



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**“SISTEMA DE CONTROL DE INGRESO DE VEHICULOS”
INFORME DE MATERIA DE GRADUACION**

**Previa la obtención del Título de:
INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN
SISTEMAS TECNOLÓGICOS
INGENIERO EN ELECTRONICA Y TELECOMUNICACIONES**

**PRESENTADA POR:
FOULTON TOMAS QUIMI LINO.
EDUARDO EMANUEL BLACIO DIAZ**

GUAYAQUIL - ECUADOR

2009

AGRADECIMIENTO

*A Dios, a nuestros padres y a
todos aquellos que
siempre confiaron en
nosotros.*

DEDICATORIA

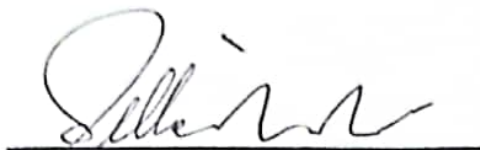
A Dios.

Eduardo Emanuel Blacio Diaz

A Dios y a mis padres.

Foulton Tomás Quimí Lino

TRIBUNAL DE GRADO



Ing. Hugo Villavicencio.

DELEGADO



Dr. Ing. Carlos Valdivieso A.

DIRECTOR DE TESIS

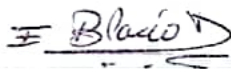
DECLARACIÓN EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestas en esta tesis, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral"

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Foulton Tomás Quimí Lino



Eduardo Emanuel Blacio Díaz

INDICE GENERAL

INTRODUCCION

I. Alcance del proyecto	1
II. Antecedentes	2
III. Identificación del problema	2
IV. Descripción breve de la solución	3
V. Razón para escoger las herramientas usadas en este proyecto	4
VI. Soluciones similares	5

CAPITULO 1

Sustento Teórico

1.1 Ambiente de LABVIEW utilizado en el proyecto	7
1.1.1 Funciones básicas empleadas	7
1.1.2 Funciones avanzadas empleadas	12
1.2 El microcontrolador	17
1.2.1 Características específicas del PIC18F4520 utilizadas en el proyecto	18
1.3 Comunicación Ethernet	23
1.3.1 El ENC28J60	23
1.3.2 Direccionamiento IP y MAC	24

1.3.3	Diferencias entre UDP y TCP	27
1.3.4	Interfaz con PIC18F4520	30
1.4	El ambiente MikroBasic	31
1.4.1	Librerías utilizadas	31
1.4.2	Programa de desarrollo utilizado MikroBasic	34
1.5	Manejo del motor activación de barrera mecánica	35
1.5.1	Investigación sobre Tipos de motores disponibles en el mercado	35
1.5.2	Recomendación del motor de paso para ser usado en este proyecto (para evaluar costos)	36
1.5.3	Comunicación del motor con el microcontrolador (El ULN2003)	38
1.6	Tecnología de Sensores Infrarrojos	39
1.6.1	La emisión y recepción de energía infrarroja	40
1.7	Tecnología RFID	41
1.7.1	Funcionamiento RFID	41
1.7.2	TAGS Pasivos	43
CAPITULO 2		
Construcción del proyecto		
2.1	Diagrama de bloques	45
2.1.1	Explicación breve del funcionamiento del circuito general	45
2.2	Descripción detallada de cada bloque	47

2.2.1	Bloque PIC/ETHERNET	47
2.2.1.1	Captura de información externa mediante botones y RFID	47
2.2.1.2	Esquemático de la conexión ETHERNET	48
2.2.1.3	Explicación del funcionamiento del circuito	50
2.2.1.4	Porción del Programa del PIC18F4520 en MIKROBASIC para la comunicación Ethernet	50
2.2.1.5	Explicación del programa del PIC18F4520	59
2.2.2	Bloque LabView: Procesamiento de la información en LABVIEW 8.5	63
2.2.2.1	Ingreso de clave de acceso	64
2.2.2.2	Panel frontal del programa en LABVIEW	65
2.2.2.3	Explicación de cada elemento del panel frontal	66
2.2.2.4	Diagrama de bloques del programa en LABVIEW	68
2.2.2.5	Panel frontal del reporte en LABVIEW	82
2.2.2.6	Explicación del panel frontal del reporte	82
2.2.2.7	Diagrama de bloque del reporte en LABVIEW	84
2.2.3	Bloque de Barrera	87
2.2.3.1	Captura de información proveniente del ordenador vía Ethernet y salida de control del motor de paso	87
2.2.3.2	Esquemático del circuito de fuerza	88
2.2.3.3	Explicación del funcionamiento del circuito	88

2.2.3.4	Porción del Programa del PIC18F4520 para control del motor de paso	89
2.2.3.5	Explicación detallada del programa del PIC18F4520 para controlar los motores	97
2.2.4	Bloque de Sensores	99
2.2.4.1	Detección mediante sensores	99
2.2.4.2	Conexión de los sensores al circuito de fuerza	102
2.2.4.3	Explicación del funcionamiento del circuito	103

CAPITULO 3

Simulación, implementación y pruebas

3.1	Pistas del PCB construidos	105
3.2	Resultados experimentales	106

CONCLUSIONES Y RECOMENDACIONES

GLOSARIO DE TÉRMINOS

ANEXOS

BIBLIOGRAFIA

ABREVIATURAS

ESPOL	Escuela Superior Politecnica del Litoral
IR	Infrarrojo
IRDA	Infra Red Asociation
LED	Diodo emisor de luz
PaP	Paso a paso
RFID	Identificación por radiofrecuencia
WWW	World Wide Web

INDICE DE FIGURAS

Figura 1.1	Search and replace string	8
Figura 1.2	Match Pattern	8
Figura 1.3	Concatenate String	9
Figura 1.4	String To String	9
Figura 1.5	String To IP	9
Figura 1.6	Number To Decimal String	10
Figura 1.7	Decimal String To Number	10
Figura 1.8	Wait until next ms multiple	10
Figura 1.9	Get Date	11
Figura 1.10	Search/Split String	11
Figura 1.11	One button Dialog	11
Figura 1.12	Prompt User for Input	12
Figura 1.13	UDP Open	13
Figura 1.14	UDP Read	13
Figura 1.15	UDP Write	13
Figura 1.16	UDP Close	14
Figura 1.17	IMAQ create	14
Figura 1.18	IMAQ USB Init	14
Figura 1.19	IMAQ USB Grab Setup	15
Figura 1.20	IMAQ USB Grab Acquire	15

Figura 1.21	IMAQ USB Close	15
Figura 1.22	IMAQ Write BMP file	15
Figura 1.23	ADO Connection Create	16
Figura 1.24	ADO Connection Open	16
Figura 1.25	SQL Execute	16
Figura 1.26	ADO Connection Close	17
Figura 1.27	ADO Connection Destroy	17
Figura 1.28	Diagrama interface serial SPI	20
Figura 1.29	Entrada SPI	21
Figura 1.30	SalidaSPI	22
Figura 1.31	Configuracion Cristal	22
Figura 1.32	Configuracion ENC28J60	24
Figura 1.33	Conexión con el ENC28J60	30
Figura 1.34	Conexión con el ULN 2003	39
Figura 2.1	Diagrama de bloques del proyecto	45
Figura 2.2	Botones de entrada	48
Figura 2.3	Conexión Ethernet del PIC	49
Figura 2.4	Esquemático conexión Ethernet	49
Figura 2.5	Ingreso de clave de acceso	64
Figura 2.6	Cuadro de diálogo de clave incorrecta	65
Figura 2.7	Panel frontal del programa en labview	65
Figura 2.8-2.26	Diagrama bloques en labview	69-81

Figura 2.27	Panel frontal del reporte en labview	82
Figura 2.28	Bloque de conexion a la base de datos del reporte	84
Figura 2.29	Control DateTimePicker para selección de fecha	84
Figura 2.30	Lazo que llena los arreglos de la consulta	85
Figura 2.31	Bloque lector de arreglo seleccionado	86
Figura 2.32	Bloque presentador de la imagen	86
Figura 2.33	Conexion motores paso al microcontrolador	88
Figura 2.34	Conexion ULN al microcontrolador	89
Figura 2.35	Circuito emisor	101
Figura 2.36	Circuito receptor	102
Figura 3.1	Circuito PCB	105
Figura 3.2	Configuracion inicial	106
Figura 3.3	Ingreso Residentes	107
Figura 3.4	Registro base de residentes	107
Figura 3.5	Ingreso de foto de residente al sistema	108
Figura 3.6	Ingreso Visitantes	108
Figura 3.7	Registro en base de datos de visitantes	109
Figura 3.8	Ingreso de foto visitante al sistema	109
Figura 3.9	Salida del vehiculo	110
Figura 3.10	Registro base datos vehiculo que sale	110
Figura 3.11	Verificación de datos ingresados al sistema	111
Figura 3.12	Vista frontal de la maqueta	111

Figura 3.13	Vista conexión microcontrolador a ethernet	112
Figura 3.14	Vista conexión sensores al micro	112

INTRODUCCION

I. Alcance del proyecto

Desarrollar un sistema integrado que contenga:

- Validación del acceso de vehículos en un recinto privado mediante tecnología RFID.
- Registro de fotografía de todos los vehículos que accedan al sistema.
- Control del brazo mecánico para permitir acceso vehicular con motor de paso.
- Registrar información de los vehículos visitantes.
- Envío de la información utilizando como medio la Ethernet.
- Utilización de sensores infrarrojos para detección de paso de vehículos.
- Validar el acceso al sistema de control mediante una clave.
- Modelo para presentación de ingreso de vehículos a escala.
- Generar un reporte diario de los ingresos o salidas en cualquier ordenador dentro de la red.

II. Antecedentes

La seguridad siempre ha sido necesaria en lugares privados, controlar el ingreso de vehículos ha sido una de las necesidades que muchos proveedores de equipos de seguridad han desarrollado, en el mercado existen varias soluciones para este tipo de problema, una más sofisticada que la otra, en este curso de hemos aprendido a utilizar herramientas que nos pueden ayudar a desarrollar una solución para la necesidad de controlar el acceso vehicular.

III. Identificación del problema

El problema principal se genera a partir de no tener un control de acceso a un recinto privado, cada vez que ingresa un vehículo, éste debe ser identificado de forma convencional, es decir mediante un guardia de seguridad que vigila la entrada. Esto crea un ambiente de inseguridad por no tener un control en cada momento ya que como humanos, erramos.

Otro problema es la falta de registro, además de no tenerlos con exactitud. Esta falta de información no permite acceder a reportes en un momento determinado.

IV. Descripción breve de la solución

Primero describiremos como funciona, cada vez que un vehículo se acerca a la entrada de un establecimiento privado, se le pondrá a disposición dos opciones, un botón que indique si es usuario visitante, la otra opción es un lector de tarjetas RFID que será utilizada para validar el ingreso a los residentes, la misma q podrá ser puesta cerca del lector de tarjetas RFID para su respectiva validación de ingreso, una vez realizado esto, el programa central enviará una orden para abrir el brazo mecánico que permitirá el acceso al vehículo con su ocupante. En el caso de ser un visitante, este debe presionar el botón de visitante, automáticamente el programa central envía a tomar una foto del ocupante del vehículo y otra a la place del mismo, el programa principal no envía la orden de levantar el brazo mecánico, esto se debe realizar manualmente previa revisión del guardia de seguridad.

Cuando el vehículo está ingresando, este será detectado por sensores que enviaran una señal para que el programa principal, una vez que haya pasado el vehículo, un sensor enviara la señal para proceder a cerrar el brazo, toda la información entre los sensores será transmitida utilizando la red Ethernet hacia el programa en LabView.

Se registrará una fotografía a todo vehículo al salir de recinto, un sensor detectará la salida del mismo y procederá a enviar una señal para que el programa enviara a tomar una fotografía del vehículo.

La solución planteada para el control de acceso vehicular consta de las siguientes partes:

Bloque que gestione la recolección datos a través de sensores hacia el programa principal.

Otro bloque es el encargado de la barrera mecánica, ellos recibirán como entradas una orden que le permita abrir el brazo mecánico.

V. Razón para escoger las herramientas usadas en este proyecto

Labview, por ser un software que permite diseñar y programar en un lenguaje conocido como es el G, es una herramienta que permite trabajar con cámaras para la captura de imágenes, permite además la interacción con dispositivos electrónicos (firmware), para obtener datos.

Microcontroladores, son pequeños chips que poseen características especiales, son programables, poseen características de un

computador tales como procesamiento, memoria y almacenamiento, es por esto que escogemos este tipo de dispositivos, por la facilidad para interactuar con varios dispositivos electrónico.

MikroBasic, es un compilador que genera a partir de un programa escrito en lenguaje BASIC unas instrucciones para que sean programadas en el micro controlador, se escogió esta herramienta por las librerías que posee para la conexión con Ethernet.

VI. Soluciones similares

En nuestro medio es común encontrar controles de acceso vehicular que ya son soluciones diseñadas por fabricantes extranjeros, estas soluciones están disponibles pero no son económicas. Además de no poseer el alcance que se plantea en este proyecto

Entre estas soluciones tenemos:

Control de acceso mediante RFID, cada visitante posee una tarjeta, que es leída al momento de ingresar por un lector.

Otra solución es la que emite un tiquete con un código de barras, este tiquete es suministrado por una maquina al momento de ingresar, y

debe ser presentado a la salida. Este sistema es común en los centros comerciales de la ciudad. Entre los sistemas ofertados por fabricantes extranjeros tenemos por ejemplo a Atmel con su sistema de control de acceso vehicular, que es un panel en el cual se pueden controlar puertas mediante TCP/IP, aquí el enlace del producto http://www.amtel-security.com/Spanish/products/GEC_panel_with_tcpip.html

Las barras vehiculares son comunes en nuestra ciudad, existen varios proveedores que ofertan estas soluciones, aquí un enlace donde se pueden ver barreras.

<http://www.seguridaddigital.com.mx/activacioncart-prodlista.asp?CategorialD=5&SubCategorialD=17> , un proveedor local es el siguiente,

<http://www.controsa.com/productos.php?importante=1&id=9>

CAPITULO 1

Sustento Teórico

1.1 Ambiente de LABVIEW utilizado en el proyecto.

Aquí se explicará las funciones utilizadas del programa de LABVIEW para este proyecto.

1.1.1 Funciones básicas empleadas

Este proyecto tiene tres partes fundamentales: El manejo de imágenes por medio de la cámara web, el envío y recepción de información vía Ethernet, y la escritura de información en una base de datos.

Para enviar información por Ethernet, y para escribir en la base de datos, es necesario enviar la información como un tipo de dato "string", por lo cual en este proyecto usaremos diferentes funciones para el manejo de strings que nos permita enviar y recibir y procesar la información deseada. Las funciones que usamos para el manejo de strings son las siguientes:

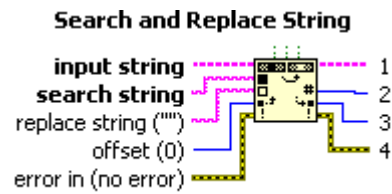


Figura 1.1

Busca un carácter determinado por el usuario dentro de una cadena de caracteres y lo reemplaza por un segundo carácter que el usuario desee.

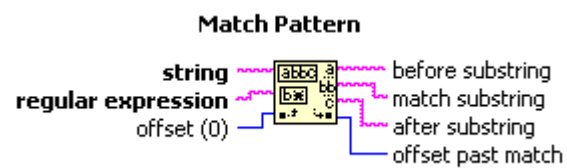


Figura 1.2

Busca en la cadena “string” la subcadena “regular expression” desde la posición “offset”. Lo que nos interesará es que “offset past match” nos dará la posición después de haber una concordancia en las cadenas.

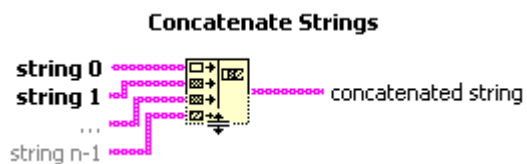


Figura 1.3

Concatena una cantidad determinada de cadena de caracteres en una sola cadena resultante.



Figura 1.4

Convierte una cadena de caracteres en un directorio válido.

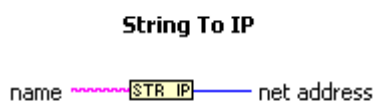


Figura 1.5

Convierte una cadena de caracteres en una dirección IP válida.

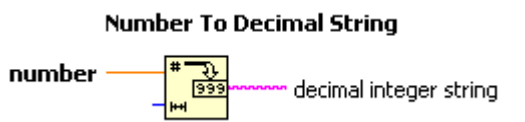


Figura 1.6

Convierte un número decimal en una cadena de caracteres.



Figura 1.7

Convierte caracteres numéricos en un número decimal.

Otras funciones que también usamos en este proyecto, y que son de fácil manejo y comprensión son las siguientes:

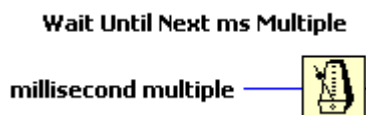


Figura 1.8

Le da un retardo al lazo de la cantidad de milisegundos especificado por el usuario.

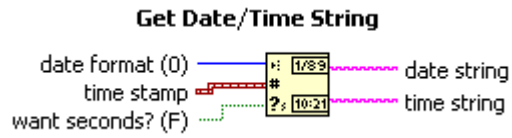


Figura 1.9

Permite obtener la fecha y hora del sistema en cualquier tiempo dado, como dos cadenas separadas.

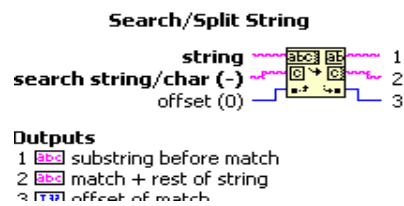


Figura 1.10

Divide la cadena original en dos cadenas separadas por el carácter indicado.

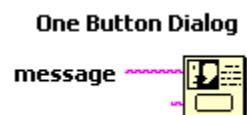


Figura 1.11

Muestra un cuadro de diálogo con un botón y un mensaje.

Aparte de estas funciones, usamos también puertas lógicas e incrementos que no necesitan mayor explicación.

1.1.2 Funciones avanzadas empleadas

Ahora, analizaremos las funciones empleadas en las tres partes principales del proyecto, más la función principal usada en la validación de la clave.

Validación por clave



Figura 1.12

Muestra un cuadro de diálogo con el usuario que se puede configurar para que tenga diferentes tipos de entradas en un mismo cuadro de diálogo.

Comunicación Ethernet

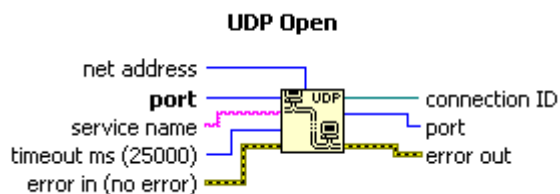


Figura 1.13

Abre un puerto de comunicación Ethernet con comunicación UDP con el número de puerto especificado por el usuario.

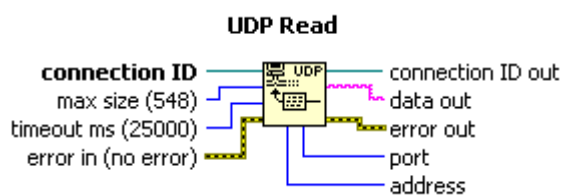


Figura 1.14

Lee los datos recibidos por UDP en el puerto especificado. El tiempo de espera por los datos es ingresado por el usuario.

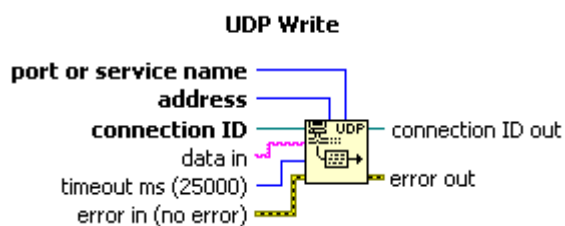


Figura 1.15

Escribe información a un puerto remoto de conexión.

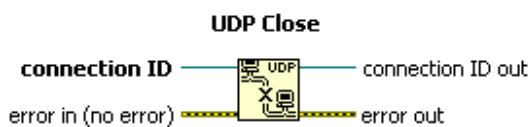


Figura 1.16

Cierra la conexión UDP.

Manejo de cámara USB:

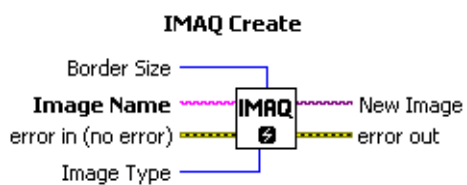


Figura 1.17

Crea un espacio de memoria temporal para una nueva imagen.



Figura 1.18

Abre una sesión para una cámara USB especificada por el usuario.

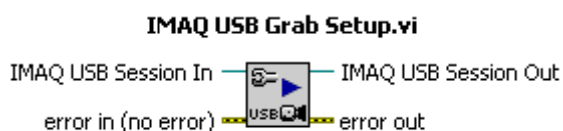


Figura 1.19

Configura y empieza la adquisición continua de imágenes.



Figura 1.20

Obtiene una imagen desde una adquisición continua de imágenes.

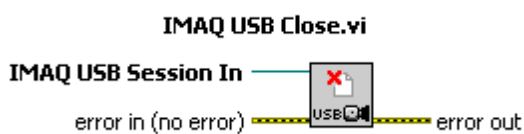


Figura 1.21

Cierra la sesión abierta anteriormente de la cámara USB.

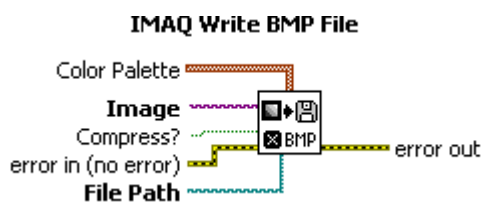


Figura 1.22

Guarda una imagen en el directorio especificado por el usuario.

Escritura de información en la base de datos

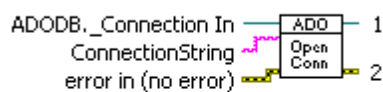
ADO Connection Create.vi



Figura 1.23

Crea un identificador para la conexión que va a ser usada con la base de datos.

ADO Connection Open.vi



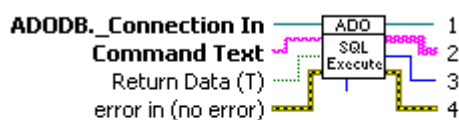
Outputs

- 1 ADODB, _Connection Out
- 2 error out

Figura 1.24

Abre la conexión para un DSN especificado.

SQL Execute.vi



Outputs

- 1 ADODB, _Connection Out
- 2 Data
- 3 Rows Fetched
- 4 error out

Figura 1.25

Permite escribir un QUERY es una base de datos, por lo tanto, permite manipular sus datos, y en nuestro caso, nos permite escribir datos en la base de datos.

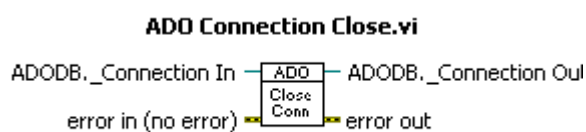


Figura 1.26

Cierra la conexión abierta anteriormente.

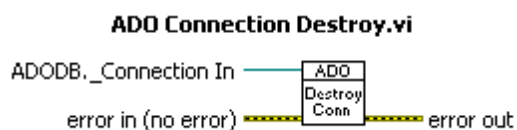


Figura 1.27

Destruye la conexión creada anteriormente.

1.2 El microcontrolador

El microcontrolador usado en este proyecto es el PIC18F4520. Este poderoso microcontrolador tiene muchas características importantes, pero solo nombraremos en este reporte las características usadas en este proyecto.

1.2.1 Características específicas del PIC18F4520 utilizadas en el proyecto

La característica principal del PIC18F4520 que va a ser utilizada será la comunicación SPI para lograr el envío de información por la red Ethernet. Por ello, vamos a describir el modo SPI del PIC 18F4520.

Comunicación SPI

La comunicación síncrona se caracteriza porque los pulsos de sincronización deben ser transmitidos a lo largo de la línea de comunicación entre el transmisor y el receptor.

El bus SPI es un bus diseñado para que sobre éste se coloque un dispositivo maestro y un dispositivo esclavo (comunicación punto a punto). Con relación al bus I2C podemos notar que éste soporta mayor velocidad de comunicación.

El dispositivo SPI posee una línea de selección CS la cual debe pasar al nivel lógico activo (en este caso bajo) para poder realizar la comunicación con el dispositivo. Desde este punto

de vista podríamos colocar sobre un bus de este tipo varios dispositivos, pero utilizando un dispositivo decodificador adicional.

Otra línea podemos observar es la línea HOLD la cual permite al procesador detener momentáneamente la comunicación.

El modo SPI del PIC 18F4520 permite que 8 bits de datos sean transmitidos y recibidos simultáneamente de manera sincrónica. Para lograr esta comunicación, se tienen 3 pines del PIC que serán usados:

- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDI/SDA
- Serial Clock (SCK) – RC3/SCK/SCL

Aquí se muestra el diagrama de bloques de la interface serial sincrónico cuando trabaja en modo SPI:

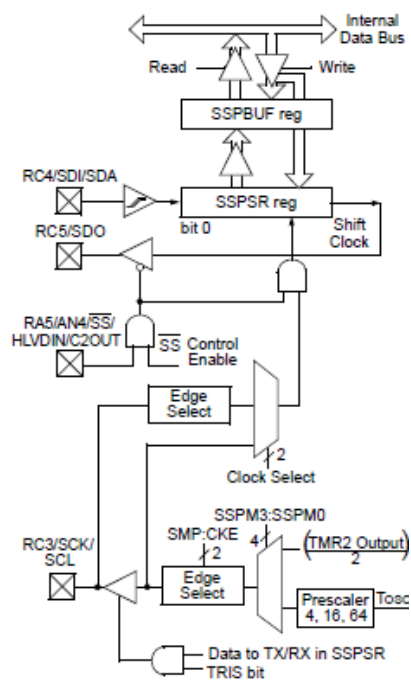


Figura 1.28

El módulo síncrono, tiene cuatro registros asociados al modo SPI:

- MSSP Control Register 1 (SSPCON1)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)

• MSSP Shift Register (SSPSR) – Not directly Accesible

SSPCON1 y SSPSTAT son los registros de control y de estado en el modo de operación SPI.

Cuando se inicializa el SPI, se necesitan especificar muchas opciones, programando los bits de control apropiados (SSPCON1<5:0> y SSPSTAT<7:6>). Con esto, se logra configurar el modo maestro, el modo esclavo, la polaridad del reloj, la fase de muestreo de la entrada de los datos, la lógica del reloj, la velocidad del reloj y el modo de selección del modo esclavo.

A continuación podemos ver cómo se realiza la entrada y la salida de datos en el modo SPI:

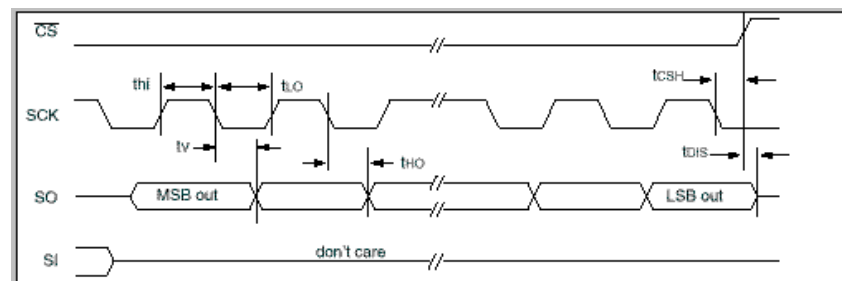


Figura 1.29

Entrada de datos SPI

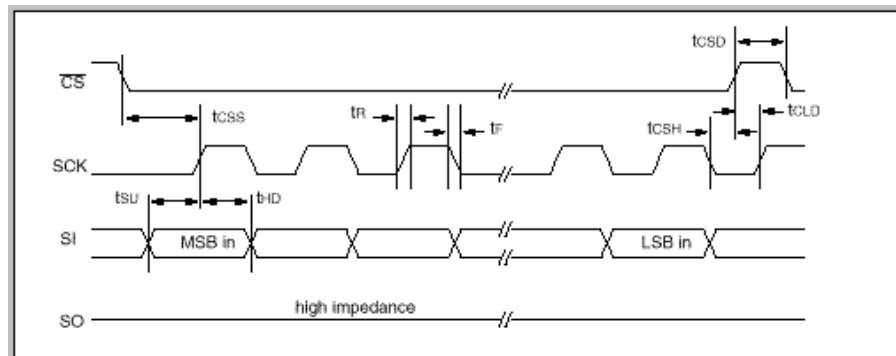


Figura 1.30

Salida de datos SPI

Aparte de la comunicación SPI, es importante acotar que el microcontrolador es polarizado con 5V en Vdd y con 0V en Vss. Aparte, se necesita conectarlo a Vdd en el pin /MCLR. También es importante que en los pines de entrada de los sensores y botoneras, la entrada será de 5V o de 0V, con una resistencia de pullup. Por último, se necesita de un cristal oscilador de 20 MHz para este proyecto con la siguiente configuración:

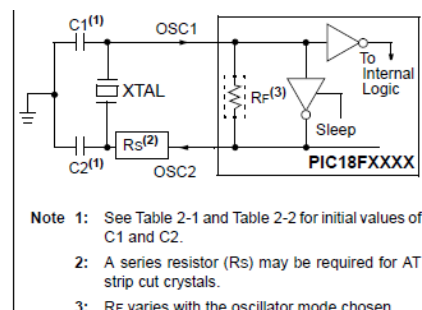


Figura 1.31

1.3 Comunicación Ethernet

1.3.1 El ENC28J60

El ENC28J60 es un circuito integrado fabricado por la empresa Microchip, que funciona como un controlador Ethernet con estándar SPI. Está diseñado para servir como una interfase para una red Ethernet para cualquier microcontrolador que cuente con comunicación SPI. El ENC28J60 contiene todas las especificaciones del estándar IEEE802.3. Incorpora un gran número de filtros de paquetes para limitar los paquetes entrantes. También tiene un DMA interno para una mayor velocidad de flujo de datos, y soporta que por hardware se logren cálculos de checksum de las direcciones IP. La comunicación con el microcontrolador se realiza mediante SPI, con velocidades mayores a 10Mbps. También tiene leds dedicados para ver la actividad de la red y del dispositivo, y para las interrupciones para transmitir y recibir. A continuación vemos una configuración típica del ENC28J60 con un microcontrolador que soporte la comunicación SPI:

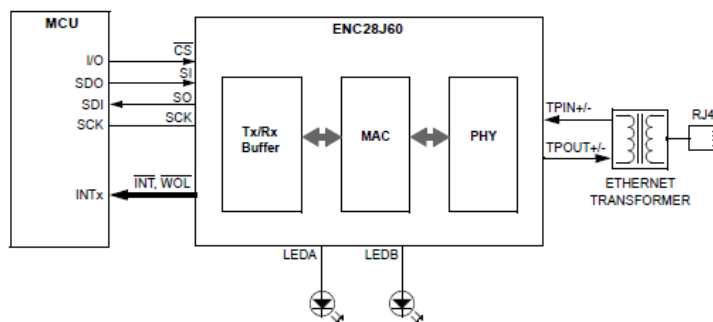


Figura 1.32

1.3.2 Direccionamiento IP y MAC

Dirección IP

Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red del protocolo TCP/IP.

A través de Internet, los ordenadores se conectan entre sí mediante sus respectivas direcciones IP. Sin embargo, a los seres humanos nos es más cómodo utilizar otra notación más fácil de recordar y utilizar, como los nombres de dominio; la traducción entre unos y otros se resuelve mediante los servidores de nombres de dominio DNS.

En su versión 6.55, una dirección IP se implementa con un número de 32 bits que suele ser mostrado en cuatro grupos de números decimales de 8 bits (IPv4). Cada uno de esos números se mueve en un rango de 0 a 255 (expresado en decimal), o de 0 a FF (en hexadecimal) o de 0 a 11111111 (en binario). Las direcciones IP se pueden expresar como números de notación decimal: se dividen los 32 bits de la dirección en cuatro octetos. El valor decimal de cada octeto puede ser entre 0 y 255 (el número binario de 8 bits más alto es 11111111 y esos bits, de derecha a izquierda, tienen valores decimales de 1, 2, 4, 8, 16, 32, 64 y 128, lo que suma 255 en total).

En la expresión de direcciones IPv4 en decimal se separa cada octeto por un carácter ".". Cada uno de estos octetos puede estar comprendido entre 0 y 255, salvo algunas excepciones. Los ceros iniciales, si los hubiera, se pueden obviar.

Una dirección IP fija es una IP asignada por el usuario de manera manual, que es lo que se realizará en nuestro proyecto.

Dirección MAC

En redes de computadoras la dirección MAC (Media Access Control address o dirección de control de acceso al medio) es un identificador de 48 bits (6 bytes) que corresponde de forma única a una tarjeta o interfaz de red. Es individual, cada dispositivo tiene su propia dirección MAC determinada y configurada por el IEEE (los últimos 24 bits) y el fabricante (los primeros 24 bits) utilizando el OUI. La mayoría de los protocolos que trabajan en la capa 2 del modelo OSI usan una de las tres numeraciones manejadas por el IEEE: MAC-48, EUI-48, y EUI-64 las cuales han sido diseñadas para ser identificadores globalmente únicos. No todos los protocolos de comunicación usan direcciones MAC, y no todos los protocolos requieren identificadores globalmente únicos.

Las direcciones MAC son únicas a nivel mundial, puesto que son escritas directamente, en forma binaria, en el hardware en su momento de fabricación. Debido a esto, las direcciones MAC son a veces llamadas "Direcciones Quemadas Dentro" (BIA, por las siglas de Burned-in Address).

La dirección MAC es un número único de 48 bits asignado a cada tarjeta de red. Se conoce también como la dirección física en cuanto identificar dispositivos de red.

La dirección MAC es utilizada en varias tecnologías entre las que se incluyen:

- Ethernet
- 802.5 o redes en anillo a 4 Mbps o 16 Mbps Token Ring
- 802.11 redes inalámbricas (WIFI).
- ATM

En nuestro proyecto también asignaremos la dirección MAC de una manera directa.

1.3.3 Diferencias entre UDP y TCP

En general, las diferencias en cómo entregan los datos UDP y TCP son similares a las diferencias entre una llamada telefónica y una tarjeta postal. TCP funciona como una llamada telefónica, ya que comprueba que el destino está disponible y preparado para la comunicación. UDP funciona como una

tarjeta postal: los mensajes son pequeños y la entrega es probable, pero no siempre está garantizada.

Normalmente, utilizan UDP los programas que transmiten pequeñas cantidades de datos a la vez o que tienen requisitos de tiempo real. En estas situaciones, las capacidades de carga pequeña y multidifusión de UDP (por ejemplo, un datagrama, muchos destinatarios) resultan más apropiadas que TCP.

UDP es notablemente diferente de los servicios y características que proporciona TCP. En la siguiente tabla se comparan las diferencias en el modo de administrar la comunicación TCP/IP según se utilice UDP o TCP para el transporte de datos.

UDP	TCP
Servicio sin conexión; no se establece una sesión entre los hosts.	Servicio orientado a la conexión; se establece una sesión entre los hosts.
UDP no garantiza ni	TCP garantiza la entrega mediante

confirma la entrega, y no secuencia los datos.	el uso de confirmaciones y la entrega secuenciada de datos.
Los programas que utilizan UDP son responsables de proporcionar la confiabilidad necesaria para el transporte de datos.	Los programas que utilizan TCP proporcionan la seguridad del transporte de datos confiable.
UDP es rápido, tiene requisitos de carga pequeños y puede admitir la comunicación punto a punto y de un punto a varios puntos.	TCP es más lento, tiene requisitos de carga mayores y sólo admite la comunicación punto a punto.

UDP y TCP utilizan puertos para identificar las comunicaciones para cada programa TCP/IP.

En nuestro proyecto, utilizaremos la comunicación UDP por su gran velocidad, y porque no vamos a mandar una gran cantidad de datos vía Ethernet.

1.3.4 Interfaz con PIC18F4520

Para conectar el PIC18F4520 con la red ethernet, usaremos el CI ENC28J60, junto con otros componentes para ayudar a la comunicación. Todos los componentes necesarios para conectar el ENC28J60 con el PIC y con la red Ethernet, se interconectan en una placa llamada la M2ETH board. A continuación podemos ver el esquemático de la conexión de la placa del ENC28J60:

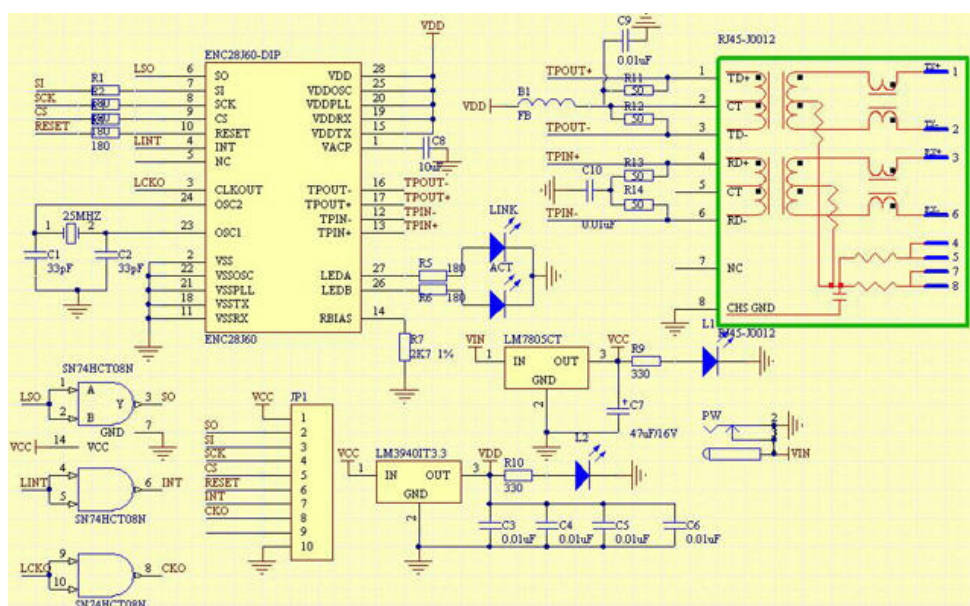


Figura 1.33

Las entradas SI, SO, SCK, RESET y CS vienen desde el PIC.

Como podemos ver, la interfaz permite conectar la placa ethernet con el pic por medio de la comunicación SPI que es lo que permite la comunicación del PIC con el integrado, y consecuentemente con la red Ethernet.

1.4 El ambiente MikroBasic

1.4.1 Librerías utilizadas

La principal librería utilizada en este proyecto en Mikrobasic es la librería SPI Ethernet Library. Esta librería está diseñada para simplificar el manejo de la comunicación con la placa del ENC28J60. Esta librería funciona con cualquier PIC que soporte comunicación SPI, y con más de 4KB de memoria ROM. Se necesita de por lo menos 38 o 40 MHz de reloj para lograr que el PIC trabaje correctamente con esta librería.

La librería SPI Ethernet soporta lo siguiente:

- IPv4 protocol.
- ARP requests.
- ICMP echo requests.
- UDP requests.
- TCP requests (no stack, no packet reconstruction).

- ARP client with cache.
- DNS client.
- UDP client.
- DHCP client.

Como la comunicación necesaria en este proyecto es SPI, se usará una función de la SPI library que es la SPI_init que inicializa el modo SPI del microcontrolador.

Funciones de la librería SPI Ethernet Library utilizadas

1.- Spi_Ethernet_Init: Esta función inicializa el módulo que inicializa el controlador ENC28J60.

Parámetros:

- resetPort: Puerto donde está el pin de reset.
- resetBit: Pin de reset.
- CSport: Puerto donde está el pin CS.
- CSbit: Pin CS.
- mac: Nombre de la variable donde está la dirección MAC.
- ip: Nombre de la variable donde está la dirección IP.
- fullDuplex: Selector del modo de funcionamiento Ethernet.

2.- Spi_Ethernet_sendUDP: Envía información vía UDP especificando los parámetros mencionados a continuación:

Parámetros:

- destIP: Dirección IP destino.
- sourcePort: Puerto abierto por el PIC para la comunicación UDP.
- destPort: Puerto donde se el dispositivo remoto receptorá la información.
- pkt: Paquete a transmitir.
- pktLen: Tamaño del paquete.

3.- Spi_Ethernet_getByte: Recepta un byte desde el buffer Ethernet.

4.- Spi_Ethernet_doPacket: Procesa el siguiente paquete receptado de la siguiente manera:

- Paquetes ARP e ICMP son reenviados automaticamente.
- Para los paquetes TCP, se llama a la función [Spi Ethernet UserTCP](#) para continuar con el procesamiento.

- Para los paquetes UDP, se llama a la función `Spi_Ethernet_UserUDP` para continuar con el procesamiento.

5.- `Spi_Ethernet_UserUDP`: Esta función es llamada y ejecutada en el momento que `Spi_Ethernet_doPacket` es ejecutado.

1.4.2 Programa de desarrollo utilizado MikroBasic

Para el programa que se va a crear para este proyecto, se va hacer uso de todas las funciones de la librería SPI Ethernet ya explicadas, y también se hará uso de directivas necesarias para sensar los puertos y para el manejo del motor.

Una de las funciones utilizadas es la función `Button`. Esta función proporciona un antirebote en la entrada del PIC. El parámetro "port" especifica el puerto de entrada a ser sensado. El parámetro "pin", es el número de pin a ser sensado. "time" es el periodo de antirebote en milisegundos. "active_state" determina si el botón es de lógica negativa o positiva.

También fueron usadas directivas propias de la programación en BASIC, como el lazo while que hace una iteración mientras se cumpla la condición dentro del paréntesis. También fue usado el lazo for, que realiza la iteración el número de veces indicada por la variable i. La instrucción condicional if – else, realiza una acción si se cumple la condición, y otra condición diferente si no se cumple la condición.

1.5 Manejo del motor activación de barrera mecánica

1.5.1 Investigación sobre Tipos de motores disponibles en el mercado

Motores de corriente continua

El motor de corriente continua es una máquina que convierte la energía eléctrica en mecánica, principalmente mediante el movimiento rotatorio. La principal característica del motor de corriente continua es la posibilidad de regular la velocidad desde vacío a plena carga.

Motor paso a paso

El motor PaP es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa es que es capaz de avanzar una serie de grados (paso) dependiendo de sus entradas de control. El motor paso a paso se comporta de la misma manera que un convertidor digital-analógico y puede ser gobernado por impulsos procedentes de sistemas lógicos.

Servomotor

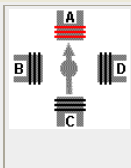
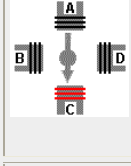
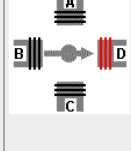
Un servomotor de modelismo —conocido generalmente como servo o servo de modelismo— es un dispositivo actuador que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y de mantenerse estable en dicha posición. Está formado por un motor de corriente continua, una caja reductora y un circuito de control, y su margen de funcionamiento generalmente es de menos de una vuelta completa.

1.5.2 Recomendación del motor de paso para ser usado en este proyecto (para evaluar costos)

Para este proyecto, usaremos el motor de paso, ya que podemos lograr que se mueva la cantidad de grados que nosotros deseamos con solo impartirle un nivel alto o bajo de voltaje. Con relación al costo, el motor de paso se adapta a la perfección a nuestras limitaciones en el proyecto, ya que para el tamaño del proyecto, que es un modelo a escala, un motor de paso pequeño, es perfecto. Estos motores de paso, no son necesarios comprarlos, sino que se los puede sacar de una impresora que ya no este en uso para no incurrir en gastos adicionales.

El funcionamiento de un motor de paso es muy simple. Simplemente hay que impartirle un nivel de voltaje alto a la bobina que se necesite, y el motor se moverá hacia esa bobina. Dependiendo de la cantidad de grados por paso que tenga el motor, serán necesarios más pasos o menos pasos para lograr una vuelta entera del motor. El funcionamiento se ilustra a continuación:

PASO	Bobina A	Bobina B	Bobina C	Bobina D	
-------------	-----------------	-----------------	-----------------	-----------------	--

1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	
3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

1.5.3 Comunicación del motor con el microcontrolador (El ULN2003)

El motor de paso no puede conectarse directamente con el microcontrolador porque el microcontrolador no puede suministrarle la suficiente corriente al motor para que gire correctamente. Para ello, incluimos en el circuito un integrado (ULN2003) que nos permitirá aumentar la corriente por medio de una serie de transistores en configuración Darlington, resultando en una corriente aumentada, suficiente para que el motor de paso funcione a la perfección. La conexión sería de la siguiente manera:

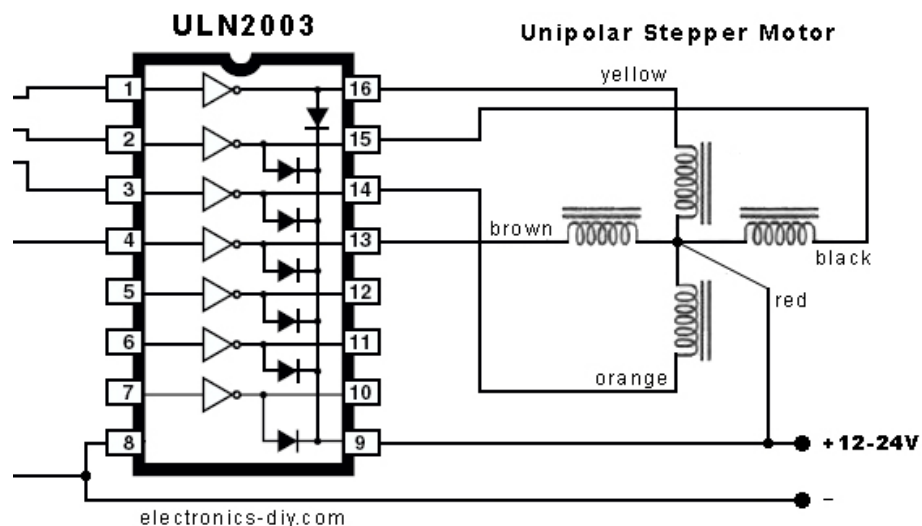


Figura 1.34

Donde las cuatro entradas que no están conectadas en el integrado, van a las cuatro salidas de control del microcontrolador. De esta manera, con solo poner un voltaje alto en cualquiera de las bobinas del motor, el motor se moverá correctamente.

1.6 Tecnología de Sensores Infrarrojos

En el proyecto de tesis, utilizamos la tecnología infrarroja para detectar el pasar de un vehículo en la entrada o salida, con el fin de bajar el brazo mecánico en el momento adecuado, se utiliza el modo

de comunicación punto a punto, a continuación detallamos en forma breve en qué consiste esta tecnología.

La radiación infrarroja, radiación térmica o radiación IR es un tipo de radiación electromagnética de mayor longitud de onda que la luz visible, pero menor que la de las microondas. Consecuentemente, tiene menor frecuencia que la luz visible y mayor que las microondas. Su rango de longitudes de onda va desde unos 700 nanómetros hasta 1 milímetro. La radiación infrarroja es emitida por cualquier cuerpo cuya temperatura sea mayor que 0 Kelvin, es decir, $-273,15$ grados Celsius (cero absoluto).

1.6.1 La emisión y recepción de energía infrarroja

Diodo emisor de luz, también conocido como LED (acrónimo del inglés de Light-Emitting Diode) es un dispositivo semiconductor (diodo) que emite luz incoherente de espectro reducido cuando se polariza de forma directa la unión PN del mismo y circula por él una corriente eléctrica. Este fenómeno es una forma de electroluminiscencia. El color (longitud de onda), depende del material semiconductor empleado en la construcción del diodo y puede variar desde el ultravioleta, pasando por el visible, hasta el infrarrojo. Los diodos emisores

de luz que emiten luz ultravioleta también reciben el nombre de UV LED (UltraViolet Light-Emitting Diode) y los que emiten luz infrarroja suelen recibir la denominación de IRED (Infra-Red Emitting Diode). En nuestro caso, usaremos un diodo emisor de radiación infrarroja.

Como receptor utilizaremos un fototransistor, que es un transistor sensible a la luz, normalmente a los infrarrojos. La luz incide sobre la región de base, generando portadores en ella. Esta carga de base lleva el transistor al estado de conducción. El fototransistor es más sensible que el fotodiodo por el efecto de ganancia propio del transistor.

1.7 Tecnología RFID

Usaremos la tecnología RFID en este proyecto para poder identificar al residente que está por llegar. A continuación hablaremos sobre esta tecnología que está sustituyendo poco a poco a las etiquetas de códigos de barras y a las tarjetas magnéticas en todas sus aplicaciones.

1.7.1 Funcionamiento RFID

Todo sistema RFID se compone de un interrogador o sistema de base que lee y escribe datos en los dispositivos y un "transponder" o transmisor que responde al interrogador.

1. El interrogador genera un campo de radiofrecuencia, normalmente conmutando una bobina a alta frecuencia. Las frecuencias usuales van desde 125 Khz hasta la banda ISM de 2.4 Ghz, incluso más.
2. El campo de radiofrecuencia genera una corriente eléctrica sobre la bobina de recepción del dispositivo. Esta señal es rectificadora y de esta manera se alimenta el circuito.
3. Cuando la alimentación llega a ser suficiente el circuito transmite sus datos.
4. El interrogador detecta los datos transmitidos por la tarjeta como una perturbación del propio nivel de la señal.

La señal recibida por el interrogador desde la tarjeta está a un nivel de -60 db por debajo de la portadora de transmisión. El rango de lectura para la mayoría de los casos está entre los 30 y 60 centímetros de distancia entre interrogador y tarjeta.

1.7.2 TAGS Pasivos

Las tags RFID pueden ser activos, semipasivos (también conocidos como semiactivos o asistidos por batería) o pasivos.

Los tags pasivos no poseen alimentación eléctrica. La señal que les llega de los lectores induce una corriente eléctrica pequeña y suficiente para operar el circuito integrado CMOS del tag, de forma que puede generar y transmitir una respuesta. La mayoría de tags pasivos utiliza backscatter sobre la portadora recibida; esto es, la antena ha de estar diseñada para obtener la energía necesaria para funcionar a la vez que para transmitir la respuesta por backscatter. Esta respuesta puede ser cualquier tipo de información, no sólo un código identificador. Un tag puede incluir memoria no volátil, posiblemente escribible (por ejemplo EEPROM).

Los tags pasivos suelen tener distancias de uso práctico comprendidas entre los 10 cm (ISO 14443) y llegando hasta unos pocos metros (EPC e ISO 18000-6), según la frecuencia de funcionamiento y el diseño y tamaño de la antena. Por su sencillez conceptual, son obtenibles por medio de un proceso

de impresión de las antenas. Como no precisan de alimentación energética, el dispositivo puede resultar muy pequeño: pueden incluirse en una pegatina o insertarse bajo la piel.

CAPITULO 2

Construcción del proyecto

2.1 Diagrama de bloques

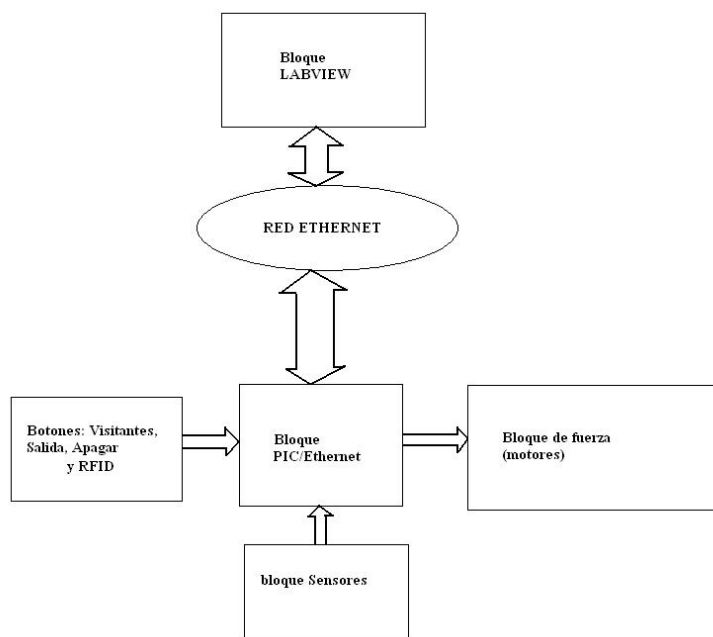


Figura 2.1

2.1.1 Explicación breve del funcionamiento del circuito general

Nuestro proyecto es un sistema de control de ingreso de vehículos, donde se tendrá tres columnas para el acceso de automóviles. La primera columna es la columna de residentes, donde el residente podrá ingresar si pasa su tarjeta de identificación de residente por el lector RFID, lo que causará que, después de ser verificado el número, se le tome una

fotografía al residente, y se le abra la plumilla de acceso. La segunda columna es la columna de visitantes. En el momento que un visitante llega a la columna correspondiente, éste presionará un botón indicando que es un visitante; luego de ello, se procederá a tomar la foto tanto del visitante como de la placa de su automóvil; por último, se le informará al encargado de la garita y del control del programa, que hay un visitante en espera, y se le facilitará a la persona encargada, abrir o cerrar la plumilla de acceso a su criterio. Finalmente, tenemos la columna de Salida. Cuando una persona desea salir, presionará un botón indicando que desea salir, y se le procederá a tomar una foto, para luego abrirle la plumilla de salida. La plumilla de salida será abierta siempre a menos que la persona que controle el sistema bloquee la puerta desde el computador controlador. Se utilizarán sensores que permitirán determinar si el automóvil ya pasó la plumilla de acceso, para luego proceder a cerrarla. Vale la pena recalcar que las fotos, la fecha y la hora serán guardadas en una base de datos, y la información enviada y recibida hacia el ordenador controlador, será vía Ethernet. Adicionalmente, para apagar el sistema habrá un botón que hará esta función.

2.2 Descripción detallada de cada bloque

A continuación detallaremos las funciones de cada bloque y como está constituido interiormente. Comprenderemos en esta sección el funcionamiento de cada elemento de cada bloque, y la función de los mismos en el proyecto.

2.2.1 Bloque PIC/ETHERNET

Este bloque se encarga de receptar la información externa, tanto de las botoneras, de los sensores y del RFID. También, este bloque se encarga del manejo de los motores de paso, pero el funcionamiento del manejo de los motores lo explicaremos con más detalle en una sección posterior.

2.2.1.1 Captura de información externa mediante botones y RFID

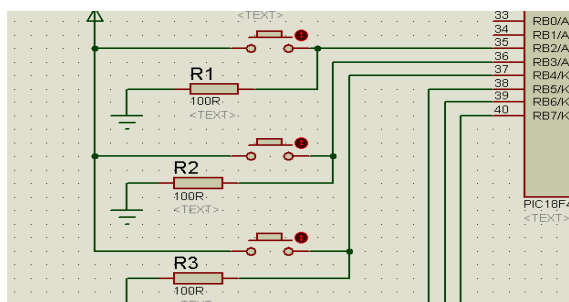


Figura 2.2

El módulo RFID tiene 3 terminales: El Terminal que va a la referencia GND del sistema; el Terminal que va a la referencia de voltaje positivo del circuito; el Terminal /enable que debe estar polarizado con un voltaje bajo para que el circuito funcione correctamente; y por último, el Terminal SOUT que enviará los datos de la tarjeta de identificación serialmente hacia el PIC18F4520 en el pin 26 (RC7).

2.2.1.2 Esquemático de la conexión ETHERNET

El esquemático para la conexión Ethernet es el siguiente:

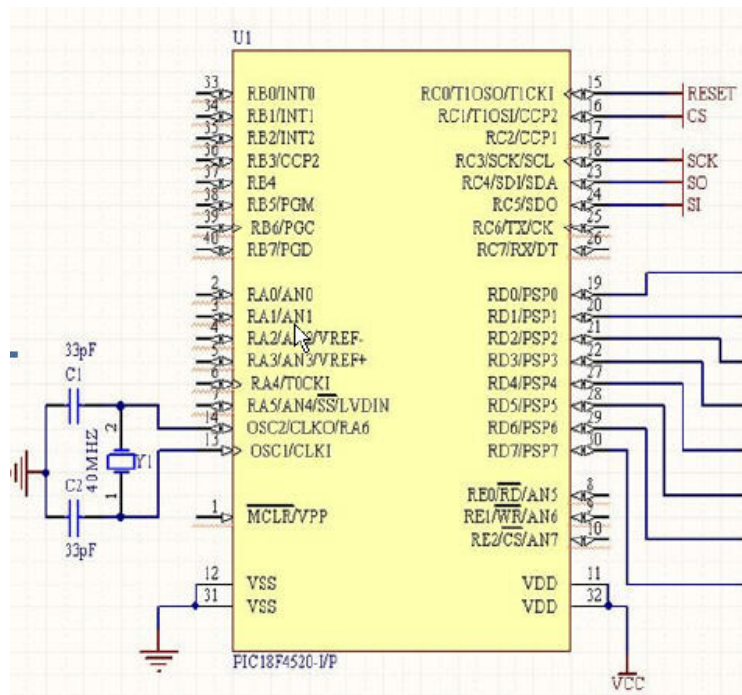


Figura 2.3

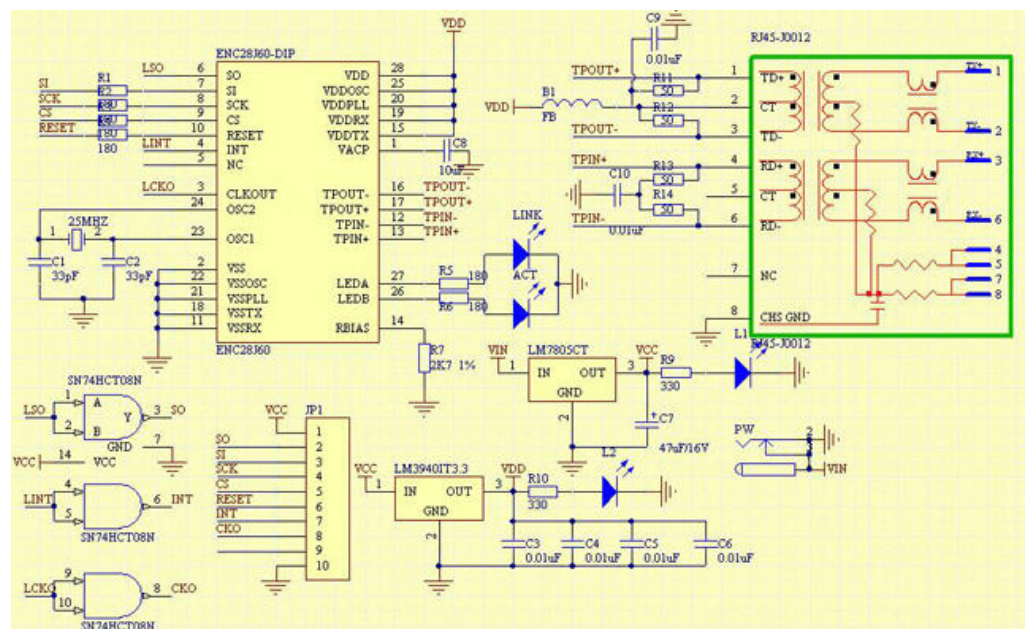


Figura 2.4

2.2.1.3 Explicación del funcionamiento del circuito

El corazón de este circuito es el ENC28J60, que es un controlador para la comunicación Ethernet. Este circuito integrado permite la comunicación entre cualquier microcontrolador que soporte comunicación SPI, y con más de 4KB de memoria ROM, con la red Ethernet. De esta manera, logramos la recepción, y el envío de paquetes en el microcontrolador vía Ethernet. Para ello, necesitamos conectar el microcontrolador con el módulo del ENC28J60 mediante los pines encargados de la comunicación SPI del microcontrolador, como está mostrado en la figura anterior. De esta manera, por medio de la comunicación SPI, el PIC18F4520 se conecta a la red Ethernet.

2.2.1.4 Porción del Programa del PIC18F4520 en MIKROBASIC para la comunicación Ethernet

```
program proyecto  
include "eth_enc28j60"
```



```

include "eth_enc28j60_api"

dim mymacaddr as byte [6]
dim myipaddr as byte [4]
dim getRequest as byte ' variables de uso general
dim dyna as byte[30]
dim txt as string[100]
dim i as byte ' No evaluamos paquetes TCP por lo que
la funcion es 0
dim IpAddr as byte[4] ' remote IP address
dim receive as byte [12] ' dato serial recibido del RFID
dim cont as longword

sub function Spi_Ethernet_UserTCP(dim byref remoteHost as
byte[4], dim remotePort, localPort, reqLength as word) as word
    result=0
end sub ' Implementamos la recepcion de paquetes UDP
reseteando 'siempre el resultado a 0

sub function Spi_Ethernet_UserUDP(dim byref remoteHost as
byte[4], dim remotePort, destPort, reqLength as word) as word
    result = 0 ' reseteo de la funcion

```

if destport = 200 then ' El puerto donde vamos a escuchar los
paquetes es 200 'cualquier petición a otro no es evaluada.

```
    ' captamos los bytes entrantes del buffer ethernet
    getRequest=spi_ethernet_getbyte()
    if getRequest="1" then
        'rutina para controlar el motor puerta residentes
    else
        if getRequest="0" then
            'rutina para controlar el motor puerta Salida
            if getRequest="2" then
                'rutina para controlar el motor puerta Salida
            end if
        end if
    end if
end if
end if
end sub
main:
adcon0=0
adcon1=15
trisa=0
porta=$08
trisb=$FF
```

```
portb=0
```

```
trisd=0
```

```
portd=$88
```

```
trise=0
```

```
porte=0
```

```
' Direccion Fisica de nuestroPIC la MAC
```

```
mymacaddr[0]=0x00
```

```
mymacaddr[1]=0x14
```

```
mymacaddr[2]=0xA5
```

```
mymacaddr[3]=0x76
```

```
mymacaddr[4]=0x19
```

```
mymacaddr[5]=0x3F
```

```
' Direccion IP del PIC
```

```
myipaddr[0]=169
```

```
myipaddr[1]=254
```

```
myipaddr[2]=203
```

```
myipaddr[3]=210
```

```
  USART_Init(2400)
```

```
  spi_init()
```

```
  spi_ethernet_init(portc,0,portc,1,mymacaddr,myipaddr,1)
```

```
  IpAddr[0] = 169
```

```
  IpAddr[1] = 254
```

```
IpAddr[2] = 203
IpAddr[3] = 96
PORTE.0=0
while true
    txt="H" 'inicializo con cualquier valor
    cont=0
    if Usart_Data_Ready = 1 then 'residente 'espera de datos
serials (RFID)
        i=12
        while i>0 'Obtención de los datos del
RFID
            PORTE.0=1
            if Usart_Data_Ready = 1 then
                receive[i-1] = Usart_Read
                i = i - 1
            end if
            spi_ethernet_dopacket()
            cont=cont+1
            if cont=5000 then
                i=0
            end if
        wend
```

```

' Primera Validación de la tarjeta RFID

IF cont<>5000 then

if receive[11]=$0A then

    if receive[10]=$30 then

        if receive[9]=$46 then

            if receive[8]=$30 then

                if receive[7]=$33 then

                    if receive[6]=$30 then

                        if receive[5]=$34 then

                            if receive[4]=$31 then

                                if receive[3]=$34 then

                                    if receive[2]=$31 then

                                        if receive[1]=$38 then

                                            if receive[0]=$0D then

                                                txt = "1 " ' residente válido

                                                spi_ethernet_sendUDP(lpAddr, 10001, 4000, @txt,
Strlen(txt))

                                                while Usart_Data_Ready = 1

                                                    spi_ethernet_dopacket()

                                                wend

                                                delay_ms(2500)

                                            end if

```



```
        end if
    end if
    end if
end if
PORTE.0=0
    if(Button(PORTB, 2, 1, 1)) then 'botón visitantes
        while(Button(PORTB, 2, 1, 1))
            PORTE.0=1
        wend
        txt = "0"
    spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt, Strlen(txt))
        PORTE.0=0
    end if
        if(Button(PORTB, 3, 1, 1)) then 'botón apagado
            while(Button(PORTB, 3, 1, 1)) wend
            txt = "3"
            spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt,
Strlen(txt))
            PORTE.0=0
        end if
            if(Button(PORTB, 4, 1, 1)) then 'botón salida
```



```
while(Button(PORTB, 4, 1, 1)) wend
txt = "4"
spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt,
Strlen(txt))
PORTE.0=0
end if

spi_ethernet_dopacket()

wend

end.
```

2.2.1.5 Explicación del programa del PIC18F4520

En el programa principal, primero que todo es necesario inicializar los módulos que usaremos en este proyecto. La función `USART_Init(2400)` inicializa la comunicación serial a 2400 baudios. La función `spi_init()` inicializa y habilita la comunicación SPI del PIC. Por último, la función `spi_ethernet_init(portc,0,portc,1,mymacaddr,myipaddr,1)` permite inicializar la comunicación del PIC con el módulo ENC28J60, asignando al PIC la dirección IP guardada en la variable `myipaddr` y la dirección MAC en la variable

mymacaddr. También se crean otras variables para manejo del programa como es la variable i, cont, entre otras.

Al observar el programa principal, nos damos cuenta que existen tres condiciones principales para evaluar respecto a datos externos recibidos.

La primera evalúa si se ha pasado la tarjeta RFID por el sensor de residentes, preguntando si hay datos seriales listos para ser leídos. Cuando la condición es verdadera, y hay datos seriales listos, se obtienen los 12 bytes de identificación de la tarjeta RFID mediante un lazo while y el índice i. Para ello, se inicializa la variable i con el valor de 12, y se va guardando cada número en un arreglo dependiendo del índice i, que en cada iteración disminuye en 1 unidad. Cuando i llega al valor de 0, se sale del lazo while, y se tiene en el arreglo receive, los 12 números de la tarjeta RFID. El siguiente paso en el programa es verificar si esos 12 números son los mismos de la tarjeta de residente del sistema de control de ingreso de vehículos. Se compara dos veces

el número, ya que experimentalmente nos dimos cuenta que luego de la primera vez que el usuario pasa la tarjeta RFID, el número guardado en receive, cambia, siendo el mismo, pero corrido dos espacios. Así igual, nos dimos cuenta que luego de la primera vez que el usuario pasa la tarjeta, el módulo RFID envía más dígitos seriales que no son importantes para nuestro programa; por lo tanto, implementamos un contador en la variable cont, que nos permitirá salir del lazo while de obtención de datos si es que el programa se queda anclado por más de 5000 iteraciones del lazo while por esos datos enviados por el módulo RFID. Finalmente, si el número ingresado es correcto, se envía vía Ethernet el número 1 que servirá como bandera para comunicar a Labview que se debe abrir la puerta de residentes y que se debe tomar la foto al residente. Se espera luego que ya no haya más datos seriales a recibir, y se le asigna un retardo de dos segundos y medio para que cualquier dato guardado en el buffer de comunicación serial, sea eliminado. Esto se realiza porque la tarjeta RFID envía la información muchas veces después de pasar por el lector RFID.

Para el ingreso de los visitantes, se espera a que se presione el botón conectado en el puerto B, pin 2. Luego de ser presionado, se ingresa a un lazo while donde se mantiene hasta que el botón sea soltado. Una vez validado que el botón ha sido presionado y soltado, se envía el dato 0 vía Ethernet, indicando que un visitante ha llegado. Las otras dos botoneras funcionan de la misma manera, con la diferencia que los botones están conectados a pines diferentes del PIC, y la información enviada para indicar que se debe apagar el sistema, y para indicar que alguien desea salir, son diferentes; se utilizará el dato 3 para indicar que el sistema debe ser apagado, y el dato 4 para indicar que hay un usuario queriendo salir. Luego, esa información será procesada por Labview, lo que explicaremos más adelante detalladamente.

La recepción de paquetes UDP se logra gracias a la función `spi_ethernet_dopacket()`. Cuando hay un dato en el buffer de recepción UDP, se ejecuta la función `Spi_Ethernet_UserUDP` que será la encargada de

procesar los paquetes UDP recibidos. Como sabemos que los datos recibidos vía ethernet serán datos de control para los motores de las plumillas de acceso, cada dato recibido procesará la apertura y el cierre de las plumillas de acceso. Si es enviado un 1, entonces la plumilla de los residentes se abrirá. Si un 0 es enviado, se debe abrir la plumilla de salida. Por último, si es enviado un dato 2, es porque la plumilla de los visitantes debe ser abierta. Después de ser abierta cada plumilla, se espera a la señal del sensor. Como se puede apreciar, se espera primero que el sensor pase de un nivel bajo de voltaje a un nivel alto de voltaje, y luego, que el voltaje del sensor pase de un nivel alto a bajo. Luego que el sensor ha pasado a bajo, se entiende que el automóvil ya ha pasado la plumilla, por lo que se la debe cerrar, y la rutina para cerrar la plumilla de acceso se ejecuta.

2.2.2 Bloque LabView: Procesamiento de la información en LABVIEW 8.5

En esta sección explicaremos con detalle el funcionamiento del programa en LabView y cada parte del mismo.

2.2.2.1 Ingreso de clave de acceso

Para el ingreso de la clave, se mostrará un cuadro de diálogo que pedirá la clave de acceso:

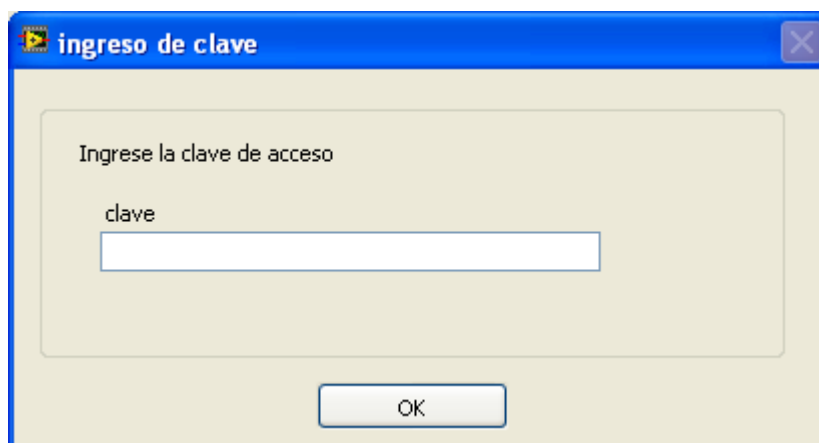


Figura 2.5

Si la clave es la correcta, el programa empezará con el panel frontal mostrado en la siguiente subsección. Si la clave es incorrecta, se mostrará un cuadro de diálogo indicando que la clave no es correcta, y el programa no se ejecutará.

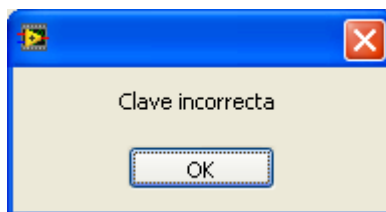


Figura 2.6

2.2.2.2 Panel frontal del programa en LABVIEW

A continuación se muestra el panel frontal del programa en LabView. Éste será el panel de control para la persona que maneja el sistema de seguridad después de haber ingresado la clave de acceso.

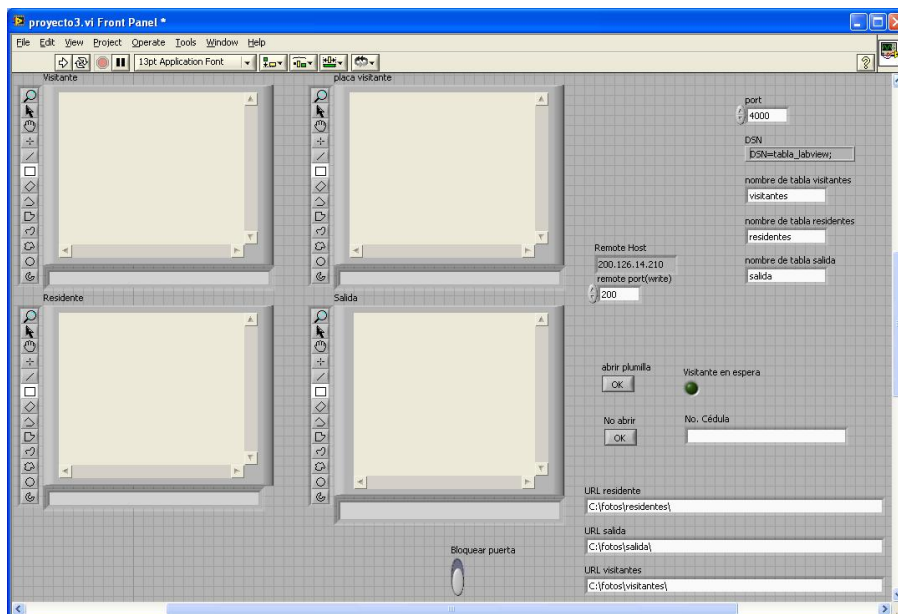


Figura 2.7

2.2.2.3 Explicación de cada elemento del panel frontal

Port: Es el número de puerto UDP que abre el computador para recibir los paquetes enviados por el PIC.

DSN: Es la línea de código que indica el DSN de la base de datos en la que vamos a escribir. El formato para ingresar el DSN es el siguiente: DSN= *“nombre del DSN”*.

Nombre de tabla visitantes: Es el nombre de la tabla dentro de la base de datos donde se guardarán los datos (fecha, hora e id) de los visitantes que llegarán.

Nombre de tabla residentes: Es el nombre de la tabla dentro e la base de datos donde se guardarán los datos (fecha, hora e id) de los residentes que llegarán.

Nombre de tabla Salida: Es el nombre de la tabla dentro de la base de datos donde se guardarán los datos (fecha, hora e id) de los usuarios que saldrán.

Remote Host: Es la dirección IP del PIC hacia donde el ordenador se comunicará.

Remote Port (write): Es el puerto que abre el PIC para escuchar los paquetes UDP que se enviarán desde el computador.

Pantallas: Mostrarán la foto tomada del residente, visitante, de la placa del visitante y del usuario que va a salir.

Botón bloquear puerta: Si está en su posición hacia arriba, la puerta de salida estará bloqueada y el usuario no podrá salir después de presionar el botón de salida.

LED Visitante en espera: Indicador que muestra que ya se le tomó la foto al visitante y a su placa, y que está a la espera que le abran la plumilla o no.

Botón abrir plumilla: Abre la plumilla de acceso a los visitantes si es presionado.

Botón No abrir: No abre la plumilla de acceso a los visitantes si es presionado.

URL residente: Es el directorio, dentro de la computadora de control, donde van a ser guardadas las fotos de los residentes.

URL salida: Es el directorio, dentro de la computadora de control, donde van a ser guardadas las fotos de los usuarios que van a salir.

URL visitantes: Es el directorio, dentro de la computadora de control, donde van a ser guardadas las fotos de los visitantes y las fotos de sus placas.

No. Cédula: Campo donde se ingresará el número de cédula del visitante que ha llegado.

2.2.2.4 Diagrama de bloques del programa en LABVIEW

A continuación mostraremos y explicaremos el diagrama de bloques del programa hecho en LabView. Se lo

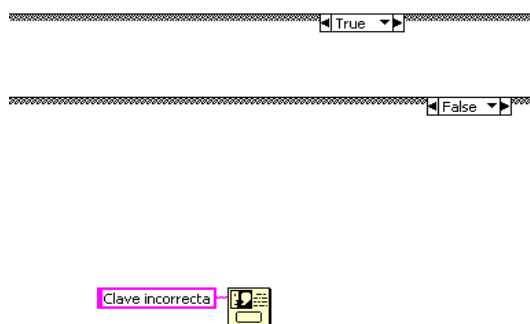


Figura 2.9

Si la clave no es correcta, se muestra un cuadro de diálogo indicando que la clave es incorrecta, y no se ejecuta el programa.

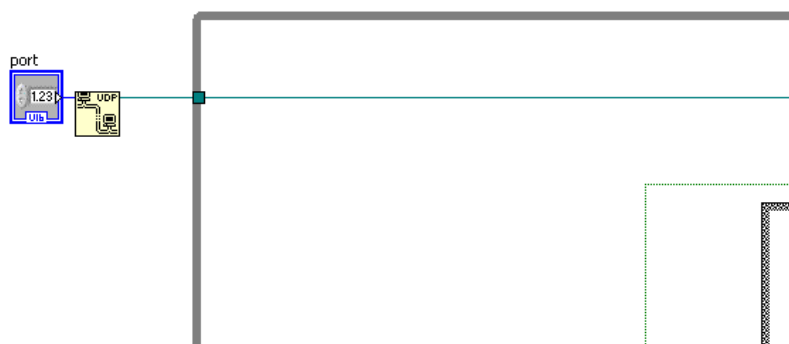


Figura 2.10

Se inicializa la comunicación UDP abriendo el puerto indicado por PORT en el panel frontal.

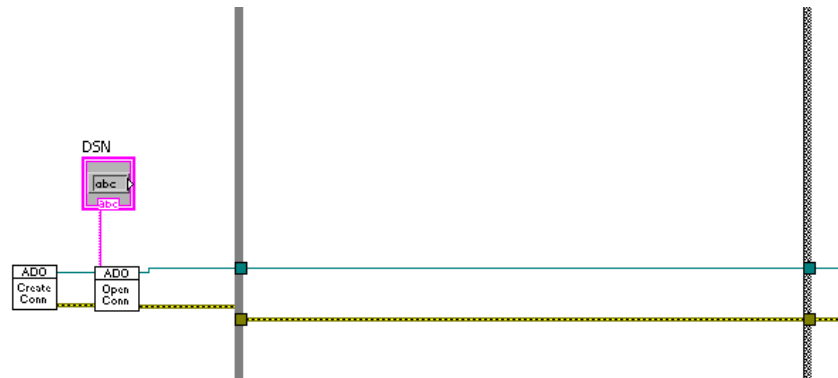


Figura 2.11

Se crea y se abre la conexión con la base de datos indicada por el DSN que se especifica en el panel frontal.

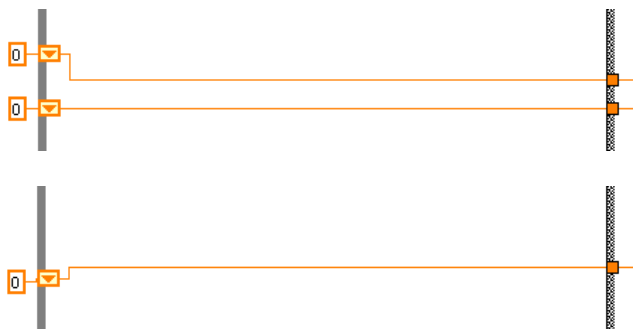


Figura 2.12

Se crean e inicializan 3 shift registers para el conteo de los id.

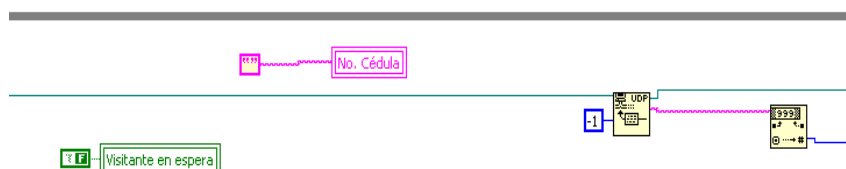


Figura 2.13

Se inicializa con falso el LED de visitante en espera; se inicializa el control de la cédula del visitante con una cadena vacía de caracteres, y se espera indefinidamente por información UDP para luego convertirla de string a número.

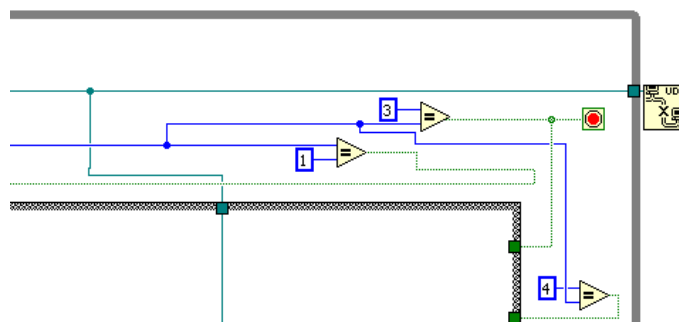


Figura 2.14

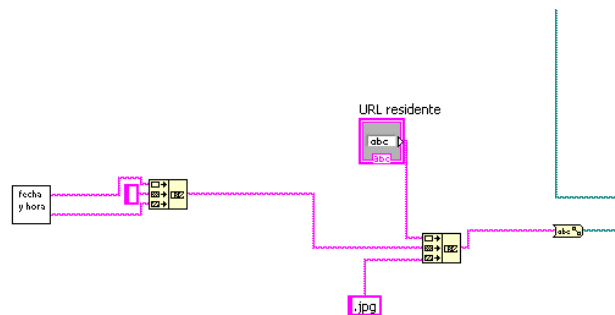


Figura 2.16

El directorio de las fotos guardadas se especifica el control de cadena de caracteres en el panel frontal, y el nombre es la fecha y la hora, pero con el formato cambiado (en vez de / ponemos - y en vez de: ponemos -).

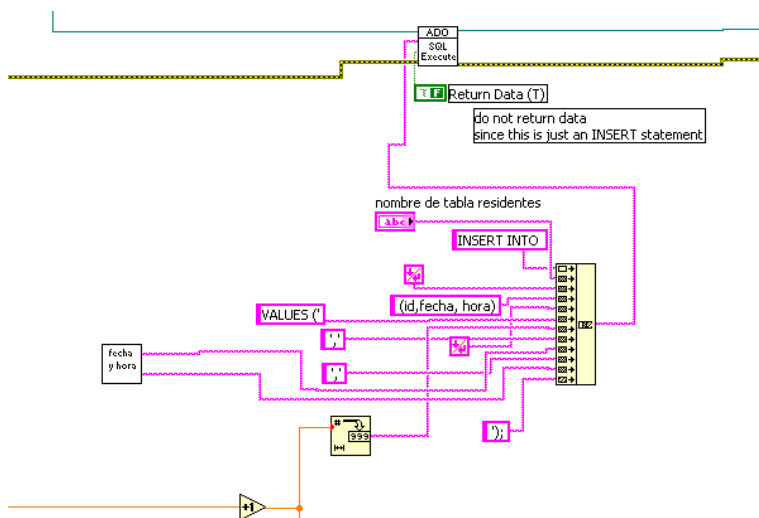


Figura 2.17

Para guardar en la base de datos, se escribe lo que se llama un “query” (línea de comandos SQL), y se lo pone como parámetro a la función SQL execute. El Query para insertar tiene este formato:

Nombre_de_la_tabla INSERT INTO (campos de la tabla)
VALUES ('valores de cada campo');

Con esto insertamos la fecha, la hora y el id en la base de datos.



Figura 2.18

Si no es uno, pero el valor UDP es tres, no se realiza ninguna acción.

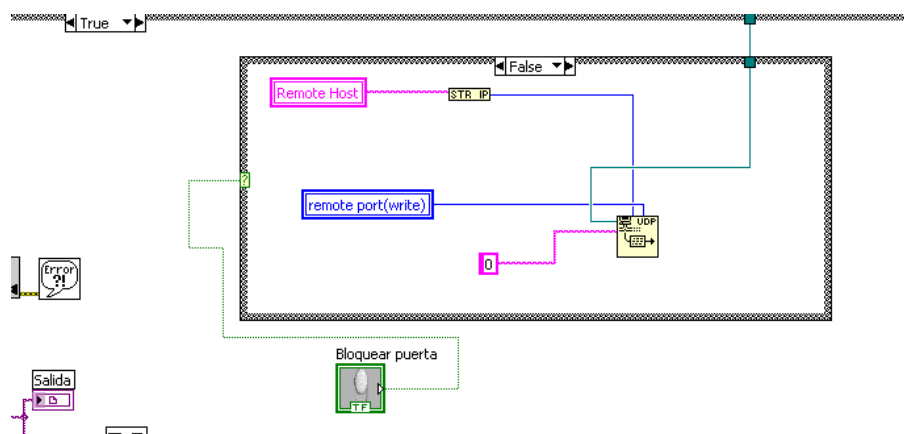


Figura 2.19

Si el valor UDP es cuatro, se escribirá en el puerto del PIC un cero para abrir la plumilla de acceso de los usuarios que están por salir si es que el switch de bloquear puerta está apagado. Si el switch está encendido, no se escribirá nada.

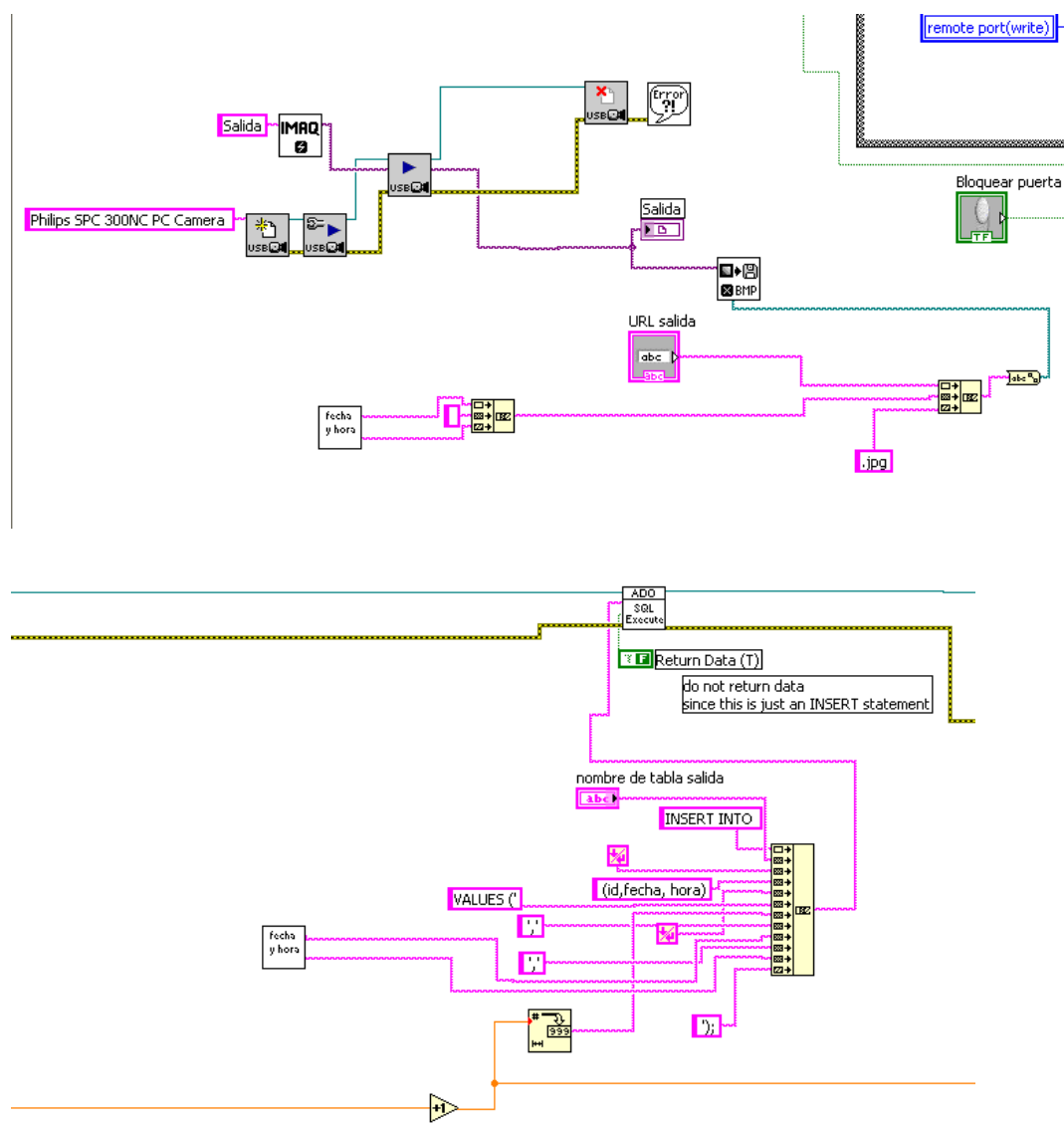


Figura 2.20

Luego, de la misma manera que cuando llegaba el residente, se procede a guardar la imagen en el directorio de el ordenador local, y se procede también a

guardar en la base de datos el id, la fecha y la hora en que el usuario va a salir.

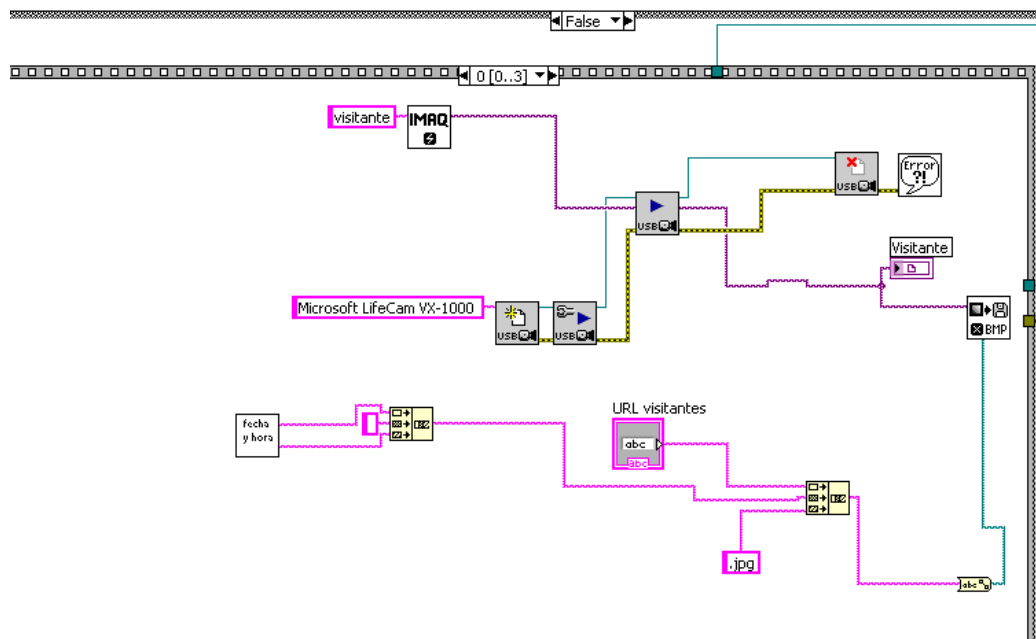


Figura 2.21

Si el valor UDP no es uno, ni cuatro ni tres, debe ser cero, por lo que el usuario que ha llegado es un visitante y lo primero que se hace es que se toma la foto del visitante y se la guarda en el directorio especificado por el control que se encuentra en el panel frontal, con la fecha y hora como nombre.

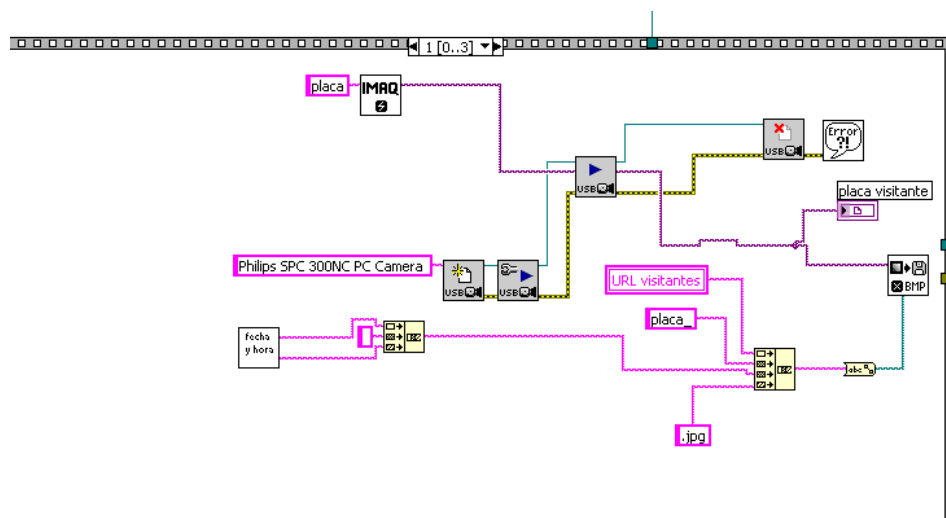


Figura 2.22

Se procede a guardar la imagen de la placa del carro en el directorio especificado, pero indicando que al nombre se le antepondrá la palabra placa_.

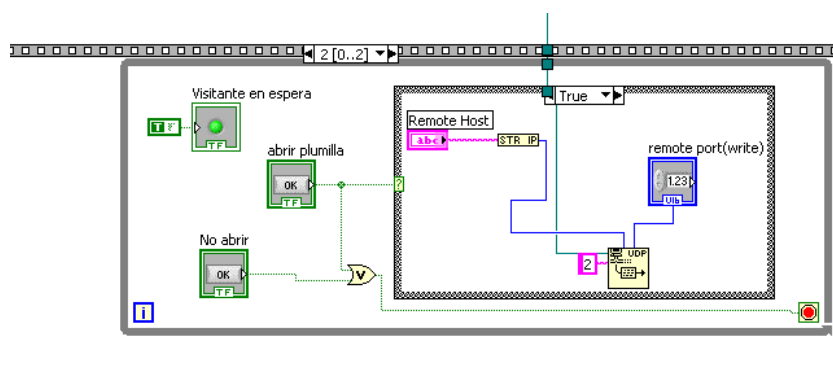


Figura 2.23

Se enciende el LED de visitante en espera, y se espera a que la persona que controla el sistema indique si

de caracteres: la primera con la fecha y la segunda con la hora.

2.2.2.5 Panel frontal del reporte en LABVIEW

A continuación mostraremos el panel frontal del reporte de entrada y salida de vehículos, en el cual podrá ver la fecha y la hora de ingreso de vehículos, y demás opciones.

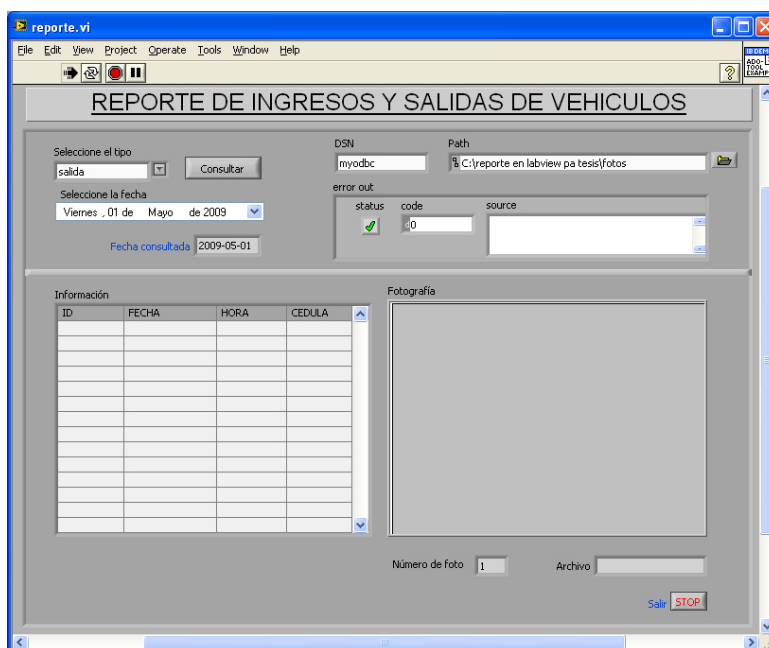


Figura 2.27

2.2.2.6 Explicación del panel frontal del reporte

Antes de utilizar el programa se debe configurar, el programa contiene parámetros que deben ser puestos correctamente, uno es el DSN y el otro es la ruta donde están las fotos, los pasos de esa configuración se detallan en el manual de usuario.

En la parte superior izquierda disponemos de un control donde se pueden seleccionar el tipo de consulta, es decir salida, visitantes y entrada, para seleccionar la fecha de la consulta se debe ir al control que posee un calendario y presionar en el día deseado, una vez realizado los pasos anteriores se puede presionar en un botón que tiene una etiqueta consultar, en la parte central izquierda veremos una lista de los resultados de nuestra consulta, se podrá ver la fecha, hora , podemos dar un clic con el ratón en uno de los ítems de la lista y automáticamente visualizaremos en la parte central derecha la fotografía del vehículo, en la parte inferior derecha veremos el numero de la foto y el nombre del archivo.

2.2.2.7 Diagrama de bloques del reporte en LABVIEW

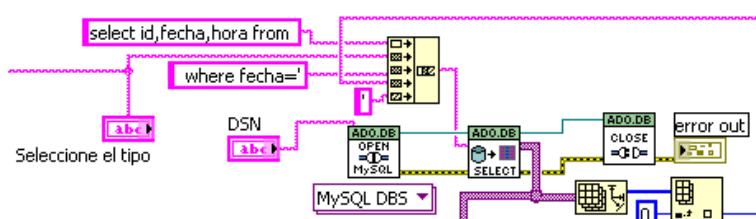


Figura 2.28

En esta parte del bloque general, se muestra la gestión con la base de datos, el DSN se conecta al bloque de consultas de ADO.net, la consulta también es introducida en este bloque, el tipo de consulta, que puede ser residentes, visitantes o salida, este bloque devuelve un arreglo con los resultados, el cual es procesado mas adelante.

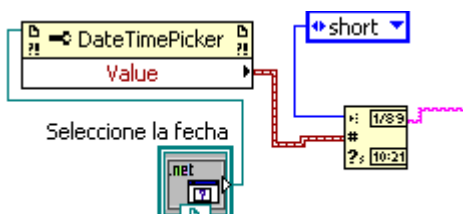


Figura 2.29

Para el manejo de la fecha, se dispone de un control llamado DateTimePicker, en mismo que retorna la fecha para realizar la consulta a la base de datos.

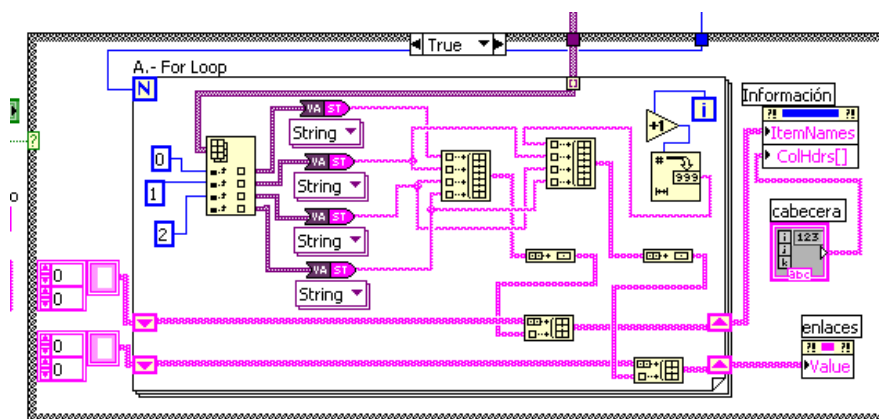


Figura 2.30

Este es el lazo que lee el arreglo devuelto por el bloque de consulta, se cambia el tipo de dato y se guarda en arreglos para después ser puestos en el control que muestra los resultados en una lista.

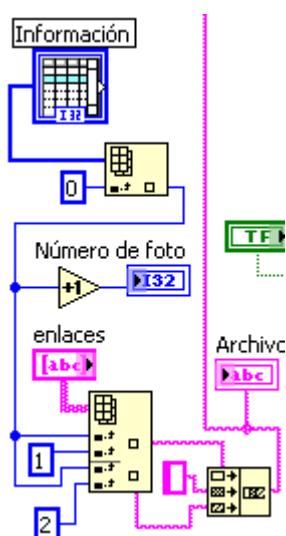


Figura 2.31

En esta parte se selecciona el archivo que ha sido escogido al dar clic con el ratón, de la lista de resultados se escoge y se concatena la fecha y hora para ir al repositorio de archivos y mostrarlo.

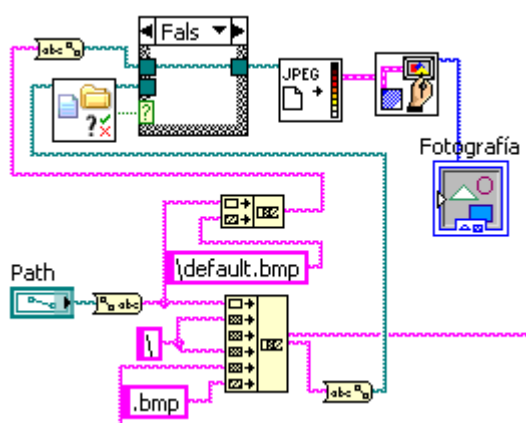


Figura 2.32

Este bloque recibe el nombre del archivo y lo une con la ruta de la configuración, y lo muestra en el panel, si no encuentra la imagen en formato bmp, muestra una imagen por defecto.

2.2.3 Bloque de Barrera

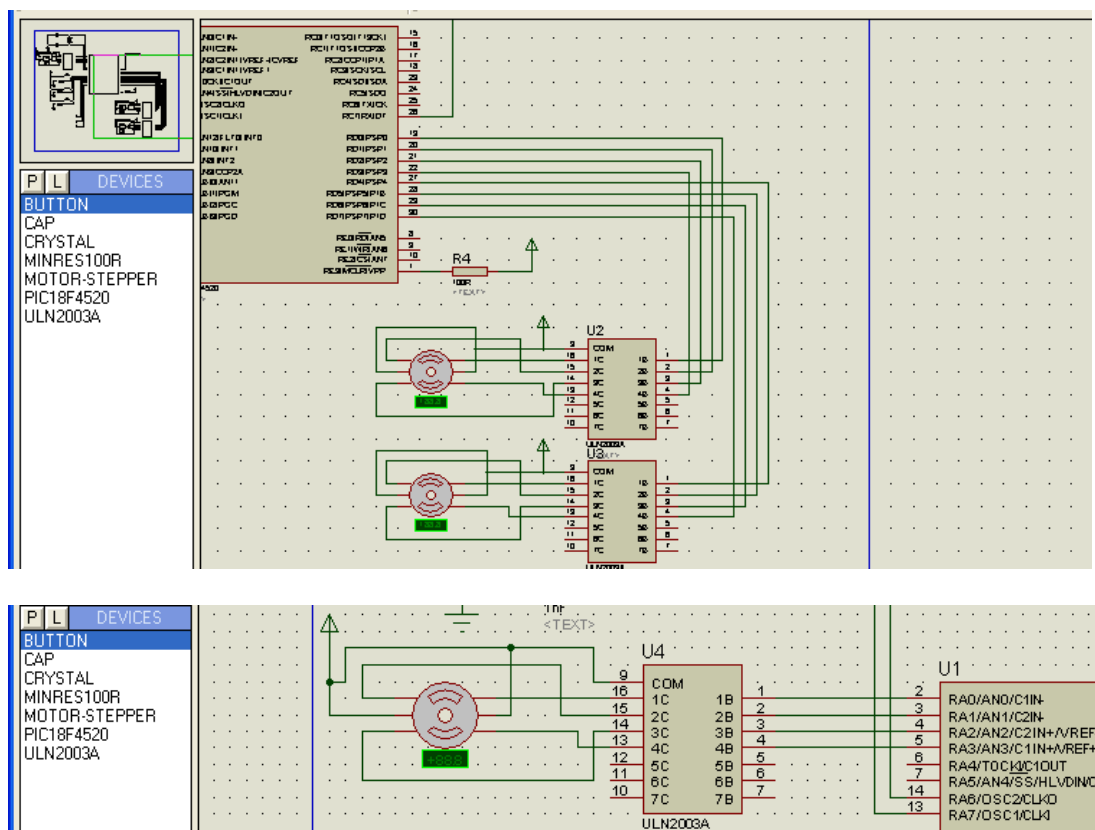
Este bloque consiste en el control de las plumillas de acceso del sistema. En esta sección detallaremos su funcionamiento.

2.2.3.1 Captura de información proveniente del ordenador vía Ethernet y salida de control del motor de paso

Luego que el programa en LabView envía la información indicando que puerta debe abrirse, el programa en MikroBasic recepta esa información mediante la función `spi_ethernet_dopacket()` que ejecuta inmediatamente la función `Spi_Ethernet_UserUDP`. En esta función ejecutaremos las rutinas para abrir la plumilla. La salida del PIC serán pulsos de bajo voltaje (0V) o de alto voltaje (5V), que, en el correcto orden, harán que el motor de paso se mueva hacia donde deseamos.

2.2.3.2 Esquemático del circuito de fuerza

A continuación se muestra como deben estar conectados los motores de paso al PIC



intermediario. La razón de usar el ULN2003 es que el motor de paso necesita una mayor cantidad de corriente para funcionar. El arreglo interno de transistores en configuración Darlington del ULN2003 nos dará la cantidad de corriente necesaria para que cada motor funcione a la perfección. Aunque no se muestra en el esquemático, es importante conectar los pines del PIC a la bobina correcta del motor como se muestra a continuación:

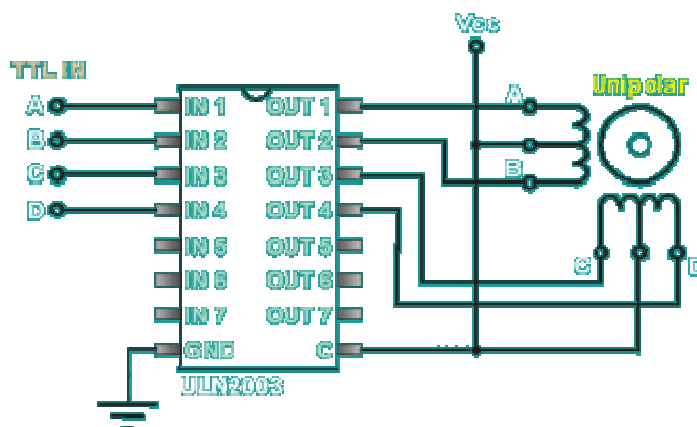


Figura 2.34

2.2.3.4 Porción del Programa del PIC18F4520 para control del motor de paso

```
if getRequest="1" then
```

'rutina para controlar el motor puerta residentes

for i=0 to 3

PORTD.0=1

PORTD.1=0

PORTD.2=0

PORTD.3=0

delay_ms(500)

PORTD.0=0

PORTD.1=1

PORTD.2=0

PORTD.3=0

delay_ms(500)

PORTD.0=0

PORTD.1=0

PORTD.2=1

PORTD.3=0

delay_ms(500)

PORTD.0=0

PORTD.1=0

PORTD.2=0


```
    PORTD.3=1
    delay_ms(500)
next i
while (PORTB.5=0) wend
delay_ms(250)
while (Button(PORTB, 5, 1, 1))wend
for i=0 to 3
    PORTD.0=0
    PORTD.1=0
    PORTD.2=1
    PORTD.3=0
    delay_ms(500)
    PORTD.0=0
    PORTD.1=1
    PORTD.2=0
    PORTD.3=0
    delay_ms(500)
    PORTD.0=1
    PORTD.1=0
    PORTD.2=0
    PORTD.3=0
    delay_ms(500)
```

```
        PORTD.0=0
        PORTD.1=0
        PORTD.2=0
        PORTD.3=1
        delay_ms(500)
    next i
else
    if getRequest="0" then 'si se debe abrir la puerta de
salida
        for i=0 to 3
            PORTD.4=1
            PORTD.5=0
            PORTD.6=0
            PORTD.7=0
            delay_ms(500)

            PORTD.4=0
            PORTD.5=1
            PORTD.6=0
            PORTD.7=0
            delay_ms(500)
            PORTD.4=0
```

```
PORTD.5=0
PORTD.6=1
PORTD.7=0
delay_ms(500)
PORTD.4=0
PORTD.5=0
PORTD.6=0
PORTD.7=1
delay_ms(500)
next i
while (PORTB.6=0) wend
delay_ms(250)
while (Button(PORTB, 6, 1, 1))wend
for i=0 to 3
PORTD.4=0
PORTD.5=0
PORTD.6=1
PORTD.7=0
delay_ms(500)
PORTD.4=0
PORTD.5=1
PORTD.6=0
```

```
    PORTD.7=0
    delay_ms(500)
    PORTD.4=1
    PORTD.5=0
    PORTD.6=0
    PORTD.7=0
    delay_ms(500)
    PORTD.4=0
    PORTD.5=0
    PORTD.6=0
    PORTD.7=1
    delay_ms(500)
  next i
else
  if getRequest="2" then 'VISITANTES
  for i=0 to 3
    PORTA.0=1
    PORTA.1=0
    PORTA.2=0
    PORTA.3=0
    delay_ms(500)
```

```
PORTA.0=0
PORTA.1=1
PORTA.2=0
PORTA.3=0
delay_ms(500)
PORTA.0=0
PORTA.1=0
PORTA.2=1
PORTA.3=0
delay_ms(500)
PORTA.0=0
PORTA.1=0
PORTA.2=0
PORTA.3=1
delay_ms(500)
next i
while (PORTB.7=0) wend
delay_ms(250)
while (Button(PORTB, 7, 1, 1))wend
for i=0 to 3
    PORTA.0=0
    PORTA.1=0
```

```
PORTA.2=1
PORTA.3=0
delay_ms(500)
PORTA.0=0
PORTA.1=1
PORTA.2=0
PORTA.3=0
delay_ms(500)
PORTA.0=1
PORTA.1=0
PORTA.2=0
PORTA.3=0
delay_ms(500)
PORTA.0=0
PORTA.1=0
PORTA.2=0
PORTA.3=1
delay_ms(500)
next i
end if
end if
```

2.2.3.5 Explicación detallada del programa del PIC18F4520 para controlar los motores

Tenemos en el esquemático del circuito tres motores, que son los motores de las plumillas de acceso de los visitantes, residentes y de los usuarios que van a salir. El PIC recibirá tres datos vía UDP para el control de los motores: 1 para el motor de la plumilla de residentes; 0 para el motor de la plumilla de salida; y 2 para el motor de la plumilla de visitantes. Por lo tanto, lo primero que hace el programa es preguntar si ha llegado uno de estos datos. Si no ha llegado ninguno, pues la función no realiza ninguna acción. Si se detecta un uno, se controlará el motor conectado a los pines RD0 – RD3. Como se puede ver, primero, se activa cada bobina en orden, para hacer que el motor se mueva en sentido horario. Se hacen 4 pasos, y luego se los repite 4 veces; o sea que en total se realizan 16 pasos. Considerando que el motor tiene 7,5° por paso, se tienen aproximadamente 120°. Luego de ello, se valida si el sensor de esa plumilla está en un nivel bajo de voltaje, y

mantiene la plumilla alzada mientras el sensor se encuentre en cero mediante el primer lazo while. Con esto, logramos que la plumilla se mantenga alzada si es que el carro aun no cruza por la garita. Luego, si el sensor cambia a uno (voltaje alto – 5V), significa que el automóvil está cruzando a través de la garita, por lo que debemos mantener la plumilla alzada. Para ello, con el segundo lazo while, nos mantenemos con la plumilla alzada mientras el sensor siga indicando un voltaje alto. Insertamos un retardo de 250 mseg entre los dos lazos while de control para evitar los antirebotes cuando el automóvil está recién empezando a pasar. Cuando el sensor ya pasa a un voltaje bajo, se sale del while, y se acciona el motor 120° pero en sentido anti horario. Para ello, accionamos las bobinas en orden inverso, logrando que la plumilla de acceso regrese a la posición original.

Para los otros dos motores, se sigue el mismo procedimiento, con la diferencia que el sensor está conectado a un pin diferente del PIC que el sensor anterior, y que los motores están conectados a otros pines diferentes del PIC.

2.2.4 Bloque de Sensores

En esta sección describiremos las conexiones de los sensores y su funcionamiento.

2.2.4.1 Detección mediante sensores

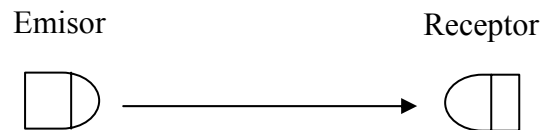
Los sensores infrarrojos usados en este proyecto, nos servirán para detectar en nuestro prototipo si un carro ya pasó la plumilla de acceso o no. Por la simplicidad de los sensores infrarrojos, y por ser económicos, se eligieron este tipo de sensores, pero evidentemente para mayores distancias como es el caso de un automóvil real, se necesitarán sensores de proximidad de más potencia y mayor alcance.

En el mercado hay diversos detectores vehiculares, donde los más usados son los de espira o bucle de piso. También se pueden usar sensores infrarrojos de proximidad pero de mayor potencia. En este proyecto,

se hará un modelo a escala del sistema de seguridad por lo que un sensor infrarrojo será suficiente.

COMUNICACIÓN

En el modo punto a punto (ver figura), el emisor y el receptor deben estar lo más cercanos posibles y en una alineación correcta. Se requiere una línea de vista entre las estaciones que se desee comunicar. Utilizamos tres circuitos, uno para el paso de residentes, otro para el paso de visitantes y el último para la salida del recinto privado.



Emisor

En el emisor se emplea la emisión infrarroja proveniente de un led emisor infrarrojo como fuente, este led está enviando todo el tiempo emisiones de luz infrarroja, que son imperceptibles para el ojo humano, esta luz infrarroja emitida esta siempre recibida por un fototransistor.

El circuito es simple, consiste en conectar un led infrarrojo a una fuente de energía para empezar a emitir, a continuación se muestra como están conectadas las partes del circuito.

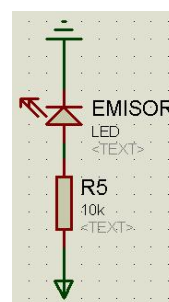


Figura 2.35

Receptor

El receptor consiste básicamente en un fototransistor que esta sensando todo el tiempo de forma ininterrumpida, esto le permite mantener una comunicación entre ellos, todo el tiempo entonces se estaría enviando al microcontrolador un alto, en caso de interrumpirse la señal, automáticamente el fototransistor ya no detecta la señal enviando de esta forma a la salida del colector la señal bajo, esta señal en enviada al microcontrolador que mantiene el brazo mecánico levantado mientras la señal del infrarrojo este interrumpida.

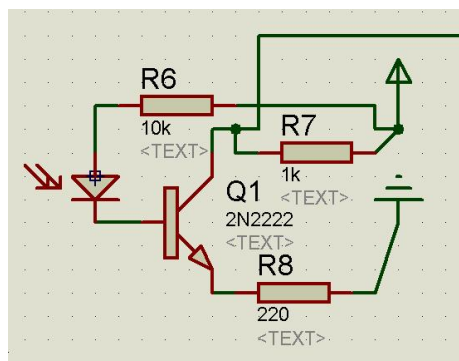


Figura 2.36

2.2.4.2 Conexión de los sensores al circuito de fuerza

Los sensores se conectan al circuito de fuerza mediante el microcontrolador PIC 18F4520, los sensores sin detectar el paso envía un cero lógico en todo momento, una vez detectado la interrupción al pasar un vehículo, envían un pulso alto indicando que un está de paso un carro y mantiene el uno lógico hasta que pase el vehículo, ocurrido esto se envía vuelve a enviar un cero lógico, el microcontrolador tiene programado una rutina que al detectar este cambio envía a girar al motor de paso para que procede a cerrar la barra mecánica.

2.2.4.3 Explicación del funcionamiento del circuito

El circuito que emite la una señal a través del sensor es muy simple, consiste en conectar a un diodo emisor infrarrojo un voltaje de 5V y una corriente de 5A. Esto hace que el diodo todo el tiempo este emitiendo la señal. En cuanto al receptor, consiste en un fototransistor que recibe la señal infrarroja y a través de un transistor 2N222 logra mantener en bajo cuando la tiene una señal emitida por el emisor, cuando se interrumpe la señal, esto hace que el transistor se sature y envíe al colector

un alto para que esa señal de alto sea enviada al microcontrolador indicando que hay un vehículo pasando.

CAPITULO 3

Simulación, implementación y pruebas

3.1 Pistas del PCB construidos

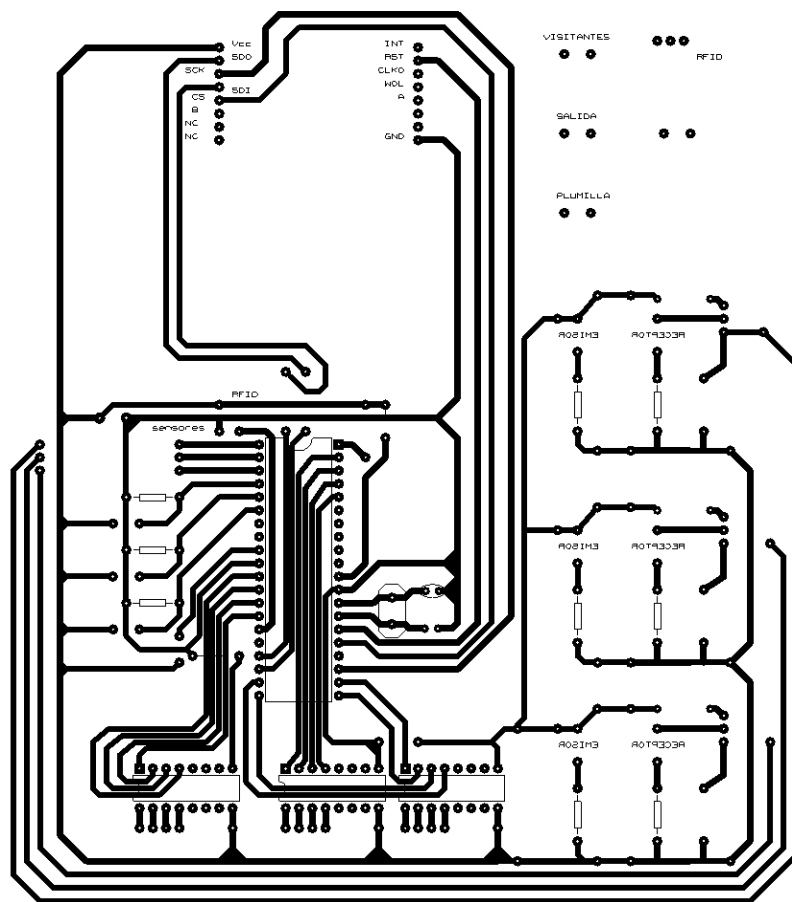


Figura 3.1

3.2 Resultados experimentales

De los resultados obtenidos durante el desarrollo, mostramos a continuación los datos obtenidos durante la conexión con Ethernet desde LabView, los datos ingresados a la base de datos durante un ingreso de algún vehículo, con su respectiva configuración.

Configuración inicial

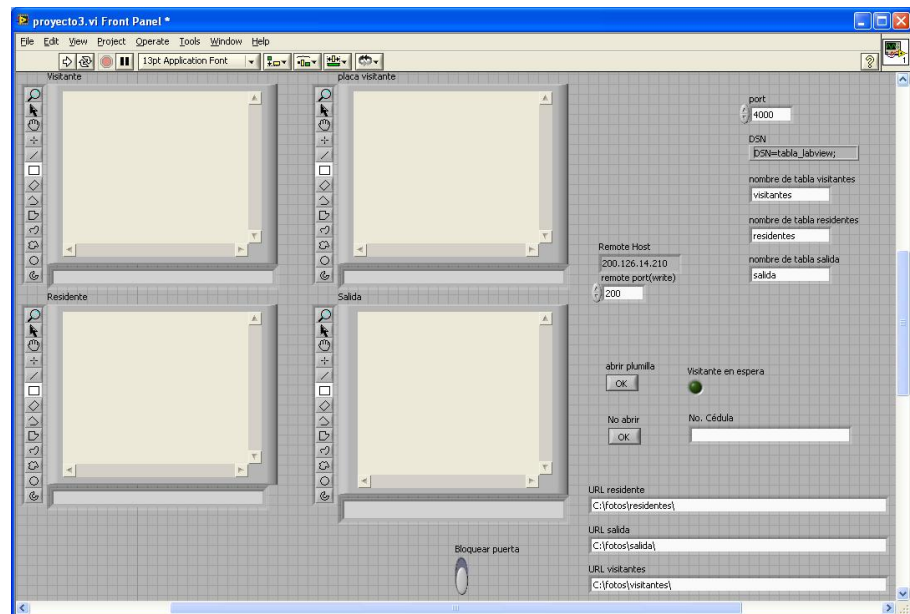


Figura 3.2

Ingreso de un residente

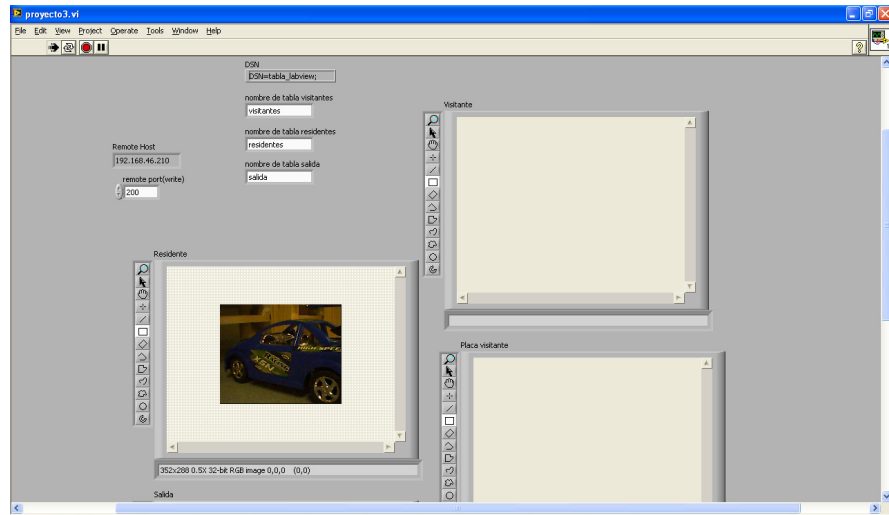


Figura 3.3

Registro en la base de datos del residente

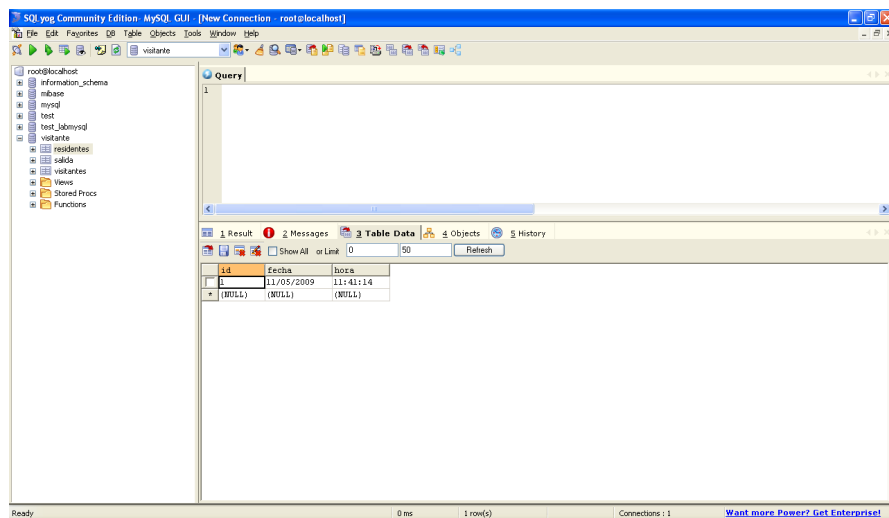


Figura 3.4

Verificación de ingreso de la foto del residente al sistema

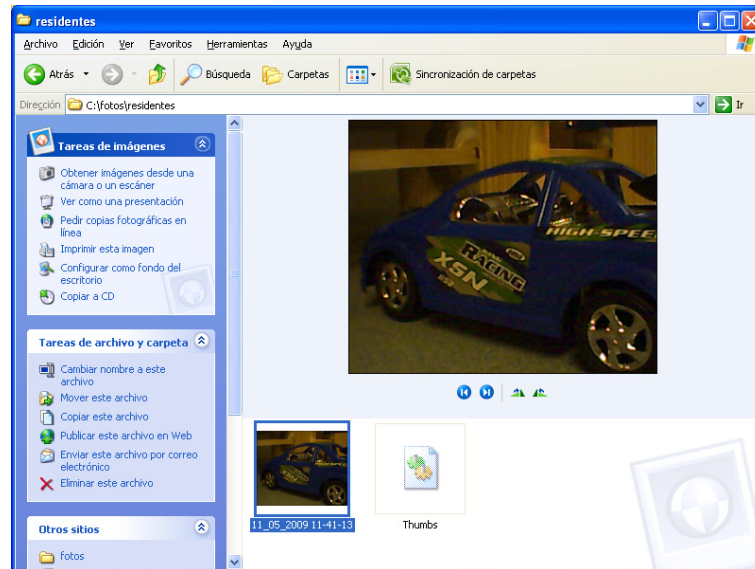


Figura 3.5

Ingreso de un visitante

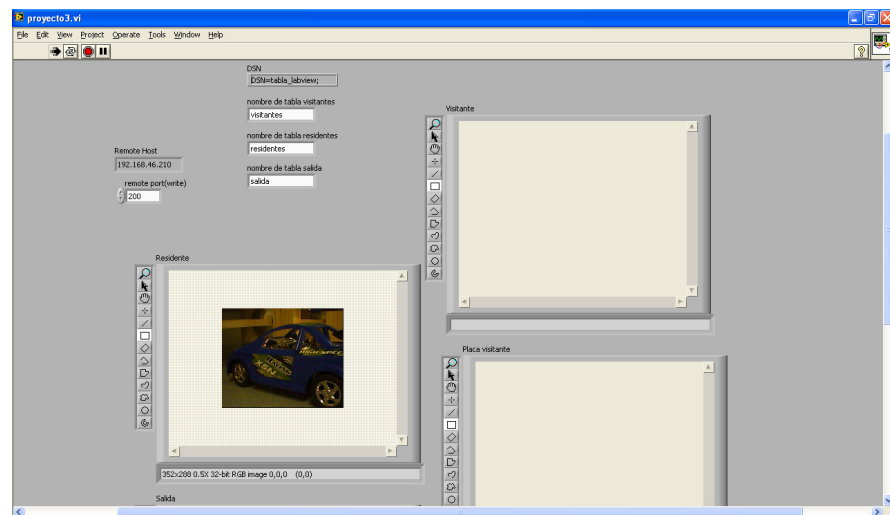


Figura 3.6

Registro en la base de datos del visitante

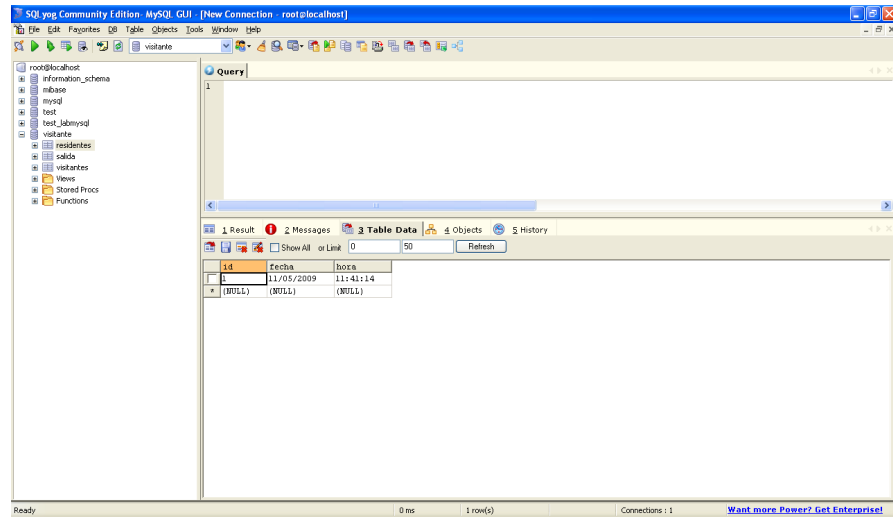


Figura 3.7

Verificación de ingreso de la foto del visitante al sistema

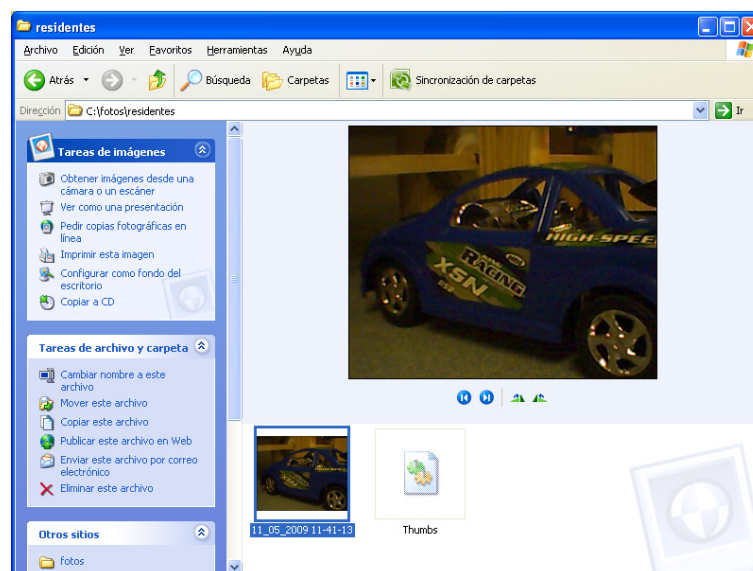


Figura 3.8

Salida de un vehículo

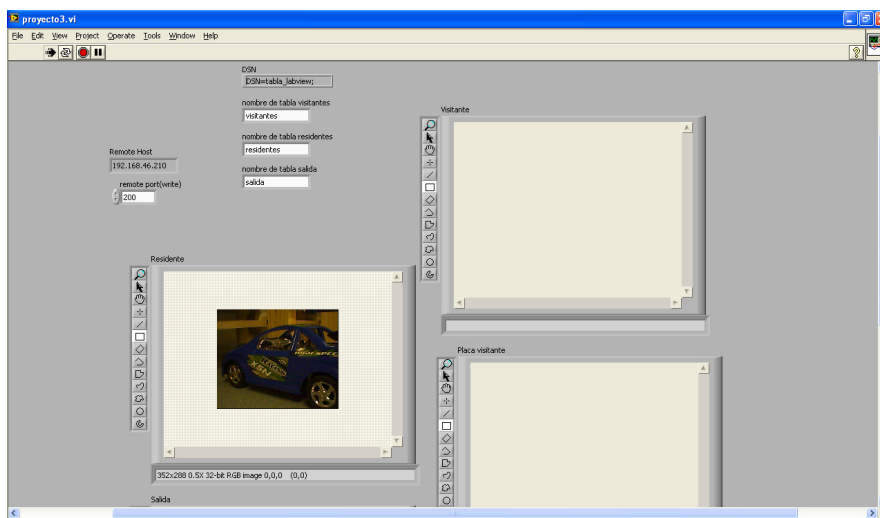


Figura 3.9

Registro en la base de datos del vehículo que sale

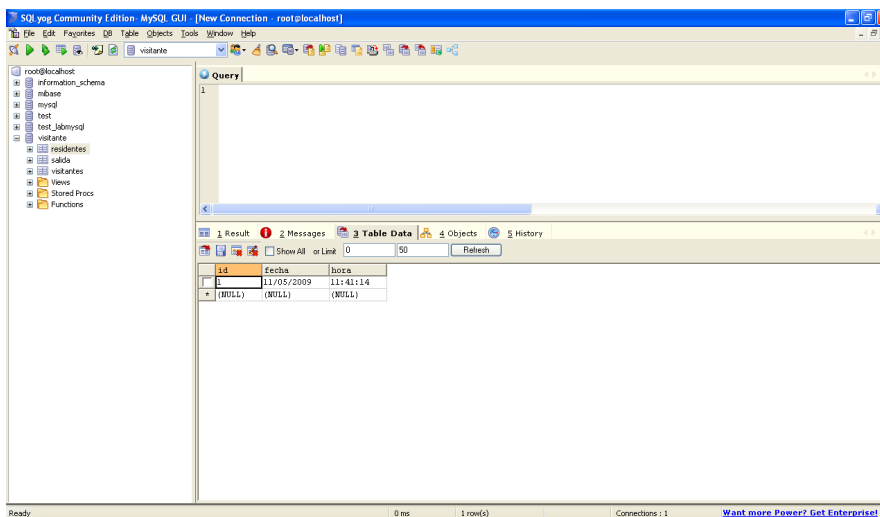


Figura 3.10

Verificación del registro de la foto del vehículo que sale al sistema

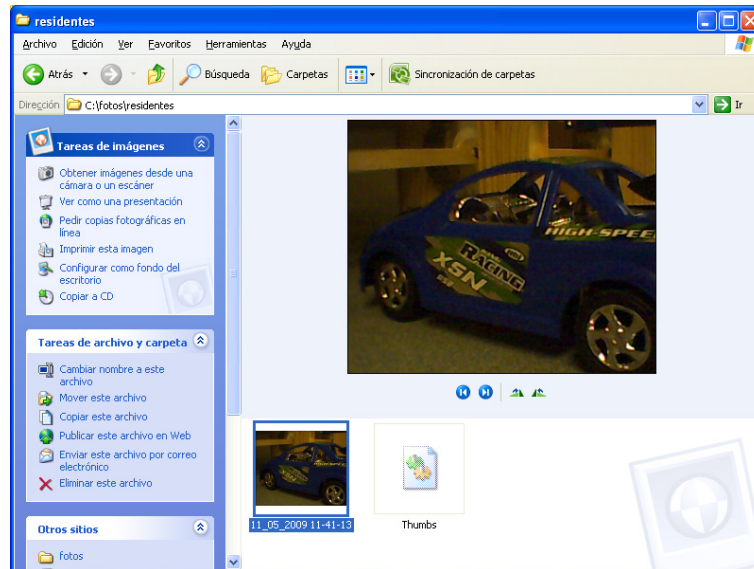


Figura 3.11

Vista frontal de la maqueta



Figura 3.12

Vista del circuito de conexión del microcontrolador y el modulo Ethernet

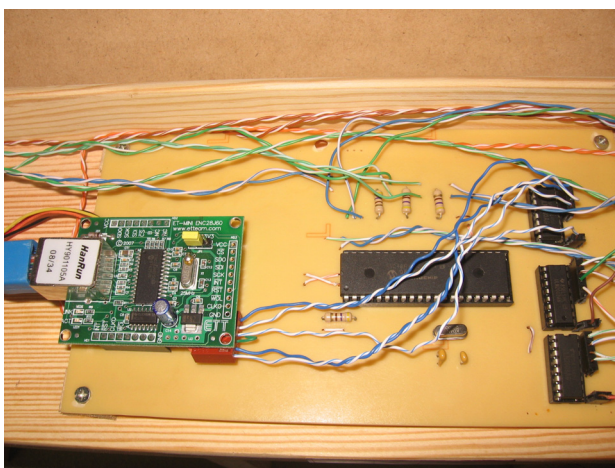


Figura 3.13

Vista del circuito de conexión entre los sensores al microcontrolador

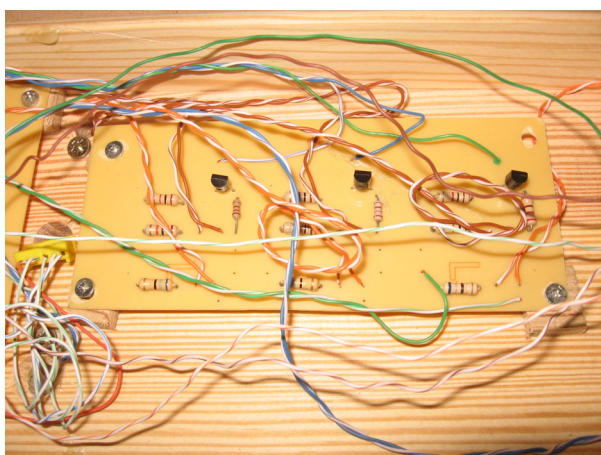


Figura 3.14

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. La identificación del residente en nuestro sistema de ingreso de vehículos se realizó con éxito por medio del sistema RFID, que nos permitió reconocer al usuario de manera efectiva por medio de un número único para los residentes.
2. La comunicación Ethernet entre el PIC y el programa de control de LABVIEW nos brindó gran escalabilidad en el proyecto en general, y eliminar las limitaciones de las distancias, ya que el computador donde se controla la garita puede estar muy lejos o muy cerca del sistema electrónico.
3. La programación en Basic para el PIC fue para nosotros una herramienta muy poderosa, en especial por la facilidad que tiene para realizar la comunicación Ethernet por medio de la librería SPI_Ethernet, donde las funciones de manejo de datos vía Ethernet ya están todas implementadas, y su uso es relativamente sencillo.
4. De la misma manera, las funciones en LABVIEW que manejan comunicación UDP, nos facilitaron bastante el trabajo de este proyecto por su fácil manejo y sencillo funcionamiento.

5. El módulo del ENC28J60 ya construido fue una gran ventaja para este proyecto, ya que no tuvimos que implementarlo, sino usarlo directamente. Con esto, y las funciones del software utilizado, la comunicación Ethernet fue una tarea más sencilla.

6. LABVIEW nos brindó grandes facilidades en el manejo de imágenes por cámaras web. Las funciones del NI VISION y el NI IMAQ se derivan en múltiples aplicaciones, de fácil entendimiento. Los ejemplos proporcionados en el paquete de NI VISION nos permitieron entender ampliamente el manejo de imágenes y de cámaras web en LABVIEW.

7. De la misma manera, LABVIEW resultó ser una herramienta poderosa en el manejo de bases de datos; la inserción de datos resultó ser una tarea sencilla con las funciones del paquete LABSQL (ADO functions). Solo se necesitó un poco de conocimiento de lenguaje SQL para manejo de queries de inserción de datos. Así igual fue necesario aprender ciertas cosas referentes a bases de datos para la creación de las tablas y los campos a utilizar.

8. El manejo de motores de paso simplificó el control de las plumillas de acceso, brindándonos la facilidad de levantar o bajar la plumilla la cantidad de grados que deseamos, sin mayores problemas de control.
9. El PIC18F4520 fue una perfecta elección para este proyecto, porque soporta perfectamente la comunicación SPI y por la cantidad de memoria ROM que supera los 4KB.
10. Aunque los sensores infrarrojos usados en este trabajo no se asemejan a los sensores usados si se fuera a realizar una aplicación real del proyecto, la entrada del PIC proveniente de los sensores, simplemente sensa un voltaje alto, o un voltaje bajo, por lo que los sensores pueden ser de diversos tipos, con tal de que su salida produzca un voltaje bajo de 0V y un voltaje alto de 5V.

RECOMENDACIONES

1. Es necesario recordar que las cámaras web usadas en este proyecto deben soportar Direct Show, por lo que es recomendable que antes de usar el proyecto, se aseguren que las aplicaciones en LABVIEW detectan las cámaras correctamente, ejecutando el "Enumerate.vi".

2. Es recomendable usar una fuente que soporte suficiente cantidad de corriente, para los motores y los sensores, de lo contrario, pueden haber errores en el funcionamiento del circuito.

3. También es importante recordar que la dirección IP de la computadora que monitoreará el sistema debe ser grabada dentro de la programación del PIC antes de que el sistema empiece a funcionar.

4. La creación de la base de datos con sus respectivas tablas y sus respectivos campos dentro de cada tabla, corre por cuenta del usuario. Es necesario crear la base de datos y sus tablas antes de empezar a correr el programa.

5. De igual manera, es recomendable que la persona que va a controlar el sistema de ingreso de vehículos, revise bien los campos del panel frontal de LABVIEW e ingrese los valores correctos dependiendo de la computadora donde está trabajando.

GLOSARIO DE TERMINOS

Labview Software que provee herramientas para la simulación, utiliza el lenguaje G para su programación.

Mikrobasic Software que genera archivos para la programación de microcontroladores, utiliza el lenguaje basic como interfaz con el usuario.

ANEXOS

Como anexos a este proyecto, se adjunta los siguientes direcciones donde se detalla a profundidad elementos empleados.

A) Datasheet del microcontrolador 18F4520

<http://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf>

B) Datasheet del circuito integrado ENC28J60

<http://www.blog.lausdahl.com/file.axd?file=User's+manual+ET-MINI+ENC28J60.pdf>

<http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf>

C) Datasheet del módulo RFID reader

<http://www.parallax.com/dl/docs/prod/audiovis/RFID-Reader-v1.2.pdf>

D) Datasheet del circuito integrado ULN 2003

<http://www.datasheetcatalog.org/datasheet2/f/0c6x6a46ig46qlxf3j2qsaii8o3y.pdf>

E) Manual de usuario del programa de control.

Instalación de aplicaciones:

A continuación se detallarán las aplicaciones que deben ser instaladas y los pasos previos necesarios para que el programa de LABVIEW funcione correctamente.

1.- LABVIEW 8.5

2. - NI v8.5 Vision Acquisition Software

3. - NI v8.5 Vision Development Module

4.- ni_imaq_usb_installer_86

5.- Las LabSQL ADO functions deben ser copiadas en la carpeta de user de LABVIEW 8.5.

6. - mysql-connector-odbc-3.51.27-win32

7. - mysql-5.0.77-win32 ó MySQL yog

8.- Se deben instalar al menos cuatro cámaras con sus respectivos controladores.

9.- Se debe crear el DSN de sistema en las herramientas administrativas.

10.- Se debe crear la base de datos con sus respectivas tablas:

Visitantes, que contendrá id, fecha, hora y cédula

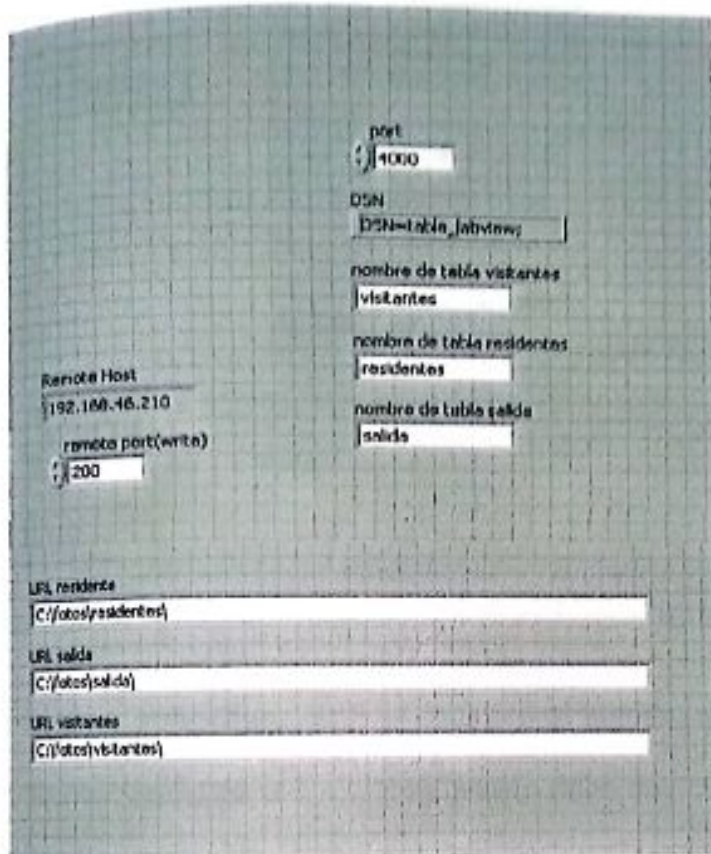
Residentes, que contendrá id, fecha, hora

Salida, que contendrá id, fecha, hora

Panel de control

A continuación se explicará cómo manipular el panel de control de LABVIEW para el acceso de vehículos del sistema.

Configuración Inicial



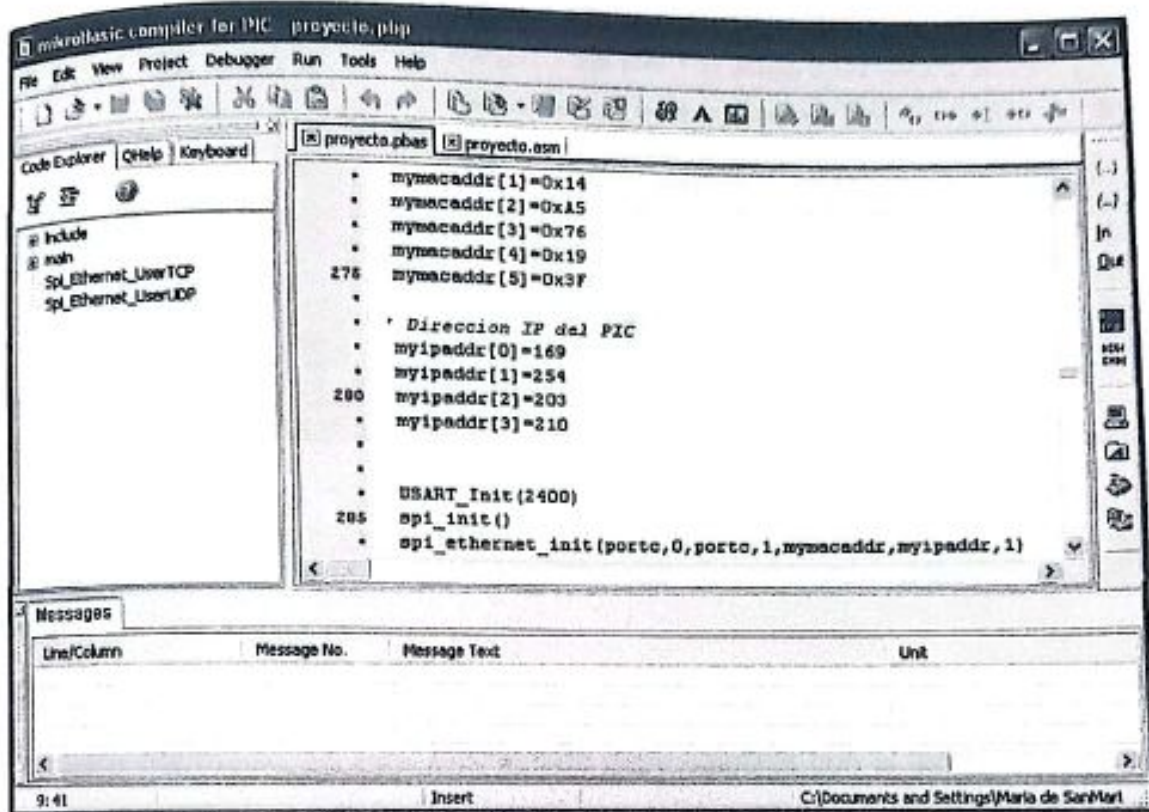
The screenshot shows the initial configuration panel with the following fields and values:

- port: 4000
- DSN: DSN=tabla_labview
- nombre de tabla visitantes: visitantes
- nombre de tabla residentes: residentes
- nombre de tabla salida: salida
- Remote Host: 192.168.46.210
- remote port(write): 200
- URL residente: C:/fotos/residentes
- URL salida: C:/fotos/salida
- URL visitantes: C:/fotos/visitantes

Antes de iniciar el programa de control del sistema de ingreso de vehículos, es necesario configurar ciertos parámetros iniciales que permitirán que el programa funcione correctamente. Estos parámetros son:

-Remote Host: En este campo se escribirá la dirección IP del PIC del sistema de ingreso de vehículos. La dirección IP del PIC se encuentra grabada en el

programa MikroBasic del PIC que también se incluye en el paquete de software de este producto:

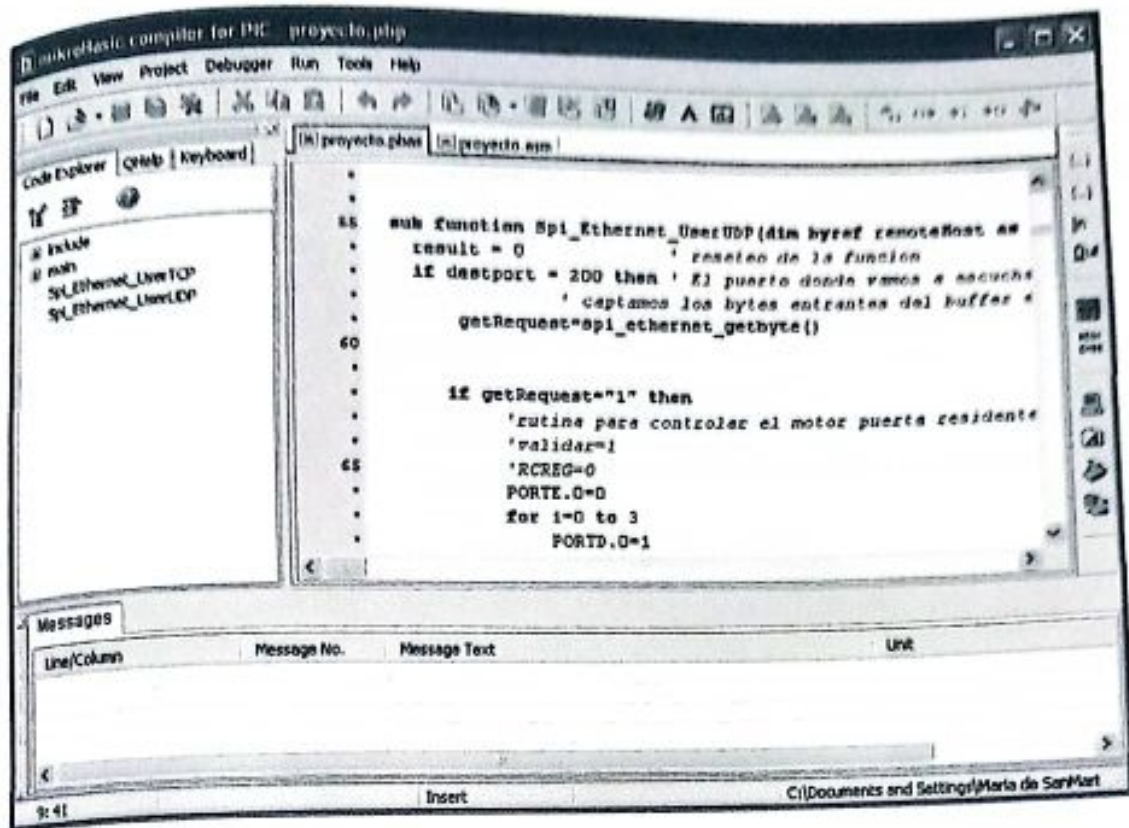


```

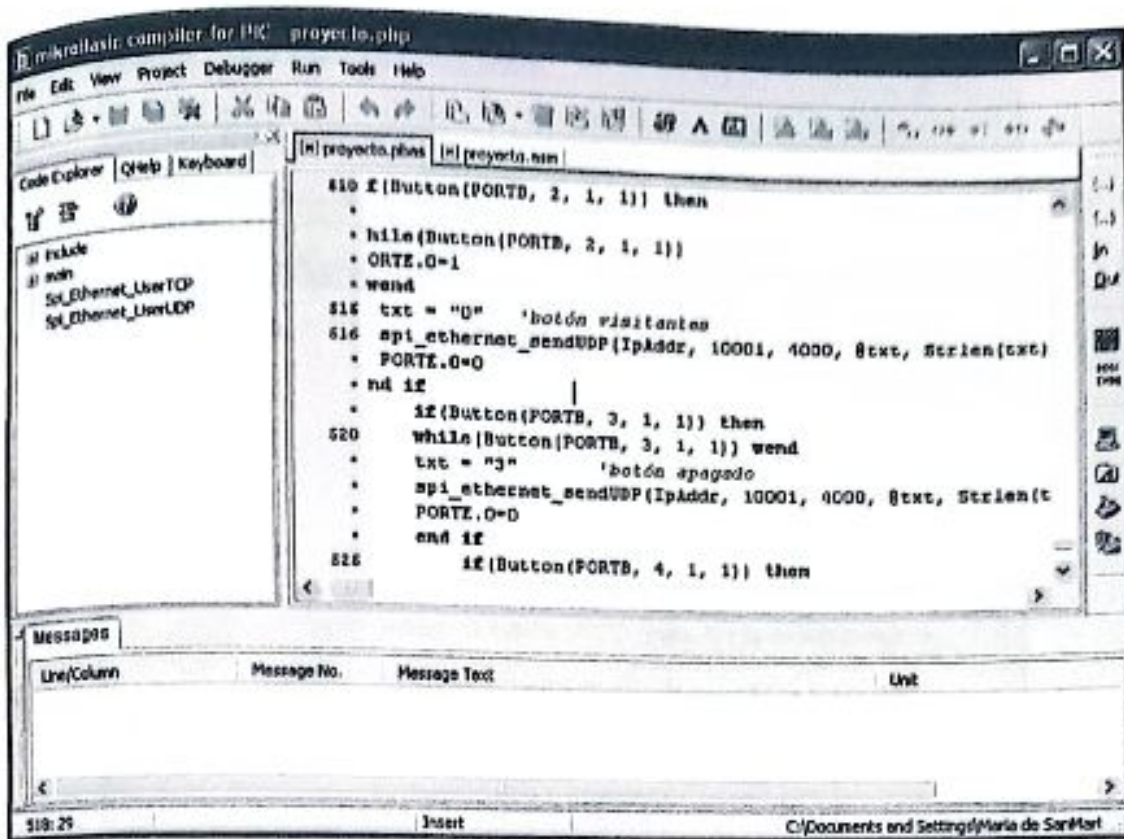
* mymacaddr[1]=0x14
* mymacaddr[2]=0x15
* mymacaddr[3]=0x76
* mymacaddr[4]=0x19
276 mymacaddr[5]=0x3F
*
*   * Direccion IP del PIC
* myipaddr[0]=169
* myipaddr[1]=254
280 myipaddr[2]=203
* myipaddr[3]=210
*
*
*   USART_Init(2400)
285 spi_init()
* spi_ethernet_init(porto,0,porto,1,mymacaddr,myipaddr,1)

```

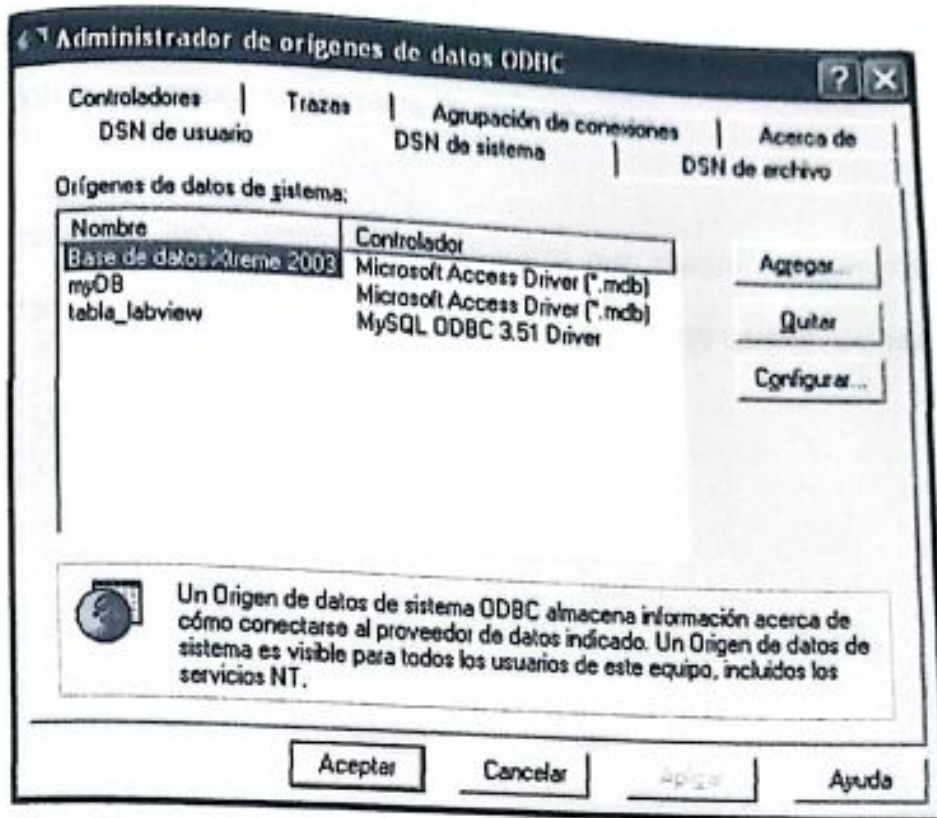
-Remote port(write): Aquí se indicará el número de puerto donde se enviará la información hacia el PIC. Este puerto debe coincidir con el puerto desde donde el PIC escucha la recepción de paquetes UDP especificado en el programa MIKROBASIC del PIC:



-Port: Es el número de puerto donde LabView va a escuchar los paquetes enviados por medio del PIC. Este puerto debe coincidir con el tercer parámetro de la función `spi_ethernet_sendUDP` en el programa MIKROBASIC:



-DSN: Especifica el nombre del DSN creado en las herramientas administrativas de Windows para lograr la conectividad entre nuestro programa y la base de datos. El formato es: DSN= "nombre del DSN". Para ver los DSN que se tienen en el sistema se va a Panel de control -> Herramientas Administrativas -> Orígenes de Datos ODBC -> DSN de Sistema. Escogemos el DSN de MySQL para guardar datos en dicha base de datos.



-Nombre de tabla visitantes: Es el nombre de la tabla dentro de la base de datos donde se guardará la información de los visitantes.

-Nombre de tabla Residentes: Es el nombre de la tabla dentro de la base de datos donde se guardará la información de los residentes.

-Nombre de tabla Salida: Es el nombre de la tabla dentro de la base de datos donde se guardará la información de los usuarios que saldrán del sistema

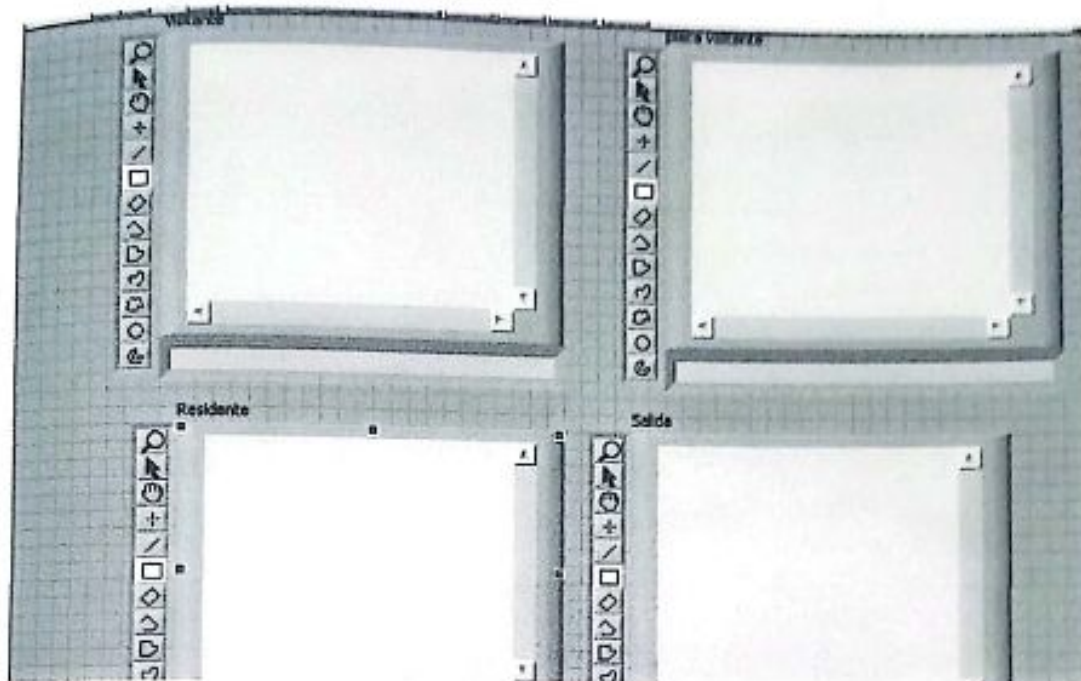
-URL residentes: En este campo se guardará la ruta dentro del computador de control donde se guardarán las fotos de los residentes.

-URL visitantes: En este campo se guardará la ruta dentro del computador de control donde se guardarán las fotos de los visitantes.

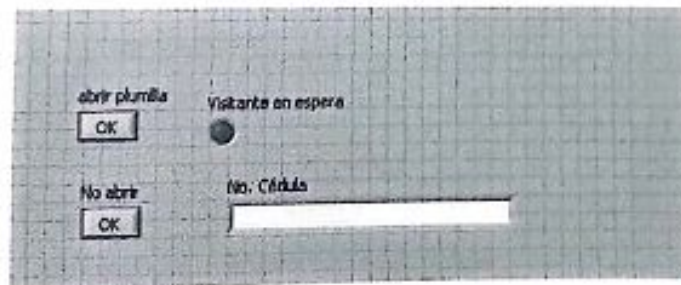
-URL Salida: En este campo se guardará la ruta dentro del computador de control donde se guardarán las fotos de los usuarios que saldrán del sistema.

Manejo del programa

En los 4 visores de imágenes se visualizarán las fotos tomadas de los diferentes usuarios que han llegado o salido del sistema.



Al llegar un visitante, se encenderá el LED "Visitante en espera". En ese momento, se ingresará la cédula del visitante, y con los botones "abrir plumilla" y "no abrir" se controlará el acceso del visitante.



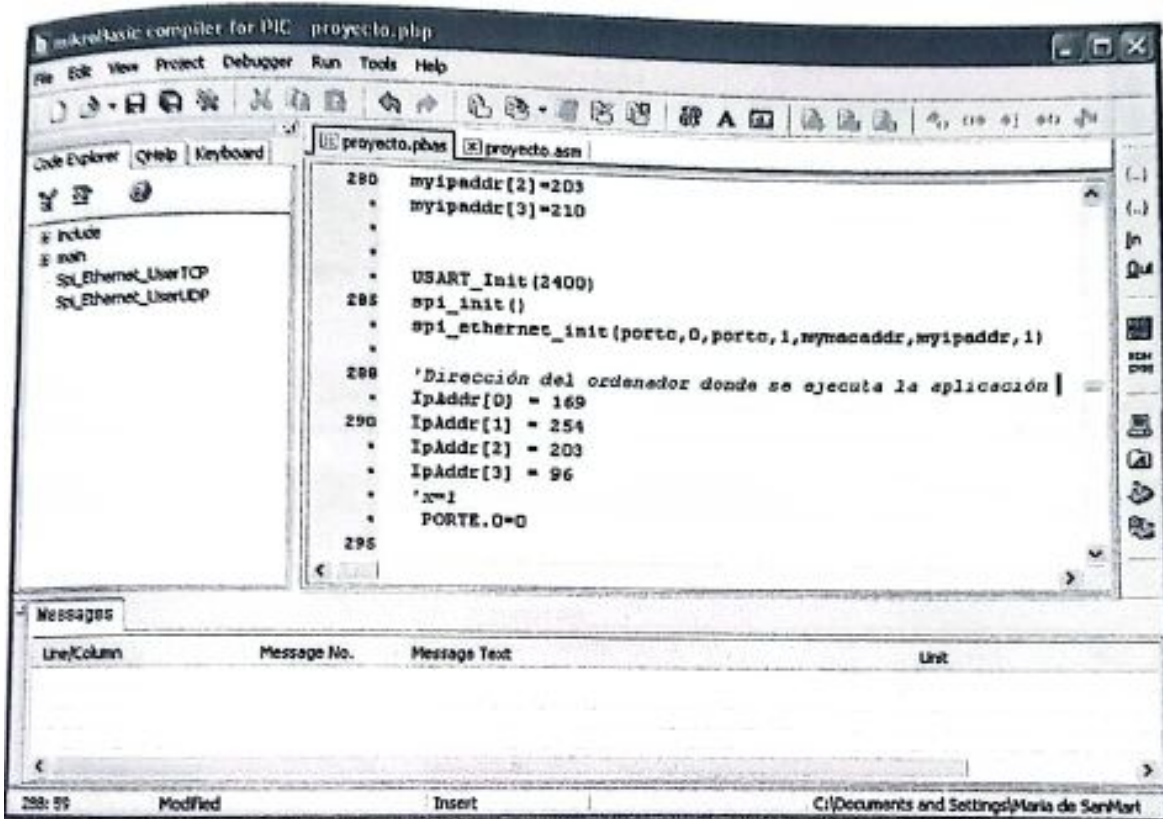
En el momento que un usuario quiera salir, la plumilla de acceso se le abrirá automáticamente a menos que el switch "bloquear puerta" esté encendido (Posición alta).



La validación de ingreso de los residentes es automática por medio de su tarjeta RFID.

Programación del PIC

Si se necesita cambiar el computador de control donde es ejecutado este programa, es necesario reprogramar el PIC, indicando la nueva dirección IP hacia donde la información es enviada:



The screenshot shows the MikroBasic compiler for PIC interface. The main window displays the following code:

```
280 myipaddr[2]=203
    * myipaddr[3]=210
    *
    *
    * USART_Init(2400)
285 spi_init()
    * spi_ethernet_init(porto,0,porto,1,mymacaddr,myipaddr,1)
    *
290 'Dirección del ordenador donde se ejecuta la aplicación |
    * IpAddr[0] = 169
290 IpAddr[1] = 254
    * IpAddr[2] = 203
    * IpAddr[3] = 96
    * 'x=1
    * PORTE.0=0
295
```

The Messages window at the bottom is empty, with columns for Line/Column, Message No., Message Text, and Unit. The status bar at the bottom shows '298: 59 Modified Insert C:\Documents and Settings\Maria de SenMart'.

De la misma manera, se pueden cambiar el número de puertos UDP para el envío y recepción de paquetes, siendo la variable "destport" el puerto para escuchar los paquetes, y el tercer parámetro de la función pi_ethernet_sendUDP, el puerto de escritura hacia el computador.

mikroBasic compiler for PIC - proyecto.php

File Edit View Project Debugger Run Tools Help

Code Explorer Help Keyboard

projecto.pbas | proyecto.asm

```

55 sub function Spi_Ethernet_UserUDP(dim bytes remoteHost as
    result = 0 'reseteo de la funcion
    if destport = 200 then ' El puerto donde vamos a escuchar
        'captamos los bytes entrantes del buffer
        getRequest=spi_ethernet_getbyte()
    .
    .
    .
    if getRequest="1" then
        'rutina para controlar el motor puerta residente
        'validar=1
        'RCREG=0
65 PORTC.0=0
    .
    .
    for i=0 to 3
        PORTD.0=1
    .
    .
    .

```

MESSAGES

Line/Column	Message No.	Message Text	Unit

500: 50 Modified Insert C:\Documents and Settings\Marta de SanMart

mikroBasic compiler for PIC - proyecto.php

File Edit View Project Debugger Run Tools Help

Code Explorer Help Keyboard

projecto.pbas | proyecto.asm

```

    . while(Button(PORTB, 2, 1, 1))
    . PORTC.0=1
518 wend
    . txt = "0" 'botón visitantes
517 spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt, StrLen(txt)
    . PORTC.0=0
    . end if
520 if(Button(PORTB, 3, 1, 1)) then |
    . while(Button(PORTB, 3, 1, 1)) wend
    . txt = "3" 'botón apagado
    . spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt, StrLen
    . PORTC.0=0
525 end if
    . if(Button(PORTB, 4, 1, 1)) then
    . while(Button(PORTB, 4, 1, 1)) wend
    . txt = "4" 'botón salida
    .
    .
    .

```

MESSAGES

Line/Column	Message No.	Message Text	Unit

520: 47 Modified Insert C:\Documents and Settings\Marta de SanMart

F) Manual de usuario del programa de consulta de reportes diarios

Reporte de vehículos que ingresan o salen

A continuación se explicará como se configura el reporte para poder acceder a información almacenada en la base de datos, puede ser local o remoto.

Configuración Inicial

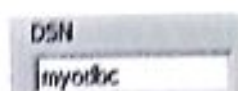
Los pasos a seguir para la configuración son los siguientes:

1. Crear un DSN para poder conectarse a una base de datos.

DSN. Especifica el nombre del DSN creado en las herramientas administrativas de Windows para lograr la conectividad entre nuestro programa y la base de datos. El formato es: DSN= "nombre del DSN".

Para ver los DSN que se tienen en el sistema se va a Panel de control -> Herramientas Administrativas -> Orígenes de Datos ODBC -> DSN de Sistema. Escogemos el DSN de MySQL para guardar datos en dicha base de datos.

2. Una vez creado el DSN estamos listos para poder configurar la aplicación, entramos al programa, y en él identificaremos una opción para editar el nombre del DSN, ponemos el nombre del DNS creado en el paso anterior.



En este ejemplo el DSN se llama myodbc

3. Adicionalmente configuraremos la ruta donde se encuentran las fotos que son almacenadas por la aplicación. Se configurará también la ruta por defecto, esta ruta por defecto es para ubicar a la foto que aparecerá cuando no haya fotos disponibles, se debe copiar una imagen con el nombre default en formato jpg en el directorio donde están las fotos.

A screenshot of a software interface showing a text input field labeled "Path". The field is empty and has a small icon on the right side, possibly representing a file explorer or search function.

Ruta donde están las fotos

Manejo del programa

Para utilizar la aplicación solo se debe ejecutarla y configurar la primera vez, luego de eso se podrá realizar consultas según la fecha que el usuario desee, la fecha podrá se escogida mediante el control calendario que estará disponible en el panel de consultas. Una vez seleccionada la fecha, se podrá presionar en el botón consultar y automáticamente saldrán las consultas correspondientes a la fecha escogida, para poder ver la fotografía, con el mouse el usuario podrá dar un clic sobre la fila de la información y automáticamente se visualizará la fotografía correspondiente.

The screenshot shows a software application window titled "Reporte de Ingresos y Salidas de Vehiculos". The window has a menu bar with "File", "Edit", "View", "Project", "Operate", "Tools", "Window", and "Help". Below the menu bar is a toolbar with various icons. The main content area is divided into several sections:

- Top Section:** Contains a "Selecciono el tipo" dropdown menu with "valido" selected and a "Consultar" button. Below it is a "Selecciono la fecha" dropdown menu showing "Lunes 18 de Mayo de 2009" and a "Fecha consultada" field with the value "2009-05-18".
- DSN and Path:** A "DSN" field with the value "myodbc" and a "Path" field with the value "C:\programa\Fotos".
- Form Fields:** A "status" field with a dropdown arrow, a "code" field with the value "10", and a "source" field.
- Information Table:** A table with the following columns: "ID", "FECHA", "HORA", and "CEDULA". The table is currently empty.
- Fotografía:** A large empty rectangular area for displaying a photograph.
- Bottom Section:** A "Número de foto" field with the value "1" and an "Archivo" field. A "Salir" button is located in the bottom right corner.

BIBLIOGRAFIA

1. Autor: Wikipedia (Varios autores)
Título: Radiación infrarroja
Fecha: 27 de mayo 2009
<http://es.wikipedia.org/wiki/Infrarrojo>

2. Autor: Wikipedia (Varios autores)
Título: Sensores Infrarrojos
Fecha: 22 de mayo 2009
http://es.wikipedia.org/wiki/Sensores_infrarrojos

3. Autor: Escuela Politécnica Superior de Alcoy, proyecto LSED.
Título: SECUENCIAS PARA MANEJAR MOTORES PASO A PASO
(UNIPOLAR)
Año: 2002 – 2003
<http://server-die.alc.upv.es/asignaturas/LSED/2002-03/MotoresPasoPaso/ftomotpap.htm>

4. Autor: Wikipedia (Varios autores)
Título: Motor Eléctrico
Fecha: 29 de Mayo del 2009
http://es.wikipedia.org/wiki/Motor_el%C3%A9ctrico

5. Autor: Varios Autores

Título: Manual de MySql

Fecha: 1 de Agosto del 2007

<http://www.programatium.com/tutoriales/cursos/mysql/index.htm>

6. Autor: Llamazares Juan Carlos

Título: ¿Cómo funciona?: Tarjetas identificadoras sin contacto o sistemas RFID

<http://www.ecojoven.com/dos/03/RFID.html>

7. Autor: Urueña León Edsel Enrique

Título: Direccionamiento IP

Año: 2005

<http://www.monografias.com/trabajos29/direccionamiento-ip/direccionamiento-ip.shtml>

8. Autor: National Instruments

Título: NI Vision for LabVIEWTM User Manual

Fecha: Noviembre 2005

<http://www.ni.com/pdf/manuals/371007b.pdf>

9. Autor: Sun Microsystems

Título: Configuring a Connector/ODBC DSN on Windows

Fecha: 5 de Junio del 2009

<http://dev.mysql.com/doc/refman/5.0/en/connector-odbc-configuration-dsn-windows.html>

10. Autor: Navaismo

Título: M2Eth-Board: conecta tu PIC a Ethernet

Fecha: 15 de septiembre de 2008

http://www.micropic.es/index.php?option=com_content&task=view&id=79&Itemid=1

11. Autor: Universidad Antonio Nariño

Título: Comunicación Serial

Fecha: 21 de Junio del 2008

http://www.geocities.com/micros_uan/cap47.html

12. Autor: mikroElektronika

Título: MikroBASIC User's Manual

http://www.mikroe.com/pdf/mikrobasic/mikrobasic_manual.pdf