecto, restando únicamente su adaptación al puerto de red del servidor de base de datos. Este programa debe estar corriendo en el servidor de base de datos del sistema, el servidor contiene una base de datos del tipo relacional elaborada en Microsoft Access® que tiene los registros de los usuarios que tienen permiso de ingresar a un área determinada, el programa elaborado permite visualizar los registros uno a uno, añadir un nuevo registro, borrar un registro, así como también buscar un campo deseado.

Los campos superiores "Nombre", "Matrícula" y "Área", muestran respectivamente el nombre matrícula y número de área de un usuario determinado, los botones "Agregar" y "Eliminar" permiten añadir o borrar algún registro de la base de datos, la ventana también cuenta con un "control data" el cual tiene ésta apariencia: « ‹ › » , estos botones permiten "navegar" en la base de datos y visualizarla registro por registro, el botón "Buscar matrícula" busca en el campo

“matrícula” registro a registro el “string” de datos que sea ingresado en el cuadro de texto que se encuentra a su izquierda, de ser satisfactoria la búsqueda, se desplegara en el cuadro de texto junto a la etiqueta: “Registros Encontrados” un uno, como se muestra en la figura anterior, de lo contrario se desplegara un cero, este valor se almacena en una variable y el que se autorice o no el acceso a un usuario dependerá a fin de cuentas de el valor de esta variable. El botón “Limpiar” permite poner en blanco todos los cuadros de texto.

Si se conecta directamente el lector de código de barras a la computadora este programa funciona correctamente, al hacer pasar una tarjeta con un código de barras los caracteres enviados se escriben en el cuadro de texto junto a la etiqueta “Buscar matrícula”, como el último carácter enviado por la lectora es un “Enter” se presiona automáticamente el botón “Buscar matrícula”, y se efectúa el proceso de búsqueda descrito anteriormente.

III.3. Pruebas

III.3.1. Servidor de base de datos

Elaboramos un programa en el entorno Visual Basic 6.0® que es la base de operación de nuestro sistema, dicho programa debe estar corriendo en el servidor del sistema, el servidor contiene una base de datos elaborada en Microsoft Access® que tiene los registros de los usuarios que tienen permiso de ingresar a un área, el programa elaborado permite visualizar los registros uno a uno, añadir un nuevo registro, borrar un registro, así como también buscar un campo deseado.[8]

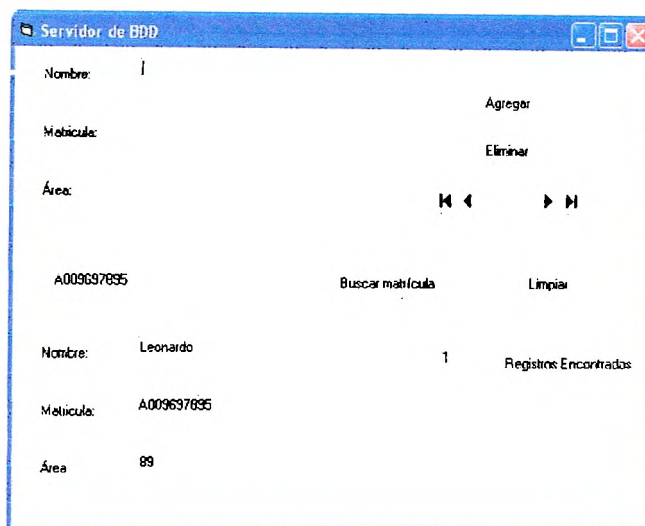


Figura 15. Ventana del programa BDD

Los campos que manejamos en nuestra base de datos son matrícula, nombre y área de un usuario.

Algunas de las complicaciones que se tuvieron en la realización de estas pruebas fueron: La incompatibilidad con ciertas versiones de Microsoft Office® (Office 2000 versión estudiantil) al invocar la base de datos de Access, esto se resolvió simplemente adquiriendo la versión posterior de dicha paquetería. También se tuvieron problemas al incorporar la parte del servidor de red ya que el programa no funcionaba adecuadamente al ser ejecutado en una red simulada con un cable cruzado, sin embargo al ejecutarse en la red del Tecnológico de Monterrey CCM realizaba todas las funciones deseadas.

Este programa se corre junto con un servidor integrado en el código de Visual Basic, y se corre un programa de cliente del lado de la interfaz Ethernet, estableciendo así una sesión de transmisión de datos satisfactoria.

Para ver el código fuente de este programa refiérase al anexo en la página 74.

III.3.2. Interfaz Ethernet

Elaboramos un programa en Dynamic C que carga una página web en la memoria flash de la tarjeta RCM3200 de Rabbit Semiconductors®. Esta página web cuenta con un controlador que permite prender y apagar los LED's de la tarjeta desde cualquier terminal conectada a Internet, mediante un navegador, como Internet Explorer o Netscape.

La tarjeta debe configurarse de acuerdo a las características de la red en que se conecte, en el caso de su manejo en el ITESM Campus Ciudad de México, debe configurarse para que adquiera una dirección IP mediante DHCP (asignación dinámica de IP).

Una vez que se adquiere y se conoce la IP de la tarjeta RCM 3200, basta con ingresar dicha IP y el nombre del archivo de la página web en la barra de direcciones de un navegador en una terminal conectada a Internet. Esto desplegará la página web que contiene el controlador de los LED's de la tarjeta de diseño.

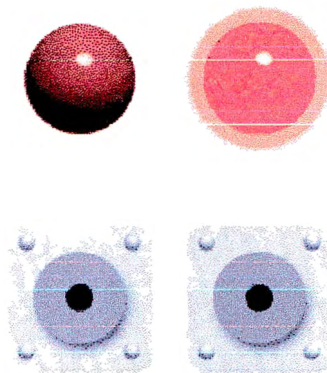


Figura 16. Control de LED's desde una página Web

Este programa es de utilidad ya que comunica la interfaz Ethernet con una terminal remota, lo cual es el propósito primordial de nuestro proyecto en esta primera etapa, una vez que el proceso de identificación sea automatizado las funciones de este programa permitirán enviar una señal que prenda un foco rojo cuando un determinado usuario no tenga acceso a cierta área del edificio donde se tenga el sistema, o en su defecto que mande una señal que active un relay, en vez de prender un foco o un LED, y que pueda abrirse la cerradura de la puerta.

Para ver el código fuente del programa refiérase al anexo 3 de la página 71.

Las complicaciones que se presentaron trabajando con la interfaz fueron numerosas, por mencionar algunas tenemos: La poca inteligibilidad de algunas librerías disponibles para la programación del módulo, la inexperiencia con el software Dinamic C® que pese a ser un entorno amigable, tiene funciones y declaraciones muy diferentes a las ya conocidas por nuestros cursos de programación. Estas adversidades se vieron parcialmente resueltas al consultar el manual de usuario de la interfaz, revisar algunos programas ejemplo y al programar por nuestra cuenta diversas funciones.

Otro problema que encontramos fue el entendimiento de la funcionalidad de la red del campus, para que la interfaz se acoplara correctamente a ésta y le fuera asignada su dirección IP correspondiente, para resolver esto tuvimos que analizar la librería específica sobre IP y programar un direccionamiento dinámico. Durante las pruebas de medio término encontramos complicaciones en la librería HTTP del módulo ya que no permitía que ciertas imágenes se desplegaran en una página Web que era creada para manipular los LEDs. Solucionamos este problema cambiando la extensión de los archivos de imagen a JPG.

III.2.3. Adquisición de Datos

La lectora de código de barras funciona como un teclado común, es decir, los datos son enviados en forma serial y en código ASCII por medio de un cable mini DIN 6 pines. Se utiliza un reloj que indica el inicio y fin de transmisión (sincronía).

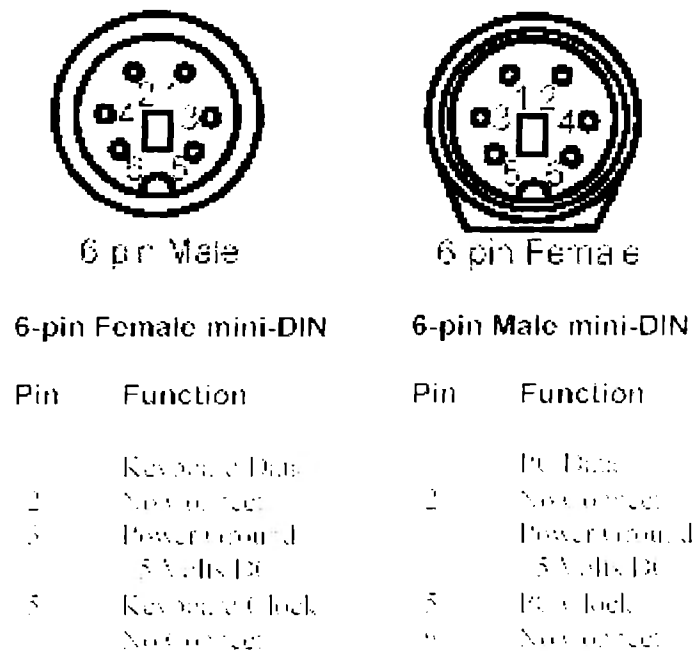


Figura 17. Distribución de pines de un conector mini DIN

La lectora de código de barras utilizado para este proyecto es el modelo *MS6720-47 KW* "Keyboard Wedge" de la marca Metrologic®, mostrado en la figura 13. Este dispositivo se encuentra en el almacén de los laboratorios de electrónica del ITESM-CCM lo cual ha sido favorable para el equipo pues no requerimos realizar una inversión más para adquirir este dispositivo; y su buen estado y funcionamiento son indicadores de que será de gran ayuda en la realización del proyecto. *MS6720-47 KW* es el primer lector de código de barras láser que une la

distancia entre los lectores de proyección fijos omnidireccionales y los lectores manuales, es decir, con este lector puede ser colocado fijamente o puede ser utilizado manualmente para realizar la lectura.



Figura 18. Lectora de código de barras MS6720-47 KW

El lector de código de barras operara de forma similar a un teclado, como se mencionó anteriormente, enviando de forma serial síncrona los scan codes de cada uno de los números o caracteres, presentando una pausa de cincuenta microsegundos entre dato y dato. Siempre que se lee un código de barras la última forma de onda que aparece en el osciloscopio es la de la tecla "enter " lo cual define el funcionamiento de la lectora y nos facilita ciertas cuestiones de trabajo.

En la siguiente figura podemos observar la forma de onda arrojada por el lector de código de barras para la tecla enter, observamos en la parte superior la señal de reloj a cierta frecuencia

Escuela Superior de Ingeniería y Tecnología
Ciudad de México

(11.51 kHz) mientras existe la transmisión del dato. El dato se caracteriza por una trama de once bits de una duración de 43 microsegundos con valores de 0 y 5 volts.

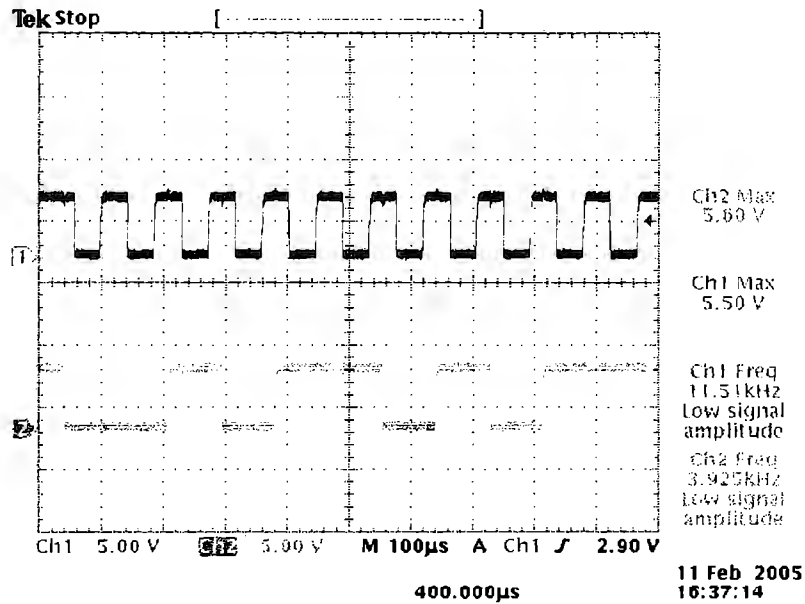


Fig.19. Señal de la tecla "enter"

Así para la letra "A" y el número 5 se tienen sus respectivas señales respectivamente:

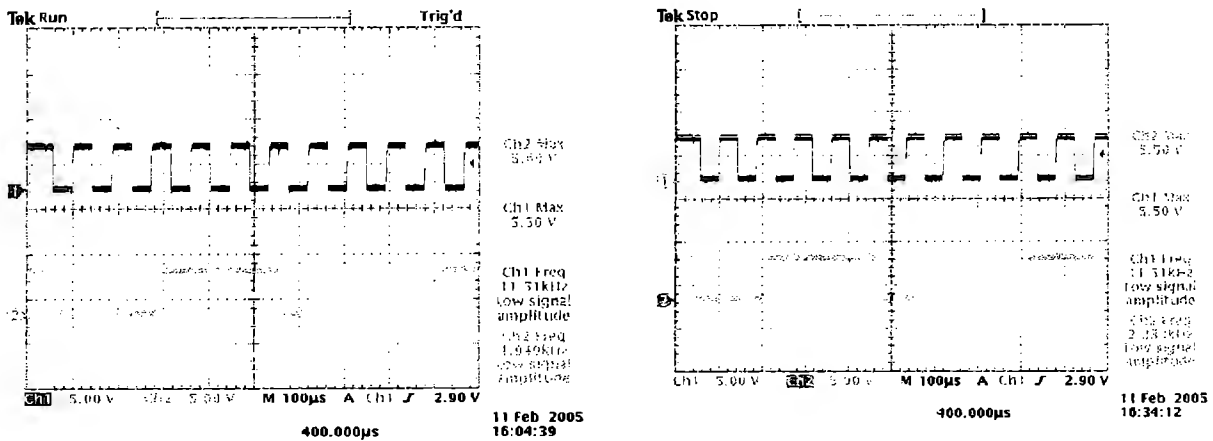


Fig.20. Señales de las teclas "A" y "5"

Las complicaciones en esta parte del trabajo se presentaron cuando quisimos identificar la forma de onda arrojada por el lector cuando realiza la validación de un código. La lectora tiene que estar alimentada mientras funciona y la tierra fue difícil de identificar puesto que la carcasa del conector, además del pin correspondiente, también está referenciada a tierra. Para analizar los pines de datos y de reloj fue necesario utilizar extensiones que permitieran colocar las resistencias de pull up y así ver la forma de las ondas en el osciloscopio.

Por último la decodificación de los datos provenientes del lector de código de barras nos dio algunos problemas en la sincronización de los relojes de la lectora y del microprocesador, arrojando basura en los datos obtenidos de la matrícula o nómina del usuario.

Para ver el código fuente de este programa refiérase al anexo en la página 65.

III.4. Resultados

III.4.1. Programa "Servidor" del sistema en una terminal remota

Este programa fue elaborado en el lenguaje de Visual Basic®, es el "cerebro" del sistema, se está ejecutando constantemente y su función consiste en recibir una matrícula o nómina del usuario que se está identificando, verificar su existencia en la base de datos y responder enviando a la interfaz Ethernet una señal que autorice o rechace el ingreso del usuario según sea el caso. El programa también genera una base de datos de tipo relacional con la información (nómina, nombre, fecha, hora y área) de los usuarios que transitan por cada una de las áreas. Para ver el código fuente de este programa refiérase al anexo 4 en la página 74.

III.4.2. Programa "Cliente" en la Interfaz RCM3200

Este programa fue elaborado en el lenguaje Dynamic C® y su función consiste en recibir la señal de autorización o rechazo proveniente del servidor de BDD y controlar el relevador del circuito de la puerta para que permanezca cerrada o se abra según corresponda.

Para ver el circuito realizado vea la figura 20, el código fuente de dicho programa se presenta en el anexo 2 en la página 67.

III.4.3. Programa de adaptación de la señal de la lectora de código de barras

Este programa fue elaborado en el lenguaje Dynamic C® y su función consiste en decodificar una serie de bits recibidos en forma síncrona y asignarles su carácter ASCII correspondiente, almacenar en memoria el dato recibido y enviarlo al servidor de BDD para que sea buscado en la base de datos por el programa "servidor" mencionado previamente (Terminal remota). Para ver el código fuente de dicho programa refiérase al anexo 1 en la página 65.

III.4.4. Elaboración de una puerta y una base de datos

Con el propósito de montar una simulación del sistema elaboramos una pequeña puerta de madera que tiene incorporado un electroimán cuyo circuito se abre y cierra con la acción de un relevador. El diseño final del circuito que controla la apertura y cierre de la puerta se muestra a continuación:

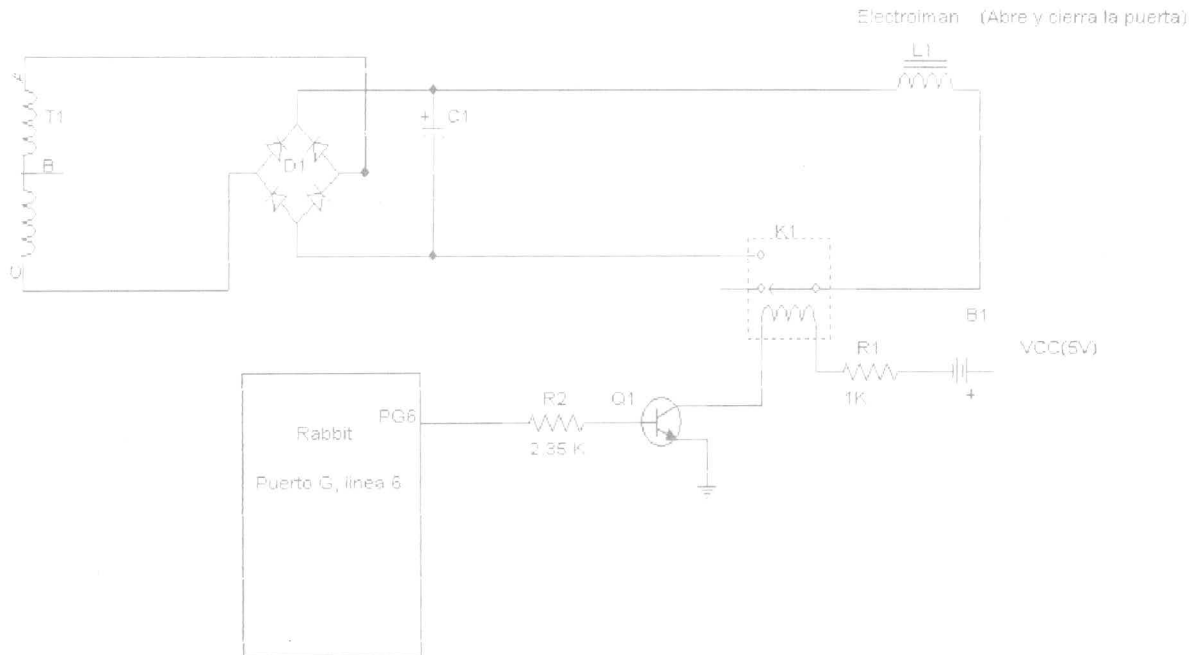


Fig. 21 Circuito para el control de la puerta de simulación.

El circuito consta de un transformador 30V – 1A que cumple las demandas de corriente y voltaje necesarias para el funcionamiento del electroimán acoplado en la puerta. Los extremos del devanado secundario de dicho transformador se conectan a un puente de diodos rectificador capaz de soportar la corriente demandada (1A), el capacitor proporciona el nivel adecuado de corriente directa DC al electroimán. Por otro lado el relevador de 5V se encuentra normalmente cerrado manteniendo la puerta cerrada. Cuando la línea 6 del puerto G (PG6) cambia de estado indicando el acceso concedido, el transistor TIP41 actúa como switch abriendo el circuito del relevador, lo que provoca que la corriente circulando por el electroimán se nulifique y así la puerta pueda ser empujada y abierta con facilidad. Al cabo de 6 segundos la línea PG6 se cambiará de estado activando nuevamente el circuito del relevador manteniendo cerrada la puerta.

La base de datos realizada en Microsoft Access® contiene los registros siguientes: Id. Personal, Nombre, Área y Cargo de cada uno de los usuarios, y en ella se realizará la búsqueda cuando algún usuario se identifique, y de encontrarse se permitirá su ingreso. Así mismo se construye una base de datos en Microsoft Access® 2000 de tipo relacional. La base una tabla compuesta por los campos: Matrícula, Apellido Paterno, Apellido Materno y Observaciones; donde estarán los datos de los usuarios cuyo ingreso está autorizado. La tabla "Registro" con los campos: ID, Matrícula, Fecha, Hora y Área; será donde se registre la hora, fecha y área de acceso del usuario que ingrese al edificio.

El acceso a base de datos se lleva a cabo mediante la interfaz de programación de aplicaciones DAO (Data Access Objects). DAO utiliza el motor de bases de datos Microsoft Jet para proporcionar un conjunto de objetos de acceso a datos: objetos de base de datos, objetos de definición de tabla, objetos de definición de consulta y objetos de conjunto de registros, entre otros. DAO proporciona el mejor rendimiento con archivos .Mdb como los que crea Microsoft Access, pero también puede tener acceso a orígenes de datos ODBC (Open Database Connectivity) a través de DAO y del motor de bases de datos Microsoft Jet.

III.4.5. Análisis Financiero

Se realizó el siguiente análisis financiero para demostrar en una forma más tangible las ventajas que traería a una empresa implementar el sistema:

Considérese una institución mediana (5 áreas):

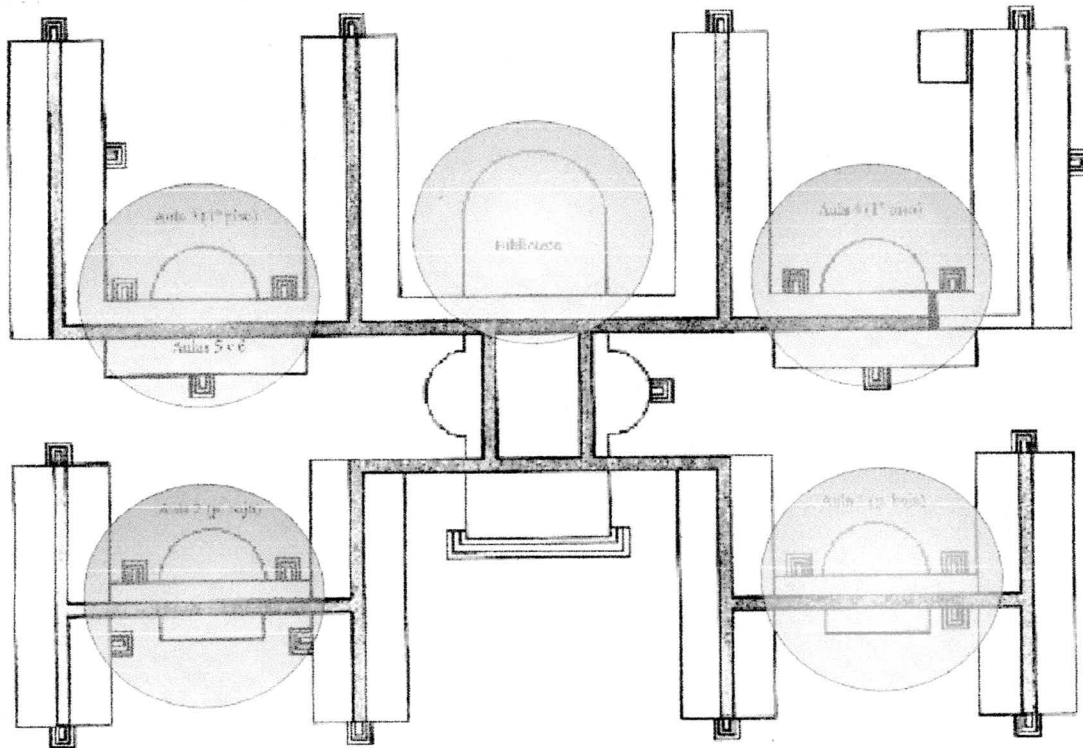


Fig. 22 Institución Ejemplificada

- Aulas 1 – 4
- Biblioteca
- Inversión Inicial

Se consideró que la empresa cuenta con la infraestructura de red adecuada para la implementación del proyecto, por lo que los conceptos que deben ser considerados en la inversión inicial son los que se muestran en la siguiente tabla:

Concepto	Costo unitario	Cantidad	Total
Tarjeta RCM3200	\$ 1,023.50	5	\$ 5,117.50
Lectora código de barras	\$ 2,127.50	5	\$ 10,637.50
Cableado	\$ 7 por metro	300	\$ 2,100.00
Puerta	\$ 4,025.00	5	\$ 20,125.00
Tarjeta para usuario	\$ 10.00	1500	\$ 15,000.00
		TOTAL	\$ 52,980.00

Tabla 2. Inversión Inicial

- Recuperación de la Inversión

En esta parte se consideró el salario promedio de personal de vigilancia en el Distrito Federal, recabando la información de distintas empresas que proveen estos servicios en Internet y en la sección amarilla.

Concepto	Sueldo Mensual	Cantidad	Total
Vigilante	\$ 3,500.00	5	\$ 17,500.00

Tabla 3. Nómina de vigilantes

Tiempo de recuperación.- Para que el proyecto sea viable, los inversionistas esperan que el tiempo de recuperación sea razonablemente corto. Relacionando únicamente las variables del costo de implementación del proyecto y el ahorro generado por la relevación hombre – máquina nos arroja un tiempo de recuperación como sigue:

$$\$52,980 / \$17,500 = 4 \text{ meses}$$

Por lo que consideramos que la implementación del sistema en este caso es factible.

A continuación se muestra una gráfica de los beneficios que traería en términos monetarios el uso del dispositivo en un periodo de dos años.

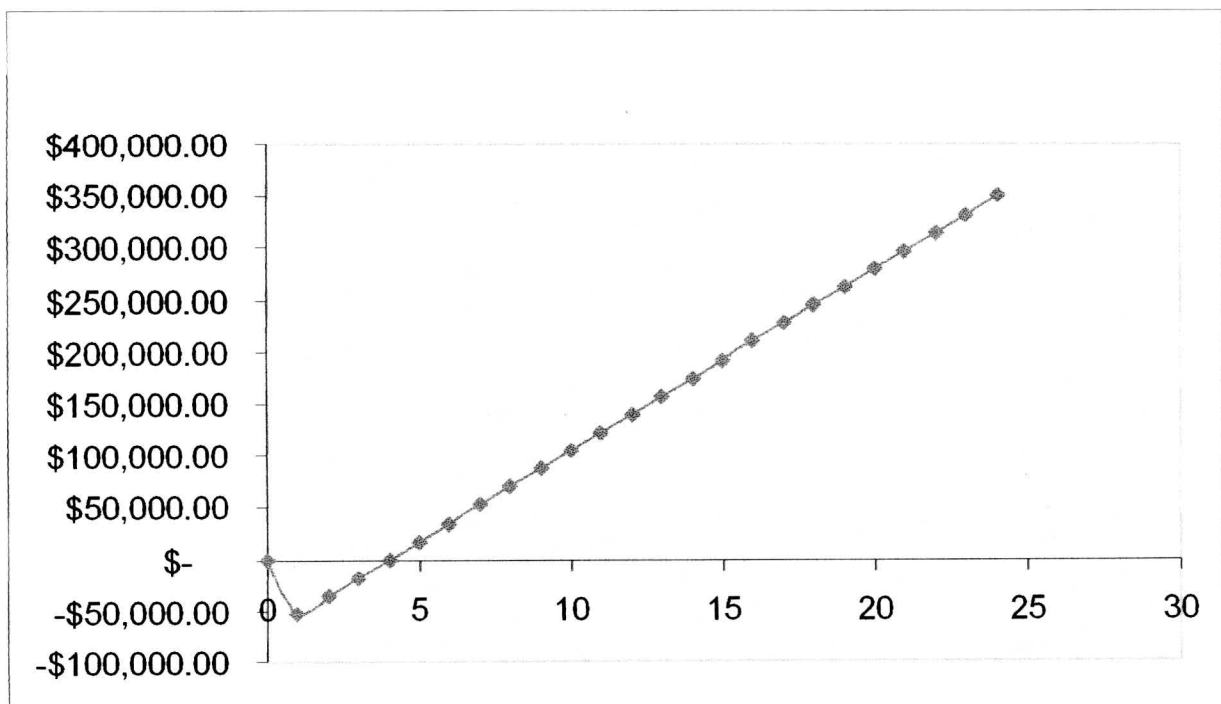


Figura 23. Ahorro obtenido en dos años.

En el eje de las abscisas se muestran los meses y en el de las ordenadas el ahorro en pesos mexicanos de dos mil cinco. Como podemos apreciar por el segundo año (mes veintecuatro) se tiene un ahorro cercano a los trescientos cincuenta mil pesos, lo que demuestra nuevamente lo redituable del proyecto.

IV. CONCLUSIONES

IV.1. Perspectivas y trabajo a futuro

La primer parte del proyecto involucró el manejo de las siete capas del modelo de referencia OSI, y logramos realizar correctamente cada una de sus etapas. Para el periodo de Proyectos de Ingeniería II nos enfocamos al desarrollo del software necesario para manipular la capa física del sistema, este desarrollo incluye una base de datos más compleja que la utilizada para la demostración del final del primer periodo, también realizamos otra base de datos donde se almacenan los datos de los usuarios que ingresan a las diversas áreas de un edificio, y que puede ser consultada por algún supervisor que tenga la debida autorización para hacerlo.

También elaboramos un programa en Dynamic C que recibe los datos que envía la lectora de códigos de barras, almacene estos datos en su memoria flash y los envíe a través de Ethernet hacia el servidor de Base de Datos, finalmente adaptamos el programa que fue elaborado en el primer periodo y que permite controlar los LED's de la tarjeta de diseño desde una terminal remota, para que pueda acoplarse con el programa elaborado en Visual Basic y envíe este par de señales hacia un foco rojo cuando un usuario no se encuentre en la base de datos y que la otra señal active un relevador para abrir una cerradura en caso de que el usuario se encuentre en la base de datos.

No cabe duda que existe mucho trabajo por hacer para perfeccionar el proyecto y así alcanzar los objetivos ideales de seguridad, flexibilidad y operación que se requieren en su implementación. Las sugerencias de los autores del proyecto son las siguientes:

- Aumentar el nivel de seguridad del sistema mediante el uso de equipos de identificación más sofisticados, tales como: equipos biométricos (lectores de retina y de huella dactilar), tarjetas inteligentes (con chip), etc.

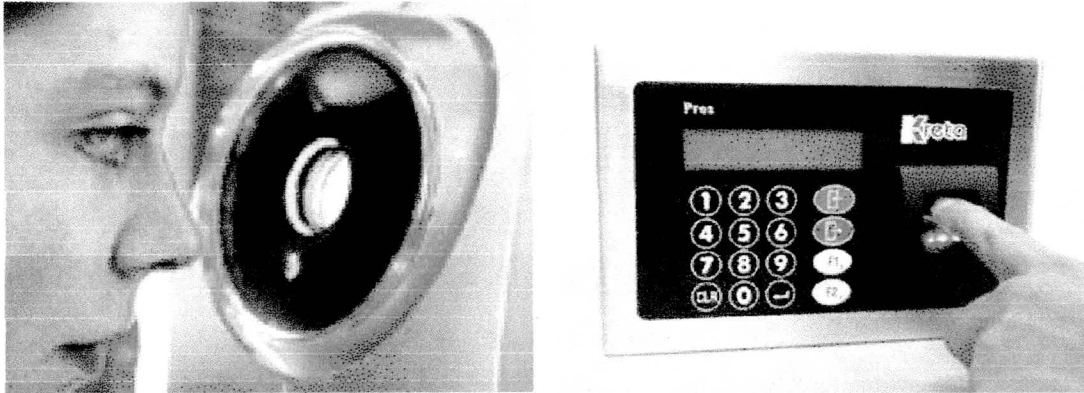


Figura 24. Equipos biométricos a. Lector de retina,
b. Lector de huellas dactilares.

- Construcción de una puerta que garantice el acceso de una sola persona a la vez (Ej. una puerta giratoria, torniquete, etc.)

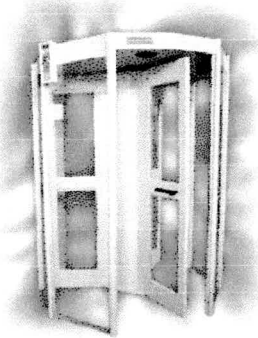


Figura 25. Puerta giratoria

- Adición de funciones al programa central (el que se encuentra en el servidor de BDD) del sistema. Como la contemplación de visitantes o de usuarios autorizados que olviden o extravíen su tarjeta de identificación. Encriptación de los datos que viajan a través de la red para garantizar la integridad de la validación de usuarios. Crear un entorno gráfico más amigable para la consulta de la base de datos generada por el tránsito de usuarios.
- Dotar al sistema con un circuito de retroalimentación que confirme que el acceso fue efectivamente concedido o negado, la implementación de éste deberá hacerse en la sección mecánica del sistema (actuador de la puerta).
- Habilitar la consulta vía Internet de la base de datos generada, que contiene la información de los usuarios que transitaron en un día determinado por las áreas del edificio. Además de almacenar para futuras consultas por lo menos las bases de datos generadas mes a mes.

IV.2. Comentarios

Estudiamos más a fondo el lenguaje de programación Dynamic C para poder manipular el hardware de la tarjeta de diseño incorporada en el módulo del RCM3200.

Mejoramos la base de datos de Microsoft Access®, para realizar una simulación más apegada a la realidad, esto es, un caso que ya podría implementarse en una empresa o institución.

Ampliamos las funciones del programa elaborado en Microsoft Visual Basic® que es la parte central de nuestro sistema, ya que recibe los datos enviados por la interfaz Ethernet, realiza la búsqueda en la base de datos y envía una señal de respuesta hacia la interfaz Ethernet.

Se acoplaron los datos de la lectora de código de barras a la interfaz Ethernet y se desarrolló el programa en Dynamic C correspondiente para almacenar estos datos en la memoria del módulo para ser enviados a través de Ethernet.

Finalmente añadimos las funciones necesarias a cada una de las partes para unir las y que el proceso de identificación, validación, registro y autorización o negación del acceso sea automático.

El proyecto realizado aprovecha las ventajas de la tecnología Ethernet y es de gran utilidad para instalaciones con diversas áreas y numerosos usuarios. No solo se provee de un ahorro económico a quien implementa el proyecto sino también de un buen nivel de seguridad.

La realización de este proyecto ha servido a cada uno de los integrantes del equipo para realizar un trabajo que combina diversas áreas de la Ingeniería Electrónica y en Comunicaciones, tal es el caso de programación de software, programación de hardware, manejo de puertos, acondicionamiento de señales eléctricas, y redes de datos. Mediante investigación y autoaprendizaje fuimos superando los obstáculos que encontramos en el desarrollo del proyecto.

A pesar de que la aplicación de este proyecto en sí resulta sencilla, y muy particular, su realización no ha sido trivial, ya que involucra un conocimiento fuerte de programación de software y hardware que ha sido obtenido por los integrantes del equipo a manera de autoestudio.

La reducción de costos que implica la implementación de este proyecto en una empresa que desee controlar el acceso a sus diversas áreas para sus empleados, nos hace pensar que podría resultar popular en el mercado.

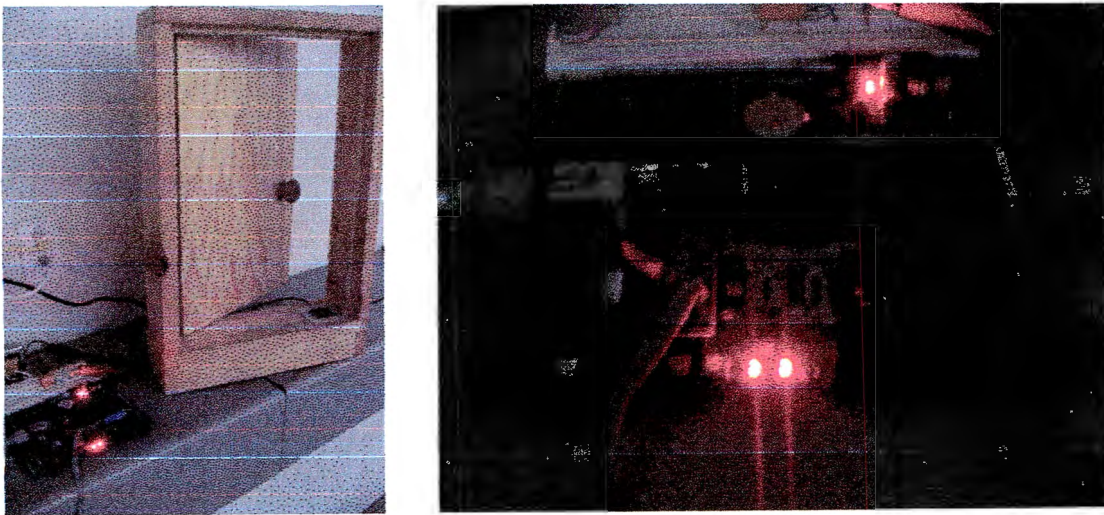


Figura 26. Prototipo Implementado

Sin duda el trabajo en equipo representa una de las herramientas principales para el desarrollo del proyecto, el compromiso y la dedicación deben ser cualidades de los integrantes.

Finalmente creemos que se cumplieron satisfactoriamente las metas planteadas para el proyecto:

- Lograr la comunicación entre la interfaz Ethernet y un servidor remoto.
- Automatizar el proceso de identificación de usuarios, desde la lectura de su tarjeta con código de barras hasta la concesión o negación de acceso.
- Llevar un registro de los usuarios que transitan en la diversas áreas de una empresa o institución.

V. REFERENCIAS

[1]

[2]

[3]

[4]

-

-

-

[5]

[6] Embedded Ethernet and Internet Complete, Axelson Jan., Editorial Pearson, Estados Unidos, 2003.

[7] Programación de Bases de Datos con Visual Basic 6, Smith Curtis, Amundsen Michael, Prentice Hall, México 1999, Págs. 47 – 70.

[8] Enciclopedia de Microsoft Visual Basic 6, Ceballos Javier, Alfaomega, México 2000, Págs. 125 – 142.

[9] PC interno 5, Programación de Sistemas, Tischer M., Jennrich B., Grupo Editor Alfa Omega, México, 1998, Págs. 242 – 296.

[10] "Código de barras," *Enciclopedia Microsoft® Encarta® 2000*. © 1993-1999 Microsoft Corporation.

[11]



[12]



VI. ANEXOS

VI.1. Programa de lectura de datos del lector de código de barras

```

/*****
lector_tarjeta.c

```

Este programa demuestra el uso de un lector de códigos de barras, basándose en la librería "card_reader.lib" presente para el microprocesador RCM3000. Dicha librería se ha omitido en este anexo por ser muy extensa.

Es necesario contar con un lector de código de barras conectado a las líneas de puerto 1, 2 y 3 de la interfaz como sigue:

Rabbit	Card Reader	
PG1	/CARD PRESENT	Proveniente del pin de reloj del lector
PG2	TRACK1 DATA	Proveniente del pin de datos del lector
PG3	TRACK1 STROBE	Proveniente del pin de reloj del lector
PG4	TRACK2 DATA	NO se usa
PG5	TRACK2 STROBE	No utilizado
PG6	TRACK3 DATA	No utilizado
PG7	TRACK3 STROBE	No utilizado

Puesto que el lector de código de barras utilizado opera únicamente con un track de datos, las demás líneas de puerto no serán utilizadas.

El programa decodificará los bits provenientes del lector y los convertirá a su correspondiente carácter ASCII. Para ver la configuración de la conexión realizada, refiérase a la figura 13.

```

*****/

```

```

// prioridad para interrupciones de la lectora, usando el puerto G
#define CR_IPLEVEL 1
#define CR_USEPORTG

// librería principal del programa
#include "card_reader.lib"

// define el número de pistas de la lectora, el lector utilizado maneja un solo track
#define READER_TRACKS 1

// define el lugar para almacenar los bits leídos
#define TRACK_BUF_SIZE 32
char track_buffer[READER_TRACKS][TRACK_BUF_SIZE];

main()
{
    int i;
    int bit_count;
    char track_string[150];

    CRinit(READER_TRACKS);

```

```
// asigna buffer al track para la lectura de datos
for(i = 0; i < READER_TRACKS; i++)
{
    CRsetBuffer(i, track_buffer[i], TRACK_BUF_SIZE);
}

while(1)
{
    CRstartRead();
    printf("Esperando tarjeta.\n");
    while(!CRcheckRead());

    for(i = 0; i < READER_TRACKS; i++)
    {
        bit_count = CRgetBuffer(i, NULL);
        printf("Track %d(%d bits)\n", i+1, bit_count);
        if(CRalphaDecode(track_buffer[i], bit_count, track_string))
        {
            printf("\nSe trata de datos ALPHA\n%s\n", track_string);
        }
        else if(CRbcdDecode(track_buffer[i], bit_count, track_string))
        {
            printf("\nSe trata de datos BCD\n%s\n", track_string);
        }
        else
        {
            printf("no se pueden decodificar los bits %s\n", track_string);
        }
        printf("\n");
    }
    printf("\n");
}
}
```

VI.2 Programa Cliente de la Interfaz Ethernet

```

/*****
Cliente.c

```

Este programa se basa en el controlador de la serie RCM3200.

La librería RCM3200.lib, contiene una función llamada brdlnit() que inicializa los pines de los puertos. Es necesario modificar dicha función según el tipo de aplicación de red.

Descripción

=====

Este programa funciona como un cliente que espera los datos provenientes del lector de códigos de barras, envía el paquete previamente almacenado y espera la respuesta del servidor remoto que definirá el estado de los LEDs así como de uno de los puertos para controlar el circuito mecánico de apertura y cierre de la puerta.

Los tiempos de transferencia pueden ser vistos en una ventana de Visual Studio.

Instrucciones

=====

Modificar los siguientes parámetros: TCPCONFIG, REMOTE_IP, REMOTE_PORT y MY_PORT según sea necesario para la red en que se pretende correr este programa. Es necesario verificar el funcionamiento de cada librería para poder hacer dichas modificaciones.

```

/*****
#class auto

#use rcm3200.lib          //librería usada

##define DCRTCP_DEBUG    // desplegar información tcp/ip
##define DCRTCP_VERBOSE // dar de alta funciones tcp/ip

/*****
* Configuración
* Los campos en esta sección deben ser modificador según los parámetros de la red .
*
/*****
*/
* Seleccione la configuración TCP/IP adecuada. Ver librería LIB\TCPIP\TCP_CONFIG.LIB para saber el
* funcionamiento.
*/

#class auto
#use rcm3200.lib
#define TCPCONFIG 1
#define MY_PORT          888

```



```

#define ETH_MTU          1000
#define BUFF_SIZE      (ETH_MTU-40)
#define TCP_BUF_SIZE ((ETH_MTU-40)*4)

#memmap xmem
#use "dcrtcp.lib"

#define INCOMING_IP      0           // aceptar todas las IP's
#define INCOMING_PORT   0           // Aceptar todos los puertos
#define TIME_OUT        5000       // timer

#define RED              "\x1b[31m"
#define GREEN            "\x1b[32m"
#define BLUE              "\x1b[34m"

#define DS1 1           //led, puerto G bit 6
#define DS2 2           //led, puerto G bit 7

tcp_Socket sock;
int bytes;
long statime;
static char buff[BUFF_SIZE];
int channel;
int i,j;

////////////////////////////////////

void digOut(int channel, int onoff)
{
    BitWrPortl(PGDR, &PGDRShadow, onoff, channel+5);
}

int receive_packet(void)
{
    /* receive the packet */
    bytes = sock_fastread(&sock,buff,BUFF_SIZE);

    switch(bytes)
    {
        case -1:
            return 4;
        case 0:
            return 2;
        default:
            statime = MS_TIMER+TIME_OUT;
            for(channel = DS1; channel <=DS2 ; channel++)
            {
                digOut(channel, 1);
            }
    }
}

```

```

    if (buff[0]=='1')
    {
        digOut(DS1, 0);
        printf("ENCONTRADO");
        costate{
            for(i=0;i<250;i++)
                for(j=0;j<1000;j++)
                {
                    DelayMs(1000);}
        }

        digOut(DS1, 1);
    }
    else
    {
        digOut(DS2,0);
        printf("NO HAY ACCESO");
        costate{
            for(i=0;i<250;i++)
                for(j=0;j<1000;j++)
                {
                    DelayMs(1000);}
        }

        digOut(DS2, 1);
    }
    return 2;
}

int send_packet(void)
{

    memset(buff,'S',BUFF_SIZE);
    buff[BUFF_SIZE-1] = '\n';

    bytes = sock_fastwrite(&sock,buff,BUFF_SIZE);
    switch(bytes)
    {
        case -1:
            return 3;
        default:
            statetime = MS_TIMER+TIME_OUT;
            return 2;
    }
}

void main()
{

    int state,status;

```

```

brdlnit();
    state = 0;

for(channel = DS1; channel <=DS2 ; channel++)
    {
        digOut(channel, 1);
    }
sock_init();

while(1)
{
    switch(state)
    {
        case 0:
if(!tcp_listen(&sock,MY_PORT,INCOMING_IP,INCOMING_PORT,NULL,0))
            printf("%sError al abrir el socket!\n",RED);
            else
            {
                printf("%siniciando sesion\n",GREEN);
                statetime = MS_TIMER+TIME_OUT;
                state++;
            }
            break;
        case 1:
            if(sock_established(&sock))
            {
                printf("%socket establecido\n",BLUE);
                statetime = MS_TIMER+TIME_OUT;
                state++;
            }
            else if ((long)(MS_TIMER-statetime) > 0)
                state = 1;
            break;
        case 2:
            state = receive_packet();
            if ((long)(MS_TIMER-statetime) > 0)
                state = 2;
            break;
        case 3:
            state = send_packet();
            if ((long)(MS_TIMER-statetime) > 0)
                state = 4;
            break;
        case 4:
            printf("%sABORTAR SOCKET\n",RED);
            sock_abort(&sock);
            state = 0;
    }
    status = tcp_tick(&sock);
}
}

```

VI.3. Programa Terminal Remota – Interfaz

```

/*****
Control_leds_web.c

```

Descripción

=====

El programa se corre en una página Web para controlar desde allí a dos dispositivos, en este caso LEDs. Dos botones se crearon para simular un interruptor que los prenda o apague desde una terminal remota.

Instrucciones

=====

1. Hacer las configuraciones necesarias, según la red.
2. Compilar y correr el programa.
3. Desde un equipo remoto abrir el navegador Web.
4. Controlar el encendido y apagado de los LEDs de la interfaz con los botones que aparecen en el navegador

```

*****/
#class auto

```

```

#use rcm3200.lib          //librería usada para el programa

```

```

#define DS1 0x40          //led, port G bit 6 bitmask

```

```

#define DS2 0x80          //led, port G bit 7 bitmask

```

```

/*****
* Configuración *
* Los campos en esta sección deben según los parámetros de su red. *
*****/

```

```

#define TCPCONFIG      5

```

```

#define TCP_BUF_SIZE  2048

```

```

#define HTTP_MAXSERVERS  2

```

```

#define MAX_TCP_SOCKET_BUFFERS  2

```

```

#define REDIRECTHOST    _PRIMARY_STATIC_IP

```

```

//#define REDIRECTHOST  "my.host.com:8080"

```

```

//#define REDIRECTHOST  "demo.zweng.com:8240"

```

```

#define REDIRECTTO      "http://" REDIRECTHOST ""

```

```

#memmap xmem

```

```

#use "dcrtcp.lib"

```

```
#use "http.lib"

#import "samples/rcm3200/tcpip/pages/browseled.shtml"    index_html
#import "samples/rcm3200/tcpip/pages/rabbit1.gif"       rabbit1_gif
#import "samples/rcm3200/tcpip/pages/logo.jpg"         logo_jpeg
#import "samples/rcm3200/tcpip/pages/ledon.gif"         ledon_gif
#import "samples/rcm3200/tcpip/pages/ledoff.gif"        ledoff_gif
#import "samples/rcm3200/tcpip/pages/button.gif"        button_gif
#import "samples/rcm3200/tcpip/pages/showsrc.shtml"     showsrc_shtml
#import "samples/rcm3200/tcpip/browseled.c"             browseled_c

const HttpType http_types[] =
{
  { ".shtml", "text/html", shtml_handler}, // ssi
  { ".html", "text/html", NULL},          // html
  { ".cgi", "", NULL},                    // cgi
  { ".gif", "image/gif", NULL}
};

char led1[15];
char led2[15];

int led1toggle(HttpState* state)
{
  if (strcmp(led1,"ledon.gif")==0)
    strcpy(led1,"ledoff.gif");
  else
    strcpy(led1,"ledon.gif");

  cgi_redirectto(state,REDIRECTTO);
  return 0;
}

int led2toggle(HttpState* state)
{
  if (strcmp(led2,"ledon.gif")==0)
    strcpy(led2,"ledoff.gif");
  else
    strcpy(led2,"ledon.gif");

  cgi_redirectto(state,REDIRECTTO);
  return 0;
}

const HttpSpec http_flashspec[] =
{
  { HTTPSPEC_FILE, "/", index_html, NULL, 0, NULL, NULL},
  { HTTPSPEC_FILE, "/index.shtml", index_html, NULL, 0, NULL, NULL},
  { HTTPSPEC_FILE, "/showsrc.shtml", showsrc_shtml, NULL, 0, NULL, NULL},
  { HTTPSPEC_FILE, "/logo.jpg", logo_jpeg, NULL, 0, NULL, NULL},
  { HTTPSPEC_FILE, "/rabbit1.gif", rabbit1_gif, NULL, 0, NULL, NULL},
}
```

```

{ HTTPSPEC_FILE, "/ledon.gif", ledon_gif, NULL, 0, NULL, NULL},
{ HTTPSPEC_FILE, "/ledoff.gif", ledoff_gif, NULL, 0, NULL, NULL},
{ HTTPSPEC_FILE, "/button.gif", button_gif, NULL, 0, NULL, NULL},

{ HTTPSPEC_FILE, "browseled.c", browseled_c, NULL, 0, NULL, NULL},

{ HTTPSPEC_VARIABLE, "led1", 0, led1, PTR16, "%s", NULL},
{ HTTPSPEC_VARIABLE, "led2", 0, led2, PTR16, "%s", NULL},

{ HTTPSPEC_FUNCTION, "/led1tog.cgi", 0, led1toggle, 0, NULL, NULL},
{ HTTPSPEC_FUNCTION, "/led2tog.cgi", 0, led2toggle, 0, NULL, NULL},
};

void update_outputs()
{
    auto int value;

    value=PGDRShadow&0x3F;

    /* update O0 */
    if (strcmp(led1,"ledon.gif"))
        value|=DS1;

    /* update O1 */
    if (strcmp(led2,"ledon.gif"))
        value|=DS2;

    WrPortI(PGDR, &PGDRShadow, value);
}
main()
{

    brdlnit();

    strcpy(led1,"ledon.gif");
    strcpy(led2,"ledoff.gif");

    sock_init();
    http_init();
    tcp_reserveport(80);

    while (1)
    {
        update_outputs();
        http_handler();

    }
}

```

VI.4. Programa Central del Servidor de BDD

Este programa establece una sesión de red con un cliente remoto, recibe los datos de la matricula de un usuario, busca esta matricula en una base de datos y contesta hacia el cliente, si el usuario se encuentra registrado en ella o no

```
'/// Declaración de variables
Option Explicit
Private db As Database
Private enct As Recordset
Private inst As Recordset
Dim resultado As String
Dim resultado1 As String
Dim resultado2 As String
Dim registro As String
Dim enviar As String
Const IP As String = "10.10.6.100" ' Dirección IP de la tarjeta Rabbit
Const sPathBase As String = "C:\BD-TEC.mdb" ' Ruta de acceso de la base de datos

'///Subrutina que responde al presionar enter dentro del programa

Private Sub Busqueda_KeyPress(KeyAscii As Integer)

If KeyAscii = vbKeyReturn Then ' Si la tecla presionada es enter.
    On Error Resume Next

    KeyAscii = 0

    Buscar ' Ir a la subrutina Buscar
End If

End Sub

' ///Función de búsqueda de la matricula leída.

Private Sub Buscar()
Dim Matr As String
Matr = Busqueda.Text

Set enct = db.OpenRecordset("SELECT * from Alumnos where Matricula=" & Matr & "" ", dbOpenDynaset)
' El recordset enct se compone del registro que coincida con la matricula leída.

With enct
If (.BOF And .EOF) Then ' Si enct no tiene registros, se borran las variables que contienen datos
    resultado1 = ""
    resultado2 = ""
```

```

resultado = " "
registro = 0
enviar = "0000000"
Winsock1.SendData enviar ' Se envía una cadena de 0's a la tarjeta.

```

```

Else
.MoveFirst ' Si enct tiene registros nos posicionamos en el registro.
resultado = .Fields("Matricula")
resultado1 = .Fields("Nombre") + " " + .Fields("Ap_Paterno") + " " + .Fields("Ap_Materno")
resultado2 = .Fields("Observaciones") ' Se guardan los datos en las variables.
registro = 1
enviar = "11111111"
Winsock1.SendData enviar ' Se envía una cadena de 1's a la tarjeta.

```

```

Set inst = db.OpenRecordset("Registro") ' Se abre la tabla Registro.
With inst
.AddNew ' Se agregan los campos matricula, hora, fecha y área de ingreso.
.Fields("Matricula") = Matr
.Fields("Hora y Fecha") = Now
.Fields("Area") = "1"
.Update ' Se actualiza la base de datos.
End With
End If

```

```

End With
MatriculaB.Text = resultado
NombreB.Text = resultado1
ObservacionesB.Text = resultado2
Text1.Text = registro ' Se actualizan los datos en los textbox de la aplicación.

```

```

Busqueda.SetFocus
Busqueda.SelStart = 0
Matr = Busqueda.Text
Busqueda.SelLength = Len(Mat) ' El foco del programa se mantiene en el textbox busqueda.

```

```

End Sub

```

```

Private Sub Command1_Click() ' Al presionar el botón Buscar matricula se llama a la subrutina Buscar.
Buscar
End Sub

```

```

Private Sub Limpiar_Click() ' Limpia las variables y los datos en todos los textbox de la aplicación.
Dim Matr As String
resultado1 = " "
resultado2 = " "
resultado = " "
registro = 0

```

```

Busqueda.SetFocus

```



```

Busqueda.SelStart = 0
Matr = "Introduce Matricula"
Busqueda.Text = Matr
Busqueda.SelLength = Len(Matr)
MatriculaB.Text = resultado
NombreB.Text = resultado1
ObservacionesB.Text = resultado2
Text1.Text = registro          ' El foco del programa se mantiene en el textbox busqueda
End Sub

```

```

Private Sub Command3_Click() ' Agregar un registro a la base de datos.
    Data1.Recordset.AddNew
End Sub

```

```

Private Sub Command4_Click() ' Eliminar un registro de la base de datos
Data1.Recordset.Delete
End Sub

```

```

Private Sub Data1_Validate(Action As Integer, Save As Integer)
Dim Matr As String

```

```

Busqueda.SetFocus
Busqueda.SelStart = 0
Matr = Busqueda.Text
Busqueda.SelLength = Len(Matr) 'El foco del programa se mantiene en el textbox busqueda.
End Sub

```

```

Private Sub Form_Activate()
Dim Matr As String
Busqueda.SetFocus
Busqueda.SelStart = 0
Matr = Busqueda.Text
Busqueda.SelLength = Len(Matr) 'El foco del programa se mantiene en el textbox busqueda.
End Sub

```

```

Private Sub Form_Load()
Set db = OpenDatabase(sPathBase) ' Se abre la base de datos
Winsock1.RemoteHost = IP
Winsock1.Connect ' Se abre el socket con la dirección IP especificada
End Sub

```

```

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim datos As String ' Al llegar datos se guardan en la variable datos.
    Winsock1.GetData datos
End Sub

```

VI.5. Especificaciones Lector de Códigos de Barras MS6720 – KW

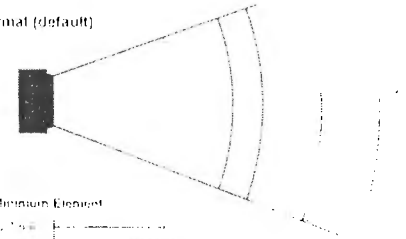
MS6720

OPERATIONAL

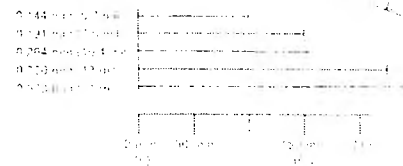
Light Source	Visible Laser Diode 650 nm \pm 10 nm
Laser Power	0.681 mW (peak)
Depth of Scan Field (programmable)	0 mm - 203 mm (0" - 8") for 0.33 mm (13 mil) bar code at default setting
Width of Scan Field	50.8 mm (2") @ face; 178 mm (7") @ 203 mm (8")
Scan Speed	1000 scan lines per second
Scan Pattern	5 fields of 4 parallel lines (omnidirectional)
Number of Scan Lines	20
Minimum Bar Width	0.145 mm (5.7 mil)
Decode Capability	Autodiscriminates all standard bar codes, including RSS; for other symbologies call Metrologic
System Interfaces	PC Keyboard Wedge, RS232, OCIA, Light Pen Emulation, IBM 468x, 468x, USB (low speed and full speed), Stand Alone Keyboard
Print Contrast	35% minimum reflectance difference
Number Characters Read	Up to 80 data characters
Roll, Pitch, Yaw	360°, 60°, 60°
Beeper Operation	3 tones or no beep
Indicators (LED)	Red = laser on, ready to scan Green = good read

Scan Area

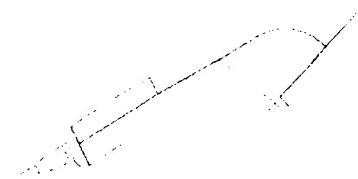
Normal (default)



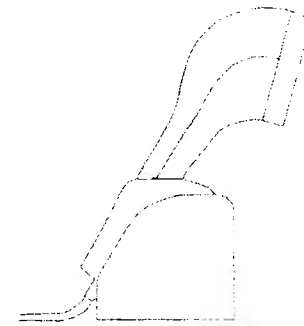
Bar Code Minimum Element



Dimensions



Standalone Stand



Auto-stable Stand

MECHANICAL

Height	219 mm (8.6")
Depth-Handle	28 mm (1.1")
Depth-Head	75 mm (2.9")
Width-Handle	48 mm (1.9")
Width-Head	77 mm (3.0")
Footprint On Stand	100 mm x 135 mm (3.9" x 5.3")
Weight	380 g (13.4 oz)
Termination	10 pin modular, RJ45 connector
Cable	Standard 2.1 m (7') straight; optional 2.7 m (9') coiled; for other cables call Metrologic

INPUT

Input Voltage	5 VDC \pm 0.25 V
Power	1.1 W @ 5V
Operating Current	220 mA @ 5V
DC Transformers	Class 2; 5 VDC @ 300 mA
Laser Class	IEC 60825-1:1993--A1:1997 - A2:2001 Class 1
FMC	FCC, ICES-003 & EN55022 Class B

ENVIRONMENTAL

Operating Temperature	0°C to 40°C (32°F to 104°F)
Storage Temperature	-40°C to 60°C (-40°F to 140°F)
Humidity	5% to 95% relative humidity, non-condensing
Light Levels	Up to 4342 Lux (450 footcandles)
Shock	Designed to withstand 1 m (3.1') drop
Contaminants	Sealed to resist airborne particulate contaminants
Ventilation	None required

MS6720 is available in the following colors



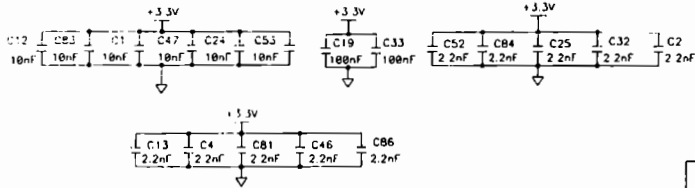
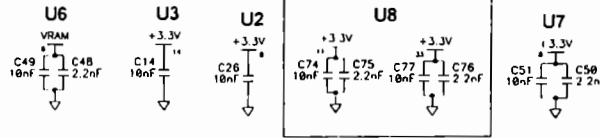
Pearl White Light Grey Black



VI.6. Diagrama Esquemático del Módulo RCM3200

POWER TABLE

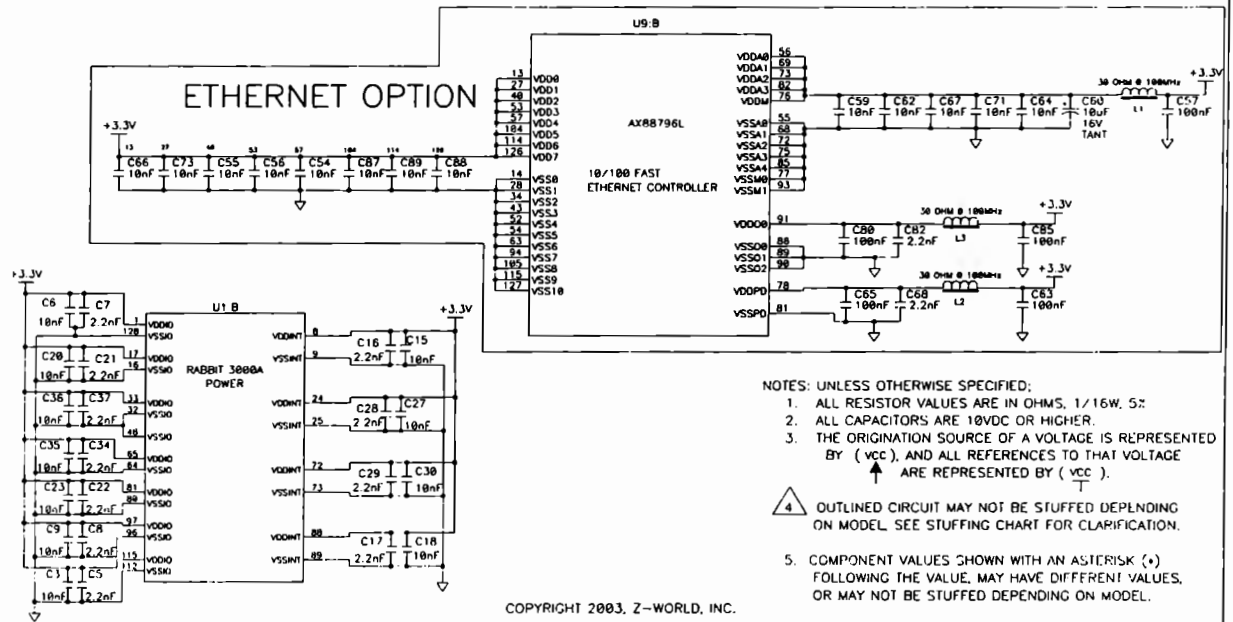
Ref Des	Device	GND	+3.3V	VRAM
U2	74LCX245	10	20	
U3	74ALVC32	7	14	
U6	SRAM 1	24	8	8
U7	FLASH	24	8	
U8	SRAM 2	34,12	33,11	



STUFFING TABLE

CIRCUIT	PART	RCM3200	RCM3210	RCM3220
SRAM 1	U6	256K	128K	256K
SRAM select	JP5	1-2	1-2	1-2
SRAM 2	U8	512K	NOT INSTALLED	512K
SRAM 2 CS	JP2	2-3	NOT INSTALLED	2-3
FLASH	U7	512K	256K	512K
FLASH select	JP3	2-3	2-3	2-3
BANK SELECT	JP4	1-2	1-2	1-2
LXT DATA BUS	JP1	2-3	2-3	1-2
POWER TO VRAM	R41	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
OSCILLATOR CIRCUIT	Y1	22.12 MHz	14.74 MHz	22.12 MHz
	C10, C11, R15	INSTALLED	INSTALLED	INSTALLED
PORT D OPTION	R29, R37, R39, R40	INSTALLED	INSTALLED	INSTALLED
LED OPTION	R67, R70	INSTALLED	INSTALLED	NOT INSTALLED
FUTURE OPTION	R45, R46, R53, R57, R74	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
INTERRUPT OPTION	R61	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
RTC	R28	INSTALLED	INSTALLED	INSTALLED
CS OPTION	U11, R80	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
	R58	NOT INSTALLED	NOT INSTALLED	NOT INSTALLED
ENCLOSURE LABEL	C74, C75, C76, C77	INSTALLED	NOT INSTALLED	INSTALLED
ENCLOSURE LABEL	Labels 1 and 3	INSTALLED	INSTALLED	NOT INSTALLED

ETHERNET OPTION

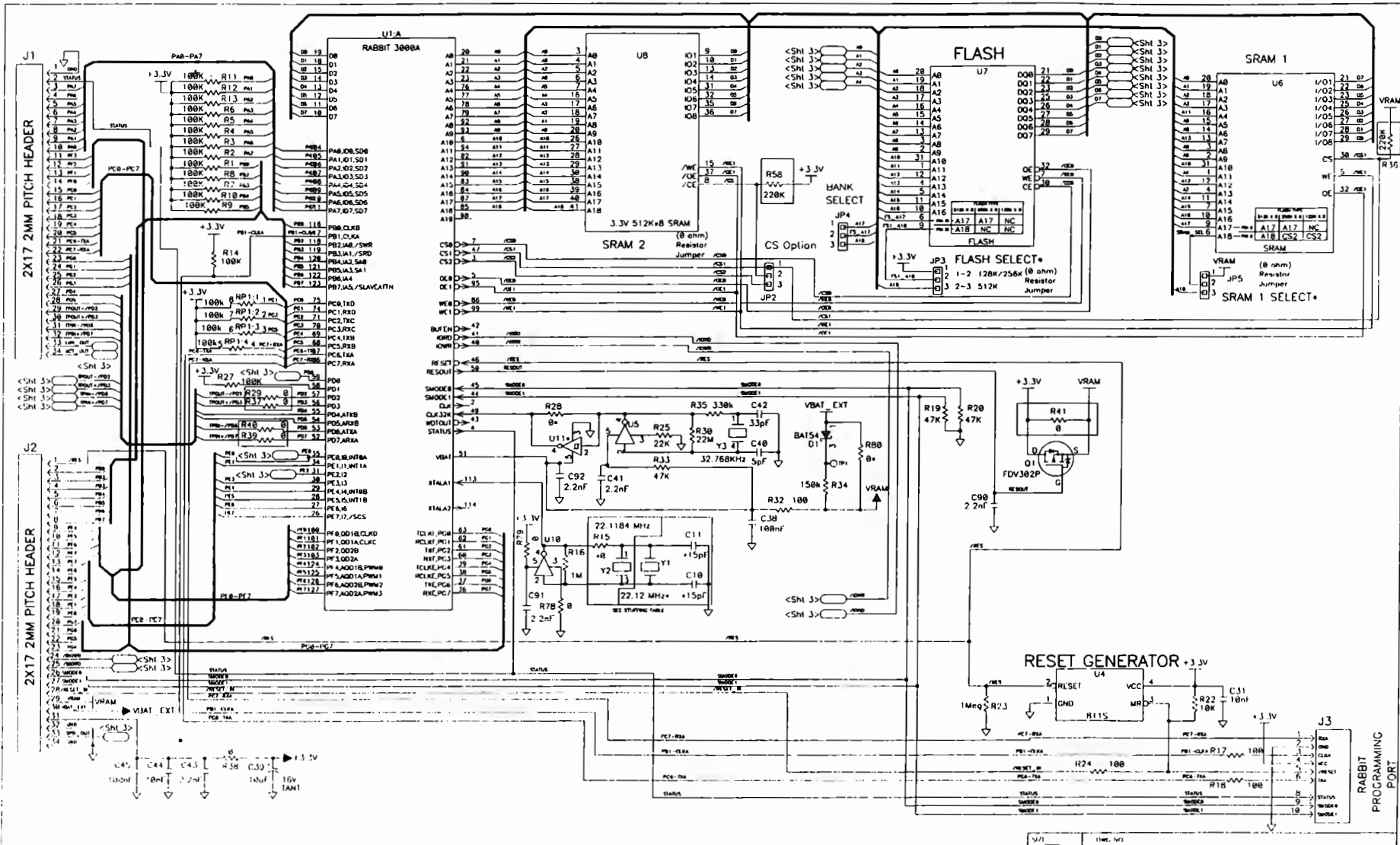


- NOTES: UNLESS OTHERWISE SPECIFIED:
1. ALL RESISTOR VALUES ARE IN OHMS, 1/16W, 5%.
 2. ALL CAPACITORS ARE 10VDC OR HIGHER.
 3. THE ORIGINATION SOURCE OF A VOLTAGE IS REPRESENTED BY (VCC), AND ALL REFERENCES TO THAT VOLTAGE ARE REPRESENTED BY (VCC).
- ⚠️ OUTLINED CIRCUIT MAY NOT BE STUFFED DEPENDING ON MODEL. SEE STUFFING CHART FOR CLARIFICATION.
5. COMPONENT VALUES SHOWN WITH AN ASTERISK (*) FOLLOWING THE VALUE, MAY HAVE DIFFERENT VALUES, OR MAY NOT BE STUFFED DEPENDING ON MODEL.

REVISION HISTORY		REVISION APPROVAL				
REV	ECCO	DESCRIPTION	APPROVAL DATE	DESIGNER	APPROVAL DATE	
A	E12016	INITIAL RELEASE	DM	7/26/02	KIS	7/26/02
B	E12037	CHANGE VALUE OF Y1 TO 22.12MHZ	DM	8/31/02	KIS	8/13/02
C	E12092	CHANGE R25,31 FROM 22K TO 49.9 AND 0, FIX OFF-SHEET REFERENCE. FIX UNSTUFFED RTC COMP	DM	10/2/02	KIS	10/2/02
D	E12104	ADDED U10, R67, R70, R78, R79, CHANGED RTC COMPONENT VALUES	DM	10/21/02	KIS	10/21/02
E	E12183	ADDED U11, CHANGED RTC COMPONENT VALUES	DM	2/4/03	KIS	2/4/03
F	E12263	ADDED RCM3220 OPTION, CORRECTED J4 PIN-OUT	DM	4/9/03	KF	4/9/03
G	E12719	Changed R3000 to R3000A, removed U11, added R28. Changed RTC resistor values, Changed R42 to ferrite	DM	4/7/04	KF	4/7/04

COPYRIGHT 2003, Z-WORLD, INC.

APPEND THE FOLLOWING DOCUMENTS WHEN CHANGING THIS DOCUMENT.	DRAWING CONTENT:	DATE:	<p>SCHEMATIC DIAGRAM</p> <p>RCM3200 SERIES</p> <p>CORE MODULE</p>	<p>2900 SPAFFORD ST. DALLAS, TX 75246 510-557-4616</p>	
	DRAWN BY: DARRIN MUSGROVE	7/22/02			
	REVIEWED BY: DARRIN MUSGROVE	4/7/04			
	APPROVALS: INITIAL RELEASE				
	DESIGNED BY: DARRIN MUSGROVE	7/26/02	<p>B 090-0152</p>	<p>3</p>	
	ENGINEERING MANAGER: K. MATTHEWS	7/26/02			
SIGNATURES	DATE	REV	NONE	RELEASED: 26JUL02	REV: 1 OF 3



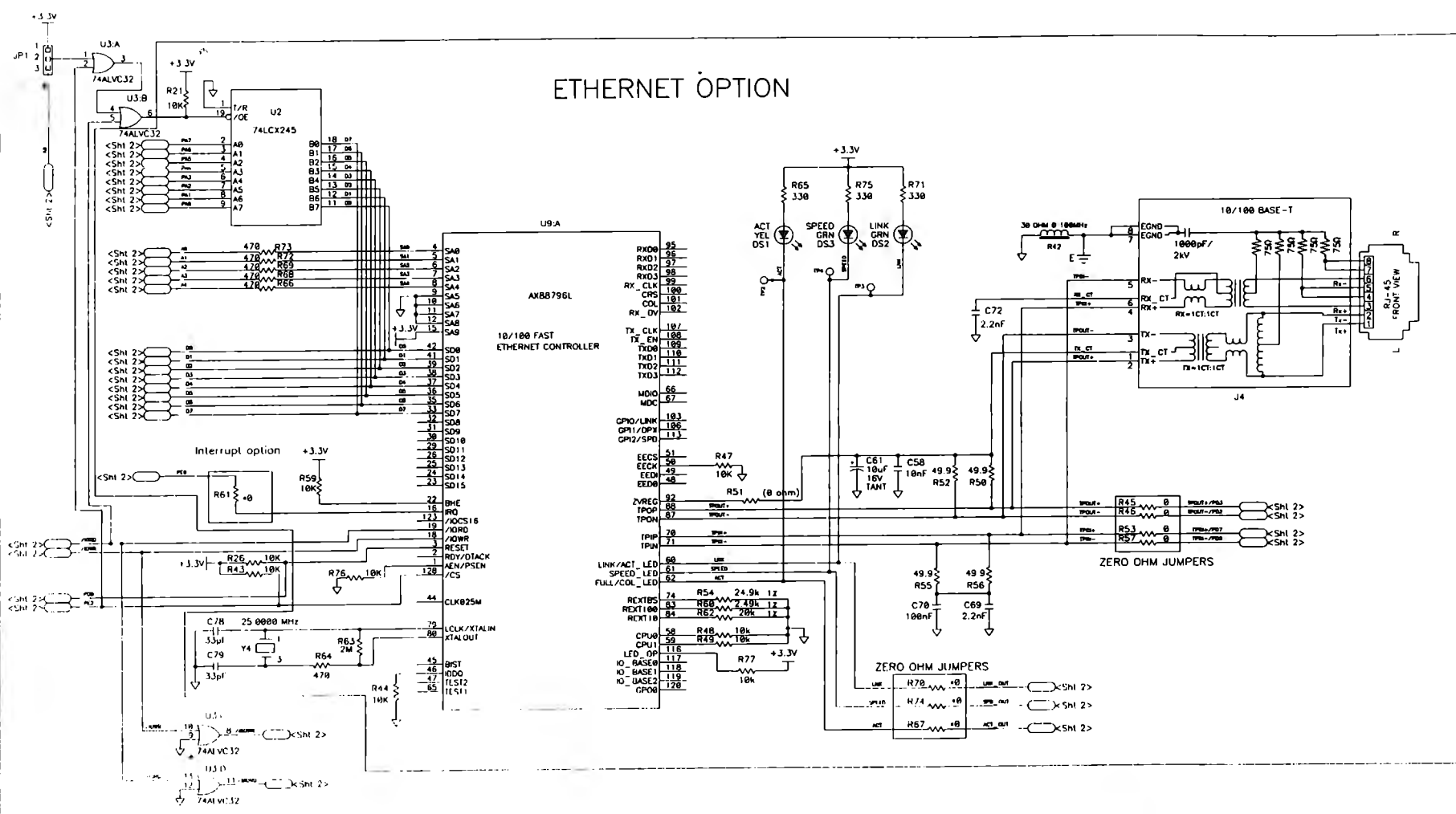
4/7/04

B 090-0152

COPYRIGHT 2003, Z-WORLD, INC.

REV: NONE P/N: 1/16 6 2 1 3

ETHERNET OPTION



47794

REV	DWG. NO.		
B	090-0152		
CALC	NONE	REV. LTR	6
		SHEET	3 OF 3

VII. PÓSTER



Sistema de Control de Acceso con Comunicación Ethernet

28 DE ABRIL DE 2005

CAMPUS CIUDAD DE MÉXICO

PROYECTOS DE INGENIERÍA II

INTEGRANTES
 JOSE ANTONIO LLANOS TORRES 905220
 LEONARDO RAMIREZ TORRES 969789

COORDINADOR DE LA MATERIA
 DR. RAÚL CRESPO SAUCEDO

ASESOR
 M.C. ISRAEL MACÍAS HIDALGO

OBJETIVO

- ▶ Desarrollar un sistema de control de acceso para validación, captura y registro de información de usuarios en un edificio inteligente.
- ▶ Implementar un dispositivo electrónico que recibe un paquete de datos de una lectora de tarjetas con código de barras para la identificación de un usuario.
- ▶ Utilizar una interfaz Ethernet en cada punto de acceso que se comunica con un servidor de base de datos permitiendo controlar el ingreso a las diversas áreas con una sola computadora.

PROBLEMÁTICA

- ▶ Inseguridad
- ▶ Acceso no controlado
- ▶ Seguimiento ineficaz de personal
- ▶ Costo elevado de equipos y personal

JUSTIFICACIÓN

Este proyecto pretende evitar la necesidad de tener una computadora o un vigilante en cada área de un edificio que se desea acceder. La comunicación Ethernet permite controlar el acceso a todas las áreas existentes de un edificio con una sola computadora. Otras de las facilidades que brinda el dispositivo es un uso eficiente de la energía (por su bajo consumo) y el ahorro de espacio ya que es pequeño y de fácil implementación en comparación con sistemas similares.

RESULTADOS

- ▶ Comunicación satisfactoria entre una interfaz Ethernet y una terminal remota.
- ▶ Elaboración del programa "Servidor" del sistema en una terminal remota.
- ▶ Elaboración del programa "Cliente" en la red RCM3200.
- ▶ Adaptación de la señal proveniente de la lectora de código de barras.
- ▶ Identificación y registro satisfactorios de un usuario, concediendo o negando su acceso.

CONCLUSIONES

- ▶ El proyecto incursiona en las áreas de redes y comunicaciones, siendo la base de un sistema completo de automatización para edificios inteligentes.
- ▶ Aprovecha las ventajas de la tecnología Ethernet, y es de gran utilidad para instalaciones con diversas áreas y numerosos.
- ▶ Se provee de un ahorro económico y de un buen nivel de seguridad a quien lo implemente.

SISTEMA PROPUESTO



LECTORA DE TARJETAS DE IDENTIFICACIÓN PERSONAL CON CÓDIGO DE BARRAS

ETHERNET

SERVIDOR DE BASE DE DATOS REMOTO

PUERTA CONTROLADA