

El capítulo 3 detalla las consideraciones para el diseño del sistema tanto para el software como para el hardware.

El capítulo 4 se explica la implementación en hardware y las pruebas del diseño.

Luego encontramos la recopilación de conclusiones y recomendaciones que obtuvimos a través de la realización de este proyecto.

Al final se dan los anexos, con un manual para la correcta operación y puesta en marcha del sistema y también se describe detalladamente el programa de control que se utilizó en el PIC para la tarjeta de control, medición y comunicación del sistema.

CAPÍTULO I

1. Generalidades

En el presente capítulo se detallan los antecedentes, la descripción general del sistema y la solución al problema planteado así como también el alcance y limitaciones que tiene este proyecto.

1.1. Antecedentes.

A medida que la electrónica y las comunicaciones se van desarrollando nos brinda nuevos sistemas de seguridad, más sofisticados y a menor costo pudiendo controlar desde lugares remotos.

En la actualidad existen una gran variedad de mecanismos y dispositivos utilizados como sensores de temperatura y presencia con salida

analógica o digital, junto con controladores dedicados a tareas específicas.

Hoy en día los programadores proveen al mercado softwares mucho más completos y con mejores aplicaciones, por ello para el desarrollo de proyectos se cuenta con herramientas que nos permiten hacer lo que proponemos.

1.2. Descripción General del Sistema.

Desarrollaremos un Sistema de Seguridad Industrial, para lo cual asumimos que una industria está dividida en diferentes sectores o zonas.

Por lo tanto nuestro proyecto consiste en monitorear la temperatura en el ambiente y el acceso de las personas en cada sector para garantizar la seguridad en la industria, todo esto por medio de cámaras y sensores además del encendido de alarmas y registro de sucesos a través del uso del microcontroladores y programas que se detallarán más adelante.

Con este sistema la persona encargada de monitorear remotamente lo que está sucediendo en la industria tendrá la certeza de que los datos que está visualizando por medio de la aplicación creada son reales.

Dado que nuestro proyecto consiste en monitoreo y registro de sucesos en una base de datos, entonces la persona encargada de monitorear será la responsable de tomar la decisión de lo que debe hacer según los datos que esté visualizando.

1.3. Planteamiento de la Solución

Se va sacar provecho de las prestaciones que ofrece el microcontrolador, es decir que con una programación en su memoria y dispositivos conectados a él sea capaz de controlar la seguridad en una industria.

Teniendo la necesidad de hacer un registro para cuando ocurra un suceso con los sensores usados, vamos a aprovechar las características que nos presenta LabView con su librería IMAQ-VISION y hacer la

captura de imágenes como también la librería Database Connectivity para conectarnos a una base de Datos para ir guardando todo.

Una aplicación muy interesante es la de tener acceso por la red al sistema implementado y para poder hacer esto es necesario usar una tarjeta de interfaz

entre nuestro controlador del sistema y la red.

Planteamos el monitoreo a través de la red por medio de cámaras IP a cada uno de los sectores de la industria, dicho monitoreo se lo podrá hacer remotamente de donde sea y cuando sea.

En la variación de temperatura tendremos las siguientes etapas: Visualización de temperatura, encendido de alarmas y registro en Base de Datos. El sensor de temperatura adquiere el nivel de temperatura dentro del área y se la visualiza en el panel de control en LABVIEW, después se realiza la comparación con una temperatura preestablecida y si no está dentro de dicho rango se enciende una alarma y se registra en una base de datos el suceso.

Para el sensor de detección de proximidad de una persona, dentro de un área existirá una cámara encendida todo el tiempo grabando las diferentes actividades, al momento de detectar a alguien el sensor, entonces, ésta cámara adquiere una imagen, la envía a una base con los datos necesarios para un futuro control.

Se realizará un programa en MikroBasic para programar el microcontrolador así como también la comunicación con la tarjeta que servirá de interfaz con la red, una aplicación con LabVIEW 8.5 para visualizar los datos de los sensores, reconocer las cámaras, capturar imágenes, también usaremos la aplicación SyBase que permite manejar una base de datos donde vamos a almacenar sensor que actúa así como fecha y hora de sucesos además de guardar la imagen capturada.

1.4. Alcance del proyecto

El alcance de nuestro proyecto está enfocado bajo los parámetros de necesidad de seguridad industrial en las que hoy en día se desarrolla un correcto ambiente laboral en una industria y como una buena alternativa presentamos nuestro sistema que puede ser desarrollado e implementado en cualquier industria

Nuestro sistema tiene la capacidad de usarse bajo las distintas condiciones de trabajo en las que intervengan exceso de temperatura límite permitida en las diferentes áreas y poder monitorear el ingreso de personal en horas no permitidas.

El uso de la tarjeta de interfaz entre el microcontrolador y la red es de gran utilidad dado que desde cualquier PC que esté en red y que tenga todos los programas instalados se puede acceder y realizar el monitoreo.

Poder visualizar cada sector de la industria a cualquier hora de trabajo mediante el uso de una cámara IP sin necesidad de estar físicamente en

la empresa, sino simplemente monitorear los eventos ocurridos desde cualquier lugar en el que nos encontremos.

1.5. Limitaciones del proyecto

El prototipo realizado es con fines didácticos, en caso de querer aplicarlo comercialmente habrá que realizar un trabajo de desarrollo adicional.

Para la realización del Sistema de Seguridad Industrial planteado podemos encontrarnos con que la tarjeta de interfaz Ethernet ENC28J60 así como también el microcontrolador 18F4520 no se encuentren disponibles en el mercado, para lo cual la solución sería importar dichos dispositivos lo cual implica tiempo y costo de envío y por ende esto causaría retraso en la implementación del sistema.

El sistema necesita de una persona que esté monitoreando para poder tomar la decisión de lo que debe hacer cuando su aplicación le está mostrando que ocurre un suceso en algún sector.

El sensor de temperatura que usamos para el prototipo es útil para ambientes que no sobrepasen los 150°, este sensor se debe cambiar dependiendo de lo que la industria desee implementar en el sistema. Es posible trabajar con temperaturas mucho más elevadas.

Con respecto al sensor de movimiento el limitante es que abarca un rango de 180° y a nivel industrial sería preferible tener un sensor que abarque los 360° porque estamos trabajando en lugares mucho más amplios.

CAPÍTULO 2

2. Base Teórica

A continuación se describen las características principales tanto de los dispositivos de hardware como del software que permiten la implementación de nuestro sistema.

2.1. Descripción de componentes del hardware del sistema

2.1.1.Sensores

Los sensores son los dispositivos que nos permite obtener información que nos sirva para procesar en el PIC y tomar decisiones de seguridad en la industria.

Características

- Precisión de $\sim 1,5^{\circ}\text{C}$ (peor caso), $0,5^{\circ}\text{C}$ garantizados a 25°C .
- No linealidad de $\sim 0,5^{\circ}\text{C}$ (peor caso).
- Baja corriente de alimentación ($60\mu\text{A}$).
- Amplio rango de funcionamiento (desde -55° a $+150^{\circ}\text{C}$).
- Bajo costo.
- Baja impedancia de salida.

2.1.1.2. Sensor de Movimiento LX16C.



Figura.2.2 Sensor LX16C

Características

- Tecnología de rayos infrarrojos capaz de detectar cualquier movimiento de su rango.
- Amplios rangos de detección.

- Sus rangos de detección abarcan zonas de arriba hacia abajo y zonas de izquierda a derecha.
- Identifica luz ambiental.
- Ahorro de energía eléctrica.
- Fácil instalación.
- Diseño moderno y estético.

Especificaciones

- Distancia de detección: 2m – 11 m
- Rango de detección: 180°
- Luz ambiental operación: menor a 5 – 1000lux
- Tiempo de activación: 5 – 480 segundos
- Frecuencia: 50 -> 60Hz
- Velocidad de detección: 0.6 – 1.5ms
- Voltaje de operación: 90 -> 130VCA

2.1.2.Fuente de Alimentación.

El Suministro de energía se toma de una fuente de poder para obtener voltajes de 5.0 V, 12.0 V, -12.0 V para alimentar el sistema.

2.1.3.Alarma.

Usamos un buzzer como alarma donde este emite un sonido continuo cuando ocurre un suceso.



Figura.2.3 Buzzer

2.1.4.Tarjeta ENC28J60

El Microchip ENC28J60 es un controller Ethernet 10Base-T (10Mbps en cables) y está constituido por un módulo PHY (nivel físico), un módulo MAC (subnivel MAC), una memoria RAM de 8kbyte para almacenar los

paquetes en recepción y en transmisión, una serie de registros de configuración y un módulo para la comunicación serial SPI.

El chip tiene solo 28 pines y requiere pocos componentes externos, por eso puede ser insertado muy sencillamente en cualquier proyecto.

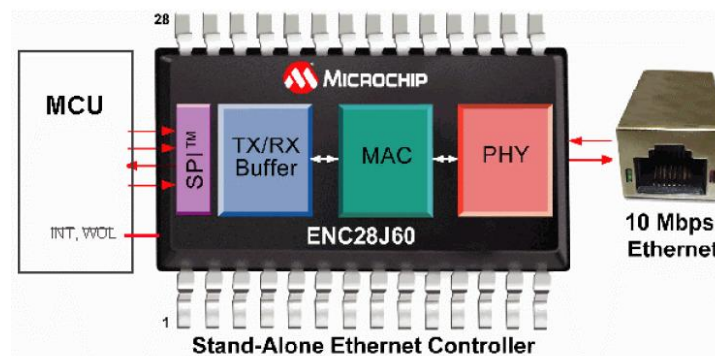


Figura.2.4. Módulo ENC28J60

Está diseñado específicamente para permitir la comunicación entre un microcontrolador y una red Ethernet. Este circuito integrado es un controlador Ethernet que soporta el estándar de comunicación IEEE 802.3.

El módulo utiliza un conector de tipo PRJ-005, que permite la conexión a una red Ethernet utilizando un Jack RJ-45. El chip ENC28J60, que es el que permite la comunicación Ethernet utilizando el estándar SPI como

interfaz de comunicación serial entre el PIC 18F4520 y el chip ENC28J60. Además, se puede observar la presencia de tres conectores denominados HD1, HD2 y HD3. Los conectores HD1 y HD2 le permiten al módulo Ethernet comunicarse con el sistema, mientras que el conector HD3 le permite conectarse a otros microcontroladores. La función que cumplen cada uno de estos pines se detalla en la tabla presentada a continuación.

PIN	TIPO	FUNCION
CS	INPUT	Habilita o deshabilita el bus SPI del ENC28J60. CS= 0 habilita el bus SPI. CS= 1 deshabilita el bus SPI.
SDO	OUTPUT	Señal serial de salida de datos.
SDI	INPUT	Señal serial de entrada de datos.
SCK	INPUT	Señal de reloj.
INT	OUTPUT	Señal de interrupción.
RST	INPUT	Señal de reset.
WOL	OUTPUT	Señal Wake-up un LAN interrupt.
CLKO	OUTPUT	Señal de reloj programable.

Tabla 2.1 Conectores del módulo ENC28J60

Esta tarjeta consta de un jumper (JP1) que funciona de la siguiente manera:

Si el módulo ET-MINI ENC28J60 se conecta a un microcontrolador que está alimentado por una fuente de poder externa de +3.3V, el puente deberá colocarse entre los punto 2 y 3, es decir, entre VDD y +3.3V. Esto permitirá que el chip ENC28J60 se conecte directamente a la fuente de +3.3V, deshabilitando de esta manera el circuito regulador de voltaje (LM3940). Si el módulo de comunicación Ethernet se conecta a un microcontrolador alimentado por una fuente externa de +5V, entonces el jumper deberá colocarse entre los puntos 1 y 2, es decir, entre +5V y VDD. De esta manera se habilitará el circuito regulador de voltaje, proporcionándole al chip ENC28J60 una alimentación de +3.3V, que es el voltaje de operación de este integrado.

2.1.5.PIC18F4520.

Es el encargado de controlar los periféricos y procesar la información de los parámetros de interés de la industria. En su interior deberá contener diversos módulos que le permitan la interacción y control con los diferentes componentes del sistema.

El microcontrolador es donde se aloja las instrucciones de control que resuelve gran parte el problema planteado.

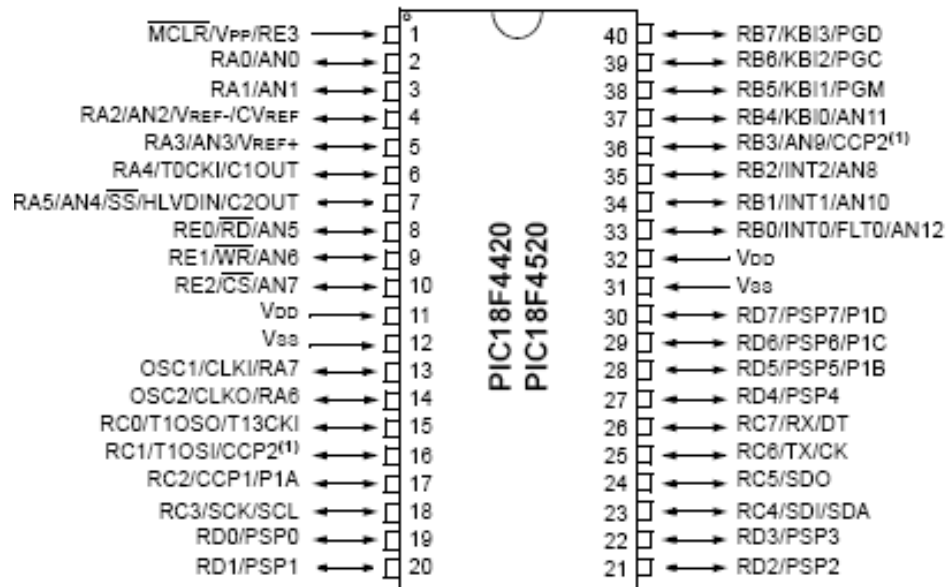


Figura.2.5 Circuito Integrado PIC18F4520

Características

La característica más importante del proyecto es la utilización del circuito integrado PIC18F4520 del fabricante Microchip Technology Inc.

- Comunicación por SPI
- Cristal de 20MHz (Osc HS)

- Dirección Física MAC
- Dirección IP
- Iniciar el SPI
- Programa particular

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		EUSART	Comp.	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I ² C™			
PIC18F2420	16K	8192	768	256	25	10	2/0	Y	Y	1	2	1/3
PIC18F2520	32K	16384	1536	256	25	10	2/0	Y	Y	1	2	1/3
PIC18F4420	16K	8192	768	256	36	13	1/1	Y	Y	1	2	1/3
PIC18F4520	32K	16384	1536	256	36	13	1/1	Y	Y	1	2	1/3

Tabla 2.2 Características del PIC18F4520

Funciones en el sistema

Entre las tareas que realiza el PIC18F4520 para el sistema son:

- Controlar por medio de sensores de temperatura diferentes zonas de una industria y al detectarse señales fuera de un rango considerado como normal se envían a encender alarmas. Las alarmas se visualizan desde LabView donde se muestra dicho suceso gráficamente y se lo almacena en una base de datos.

- Controlar la información que envían sensores de movimiento en ciertas horas, encender una alarma en caso de que se detecte a alguien y enviar la información a LabView para que la Cámara tome fotos y se envíe a la base de datos.
- Establecer una comunicación Ethernet de datos por medio del módulo de Comunicación ENC28J60.

2.1.6.Cámara IP.

Descripción Física

El modelo de la cámara que estamos usando es la IPCam SECURE300R



Figura.2.6 Cámara IP

Consta de:

- Led de Encendido
- Led de Conexión
- Led IR
- Lente
- Conector de cable de red
- Conector de alimentación de CC
- Botón restablecer
- Agujero para el tornillo de conexión al trípode

Descripción de la administración del sistema – imagen

El menú Imagen de administración del sistema contiene mandatos que proporcionan los valores de las imágenes tomadas por la IPCam SECURE300R.

- Resolución de Video
- Tasa de Compresión
- Flujo de Video
- Control de Brillo
- Control de Contraste
- Control de Saturación

- Frecuencia de Luz

2.1.7. Interfaz Comunicación Ethernet.

Para la interfaz de comunicación Ethernet usamos un cable de red cruzado con conector RJ45.

2.1.8. Equipo Terminal de información.

Es aquel donde se podrá recibir y almacenar la información del sistema con sus parámetros de interés para el técnico de la industria.

La información recibida será almacenada y podrá ser utilizada para estudiar el comportamiento del sistema.

Además contará con un programa de aplicación gráfica para visualizar los diferentes parámetros del sistema en el tiempo que envía el PIC por el puerto de comunicación.

2.2. Descripción de los programas usados en el Sistema

Para el desarrollo del proyecto fue necesario hacer uso de varios softwares sobre los cuales en esta sección se dará una breve descripción de las características más importantes.

2.2.1.MikroBasic

Uno de los compiladores de lenguaje BASIC para microcontroladores PIC que está atrayendo legiones de usuarios es MikroBasic. Si bien ya lleva un tiempo en el mercado, a partir de la versión 5.00 se ha masificado.

La programación del PIC18F4520 fue desarrollada con Mikrobasic V7.0.0.2 y la compilación de dicho programa genera un archivo hexadecimal que luego será enviada al PIC18F4520 por medio de un quemador de PIC.

2.2.1.1. ADC

Es un convertidor analógico a digital de 10 bits de resolución de nueve canales según la selección de usuario que permite la conversión de entrada de señal analógica.

Adc_Read

Prototype sub function Adc_Read(dim channel as byte) as word

Returns 10-bit unsigned value read from the specified channel.

Description Initializes PIC's internal ADC module to work with RC clock. Clock determines the time period necessary for performing AD conversion (min 12TAD).

Parameter channel represents the channel from which the analog value is to be acquired. Refer to the appropriate datasheet for channel-to-pin mapping.

Requires Nothing.

Example `tmp = Adc_Read(1)` ' *Read analog value from channel 1*

Lo usaremos para convertir las señales analógicas que nos entregan los sensores de temperatura y movimiento en digitales para poderlas enviar por medio del protocolo UDP a la red y adquirirlas, visualizarlas y procesarlas en LabView.

2.2.1.2. Librería SPI Ethernet

Ésta librería nos ahorra mucho trabajo porque sabe donde están los registros del ENC28J60 lo cual permite usar las subrutinas de manera simple. Ésta librería nos sirve para PIC de más de 4KB de ROM y con hardware SPI integrado por lo que no todos los PICs son compatibles. Para el uso de PICs de la familia 16F existen algunas limitaciones es por ello que se debe usar un PIC de la familia 18F.

2.2.1.3. Protocolo de Transporte UDP

Es un protocolo mínimo de nivel de transporte orientado a mensajes. En la familia de protocolos de Internet UDP proporciona una sencilla interfaz entre la capa de red y la capa de aplicación.

UDP añade la información necesaria para la comunicación extremo a extremo al paquete que envía al nivel inferior. Lo utilizan aplicaciones como NFS (*Network File System*) y RCP (comando para copiar ficheros entre ordenadores remotos), pero sobre todo se emplea en

tareas de control y en la transmisión de audio y vídeo a través de una red. No introduce retardos para establecer una conexión, no mantiene estado de conexión alguno y no realiza seguimiento de estos parámetros.

2.2.3.1. Puertos que usa

UDP utiliza puertos para permitir la comunicación entre aplicaciones. El campo de puerto tiene una longitud de 16 bits, por lo que el rango de valores válidos va de 0 a 65.535. El puerto 0 está reservado, pero es un valor permitido como puerto origen si el proceso emisor no espera recibir mensajes como respuesta.

Los puertos 1 a 1023 se llaman puertos "bien conocidos" y en sistemas operativos tipo Unix enlazar con uno de estos puertos requiere acceso como superusuario.

Los puertos 1024 a 49.151 son puertos registrados.

Los puertos 49.152 a 65.535 son puertos efímeros y son utilizados como puertos temporales, sobre todo por los clientes al comunicarse con los servidores.

2.2.2 LabView

2.2.2.1 Descripción

LabView es una herramienta gráfica para pruebas, control y diseño mediante la programación. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico.

Los programas desarrollados con LabView se llaman Instrumentos Virtuales, o VIs, lo que da una idea de su uso en origen: el control de instrumentos. El lema de LabView es: *"La potencia está en el Software"*. Entre sus objetivos están el de reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a programadores no expertos. Esto no significa que la empresa haga únicamente software, sino que busca combinar este software con todo tipo de hardware, tanto propio -tarjetas de adquisición de datos, PAC, Visión, y otro Hardware- como de terceras empresas.

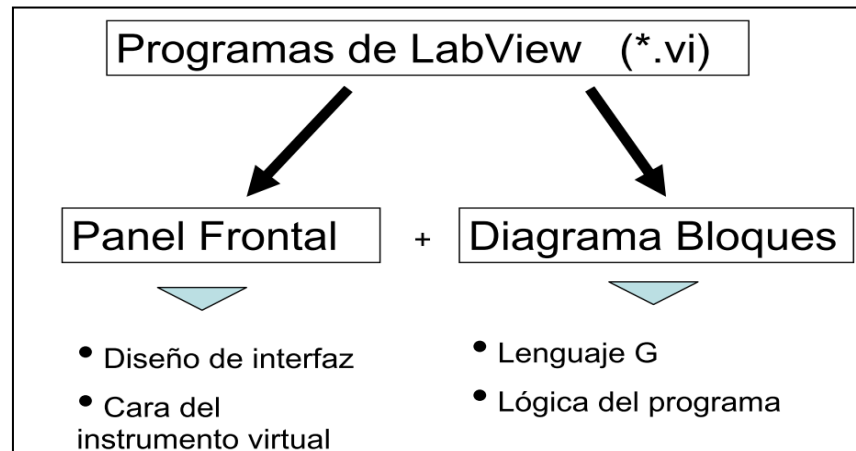


Figura.2.7 Diagrama General de LabView

2.2.2.2 Librería IMAQ-Vision

IMAQ USB Init.vi

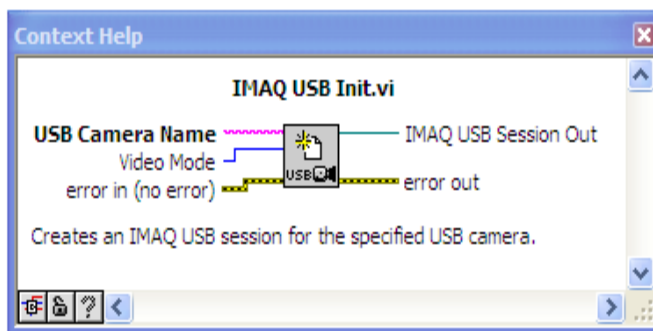


Figura2.8 Función IMAQ USB Init.

Esta función permite iniciar una sesión IMAQ USB en la cámara especificada en el parámetro USB Camera Name donde se especifica el nombre de la cámara a ser usada además cuenta con el parámetro Video Mode que es el modo de video requerido.

IMAQ USB Grab Setup.vi

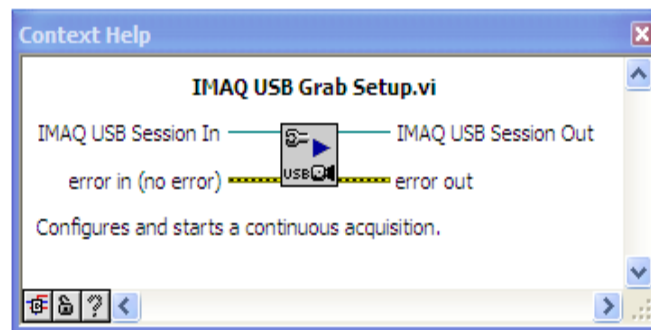


Figura2.9 Función IMAQ USB Grab Setup.

Este instrumento virtual presentado en la imagen inferior requiere básicamente de dos parámetros para operar. El primero es el denominado IMAQ USB Session In, que indica al instrumento virtual la sesión IMAQ USB desde la cual va a adquirir los datos de manera continua.

IMAQ

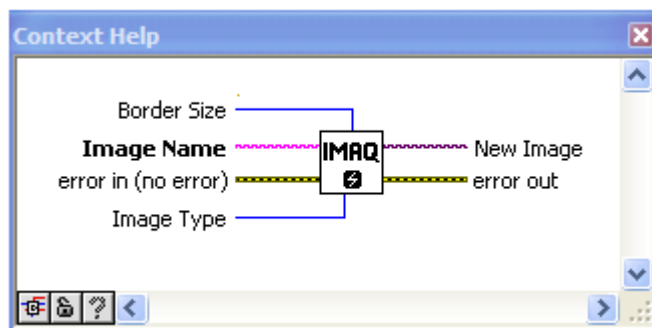


Figura2.10 Función IMAQ.

El parámetro Image Name se refiere a la sesión iniciada con la cámara Web especificada por el usuario en el panel frontal de LabVIEW.

El siguiente parámetro es el correspondiente al error de entrada que indica si ha existido algún error en el proceso previo a este instrumento virtual.

IMAQ USB Grab Acquire.vi

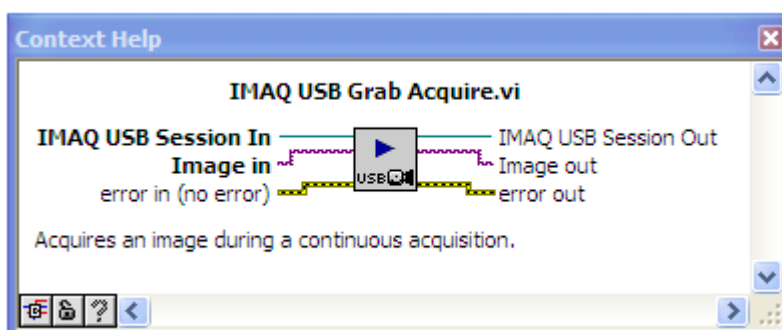


Figura2.11 Función IMAQ USB Grab Acquire.

IMAQ USB Grab Acquire que permite adquirir una imagen durante una adquisición continua, esta adquisición continua ha sido activada previamente y utilizando tres parámetros de entrada, estos parámetros son proporcionados por las funciones previamente implementadas. A la salida de esta función tenemos el error out, el IMAQ USB Session Out y el Image Out.

IMAQ USB Close.vi

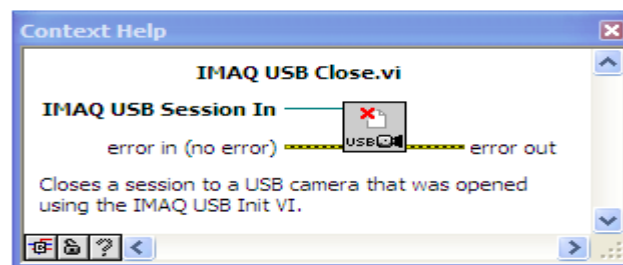


Figura2.12 Función IMAQ USB Close.

Utilizando la función IMAQ USB Close se procede a cerrar la sesión IMAQ USB mostrada en la figura.

2.2.2.3 Librería Database Connectivity

Es muy importante saber la conexión entre LABVIEW y SYBASE. Para ello necesitamos conocer lo íconos que son los que hacen posible la conexión entre ellos:

DB Tools Open Connection

Abre una conexión de base de datos utilizando la información de conexión.

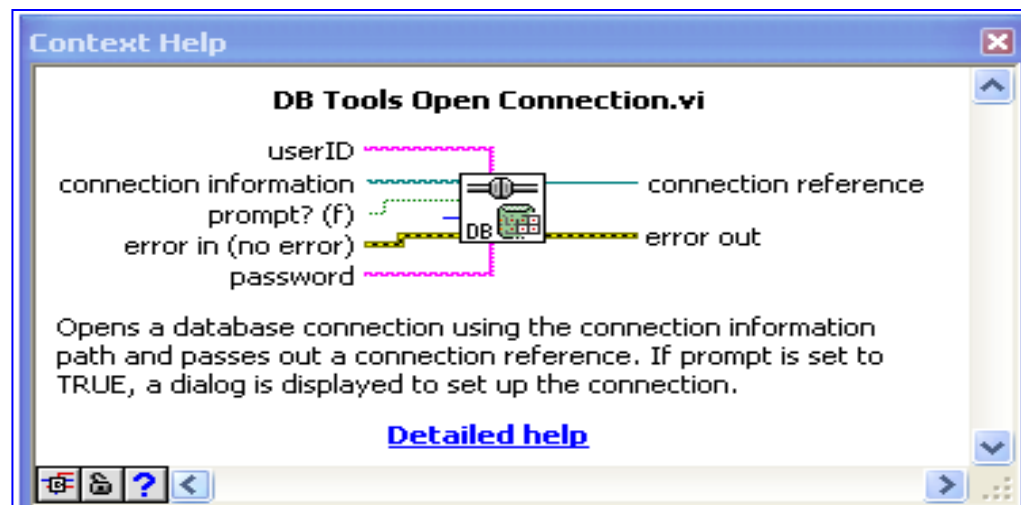


Figura2.13 Función DB Tools Open Connection.

DB Tools Close Connection

Cierra una conexión de base de datos mediante la destrucción de sus asociados respecto de referencia.

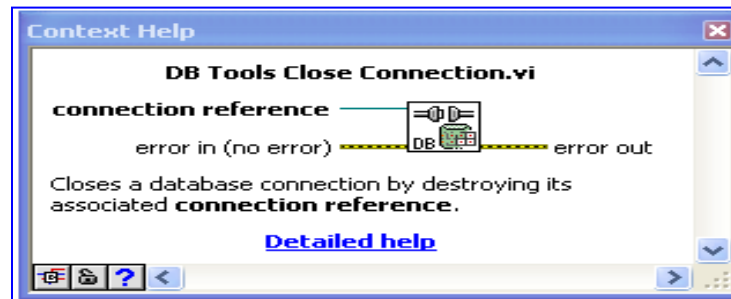


Figura2.14 Función DB Tools Close Connection.

DB Tools Insert Data

Inserta una nueva fila en la tabla de la base de datos identificada por la conexión de referencia.

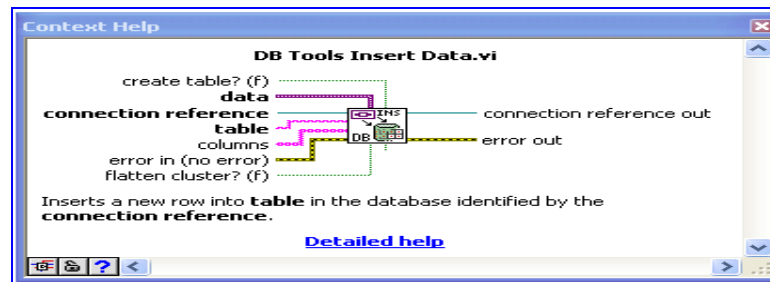


Figura2.15 Función DB Tools Insert Data.

DB Tools Select Data

Selecciona los datos de la tabla en la base de datos definidas por el marco de referencia utilizando las columnas ofrecidas en las columnas del arreglo.

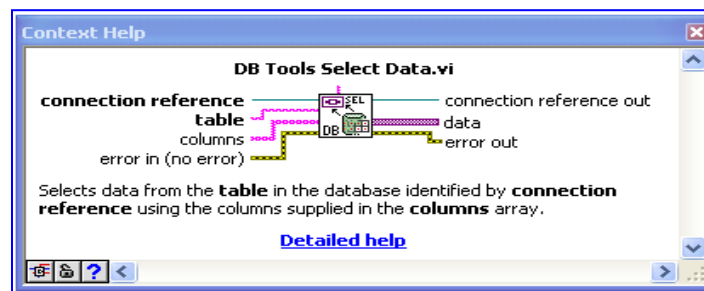


Figura2.16 Función DB Tools Select Data.

DB Tools Create Table

Crea una nueva tabla en la base de datos definidas por el marco de referencia. La tabla y la columna de información insumos describir el nombre de la tabla y las propiedades de cada columna en la tabla, respectivamente

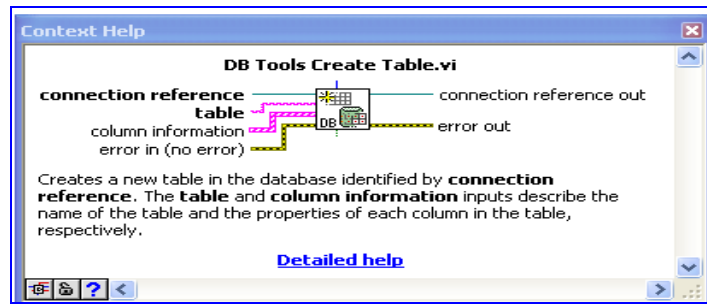


Figura2.17 Función DB Tools Create Tabla.

DB Tools Execute Query

Ejecuta una consulta SQL y muestra un conjunto de registros de referencia que deben ser liberados.

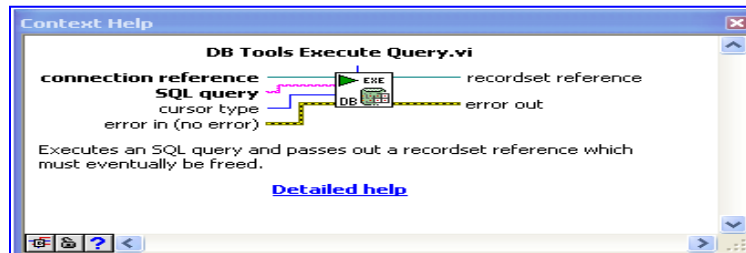


Figura2.18 Función DB Tools execute Query.

DB Tools Fetch Recordset Data

Obtiene los datos de los registros identificados por referencia. Los datos se devuelven como una matriz 2-D de variantes. Cada elemento de la

matriz se puede convertir a su nativa LabVIEW utilizando el tipo de base de datos de la variante A de Datos función.



Figura2.19 Función DB Tools Fetch Recordset.

DB Tools Fetch Element Data

Obtiene los datos ubicados en la columna índice del registro actual en el conjunto de registros identificados por los registros de referencia. La columna de índice puede ser cero el índice de posición de la columna en el conjunto de registros o el nombre de la columna. El tipo de entrada es polimórfico y determina el tipo de datos que se devuelve.

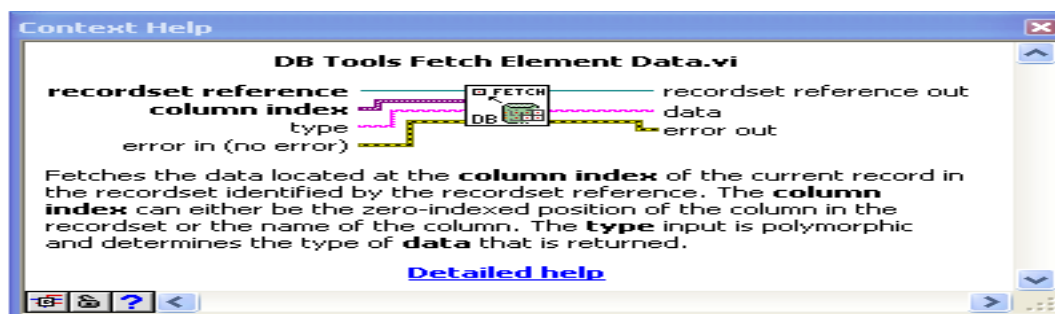


Figura2.20 Función DB Tools Fetch Element Data.

DB Tools Free Object

Libera un objeto mediante la destrucción de sus asociados y de referencia pasa a cabo una referencia diferente. ¿Qué objeto se destruye y se pasa de referencia es determinado por la identidad de la referencia aprobado en el VI:



Figura2.21 Función DB Tools Free Object.

DB Tools Drop Table

Borra la tabla especificada en la base de datos definidas por el marco de referencia.

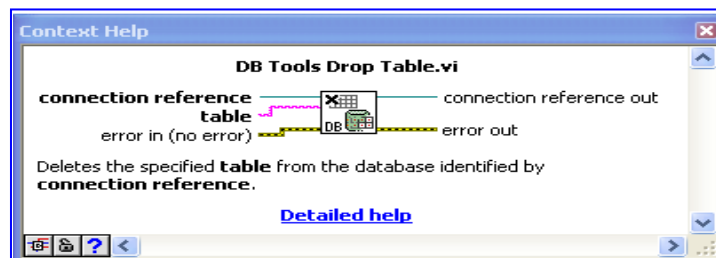


Figura2.22 Función DB Tools Drop Table.

2.2.2.4 Librería Data Communication

De esta librería usamos el protocolo UDP para la comunicación con el PIC.

UDP Open

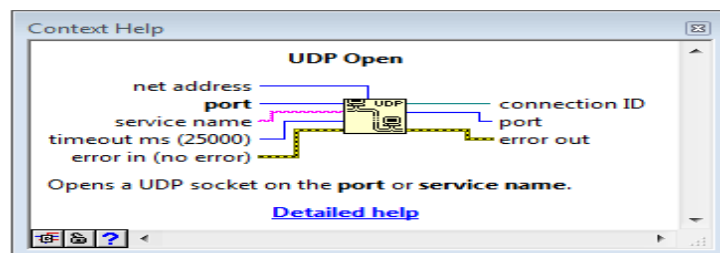


Figura2.23 Función UDP Open.

Esta función permite iniciar una sesión UDP en LabVIEW en el puerto especificado en el parámetro port. Este instrumento virtual trabaja con seis parámetros, de los cuales, tres son parámetros de entrada y tres son parámetros de salida.

UDP Read

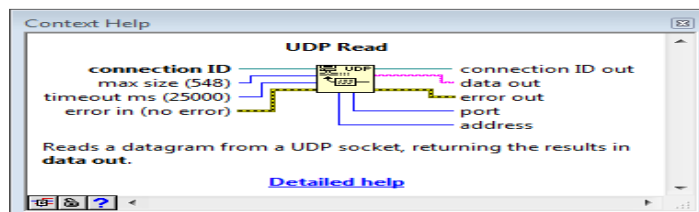


Figura2.24 Función UDP Read.

Esta función lee un datagrama desde un socket UDP, y retorna los resultados en el parámetro data out. Este instrumento virtual retorna datos si recibe cualquier byte y permanece en modo de espera únicamente si no está recibiendo ningún byte. Esta función está compuesta por nueve parámetros de los cuales seis son de entrada y tres de salida.

UDP Close

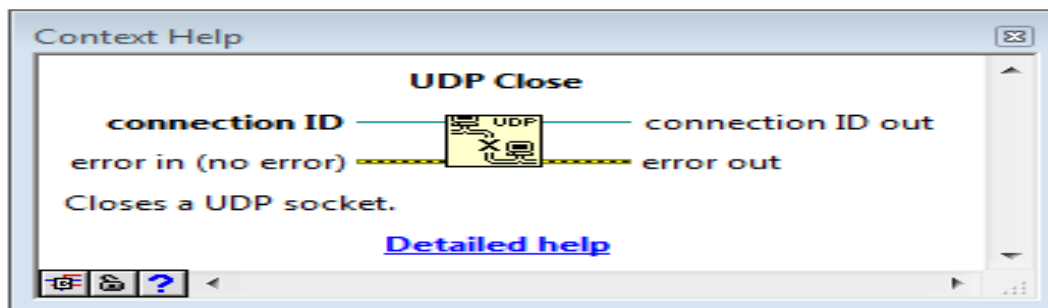


Figura2.25 Función UDP Close.

La función UDP Close permite cerrar una sesión UDP previamente inicializada. Esta función está compuesta por cuatro parámetros, dos de los cuales son parámetros de entrada y los dos restantes son los parámetros de salida del instrumento virtual.

CAPÍTULO 3

3. Diseño del Sistema de Seguridad Industrial

3.1. Diagrama de Bloques del Diseño

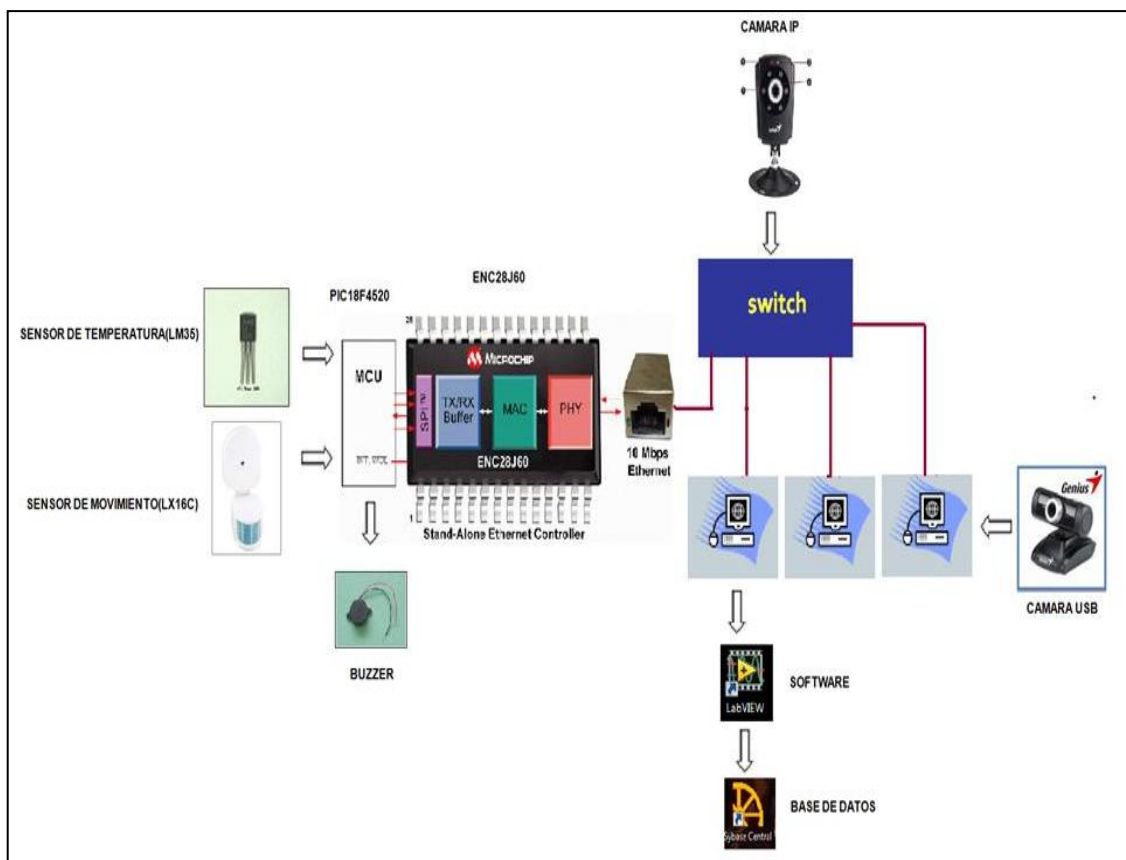


Figura.3.1 Diagrama de Bloques del Diseño

En la figura.3.1 se muestra de manera comprensible las partes que conforman el sistema que estamos desarrollando, las cuales son: bloque de sensores, bloque del PIC18F4520, bloque de interfaz de comunicación Ethernet, bloque del puerto de red y bloque de dispositivo terminal de información.

El bloque de sensores está conformado por el LM35 que es un sensor de temperatura y el LX16C que es un sensor de presencia cuyos datos serán monitoreados por el sistema.

Luego tenemos el bloque del PIC18F4520 el cual se encarga de las funciones de control del sistema con el programa MikroBasic digitalizando los datos análogos que le están entregando los sensores así como también configurar la comunicación con la tarjeta ENC28J60 por medio del protocolo UDP. Además este bloque tiene como salida el bloque de dispositivos de alerta que consta de alarmas que se encenderán cuando el nivel de temperatura sobrepase el límite programado en el PIC.

Los datos que ha procesado el PIC18F4520 son enviados al bloque de interfaz de comunicación Ethernet que está formado por el módulo ET-MINI ENC28J60 el cual permite la comunicación del sistema con la red.

La salida del módulo Ethernet es enviada al siguiente bloque que es un puerto de red donde cualquier PC que use el estándar Ethernet y esté en dicha red podrá obtener los datos que está enviando el PIC18F4520 además este bloque tiene como entrada los dispositivos de captura de imagen que son las cámaras que se deben conectar también a dicha red.

Por último tenemos el bloque de dispositivo terminal de información que es la PC con las aplicaciones de LabView y SyBase instaladas para el monitoreo.

3.2. Diseño del hardware

3.2.1. Acondicionador de señal para el sensor LM35

Dado que este sensor entrega diez milivoltios por cada grado centígrado y al trabajar con tensiones tan pequeñas cualquier ruido o interferencia puede hacernos tomar una lectura errónea, entonces es necesario hacer un adecuado tratamiento de la señal para obtener una precisión importante, pudiendo leerse fracciones de grado.

El voltaje de salida es de 10mV por grados, el rango de salida va desde 0V hasta 1,5V; es decir que antes de ingresar la señal al PIC18F4520 debe ser amplificada para que quede de 0V a 5V por medio del siguiente circuito:

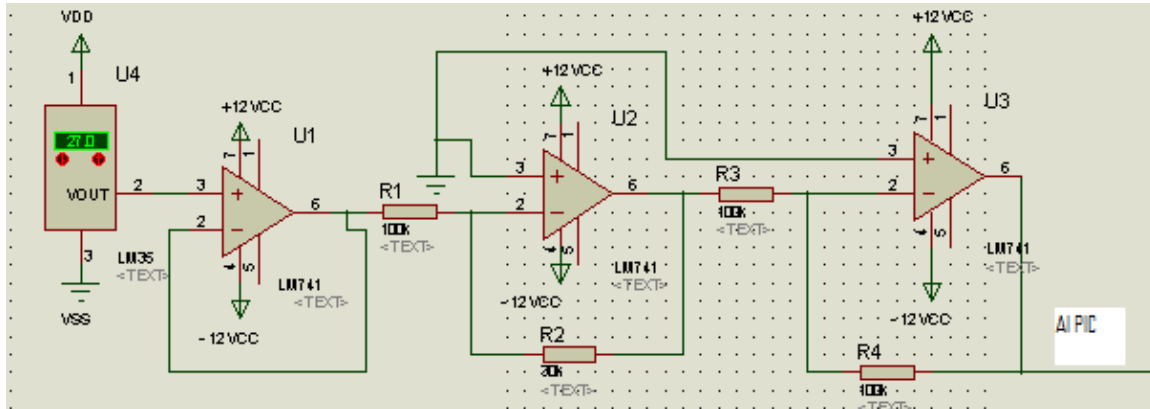


Figura3.2 Acondicionador de Señal del LM35

3.2.2. Etapa de Acoplamiento de la señal del sensor LX16C.

Dado que el sensor de movimiento LX16C trabaja con alimentación de 110VAC y nosotros necesitamos enviar esta señal al PIC18F4520 entonces para obtener esto aprovechamos que el sensor usado emite una señal de 22VDC a la salida del relay que tiene internamente cuando éste se activa.

Dicha señal se la envía al optoacoplador 4N25 que es el encargado de bajar la señal a 22VDC a 5VDC y poder enviar la señal del sensor de movimiento al PIC18F4520 por medio del siguiente circuito:

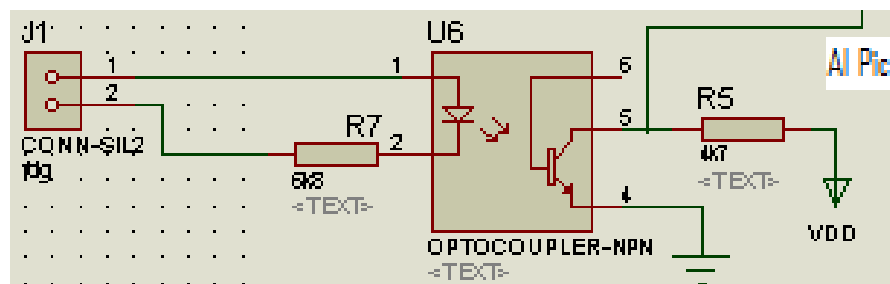


Figura3.3 Acondicionador de Señal del Sensor LX16C.

3.2.3. Etapa de verificación de conectividad Ethernet

En el puerto D del PIC están conectados unos leds que para saber que hay comunicación entre el PIC y la tarjeta ENC28J60, estos leds se encienden cuando se está enviando información desde el PIC hacia LabView vía Ethernet.

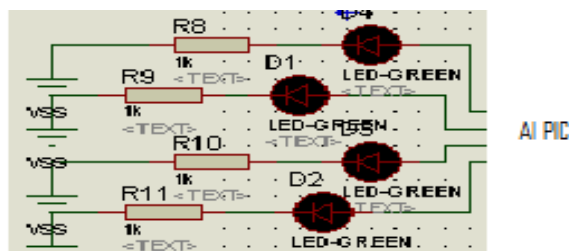


Figura3.4 Etapa de Verificación de Conectividad.

3.2.4. Pines Usados del PIC18F4520

Pines del PIC	Puerto	Periférico	Alimentación
1			Vpp
2	RA0	LM35 Sensor de Temperatura	
8	RE0	LX16C Sensor de Movimiento	
9	RE1	Buzzer	
10	RE2	Led indicador de alarma	
11			Vdd
12			Vss
13	RA7	Cristal 20MHz	
14	RA6		
15	RC0	A RST de Tarjeta Ethernet	
16	RC1	A CLKO de Tarjeta Ethernet	
18	RC3	A SCK de Tarjeta	

		Ethernet	
19	RD0	Led indicador de conectividad	
20	RD1	Led indicador de conectividad	
21	RD2	Led indicador de conectividad	
22	RD3	Led indicador de conectividad	
23	RC4	A SDI de Tarjeta Ethernet	
24	RC5	A SDO de Tarjeta Ethernet	
31			Vss
32			Vdd

Tabla.3.1 Puertos usados en el PIC18F4520

3.2.5. Diseño del sistema en PCB

Esta tarjeta fue elaborada en el programa ARES V7.4, es parte muy importante en el hardware ya que en ella se conectarán todos los elementos que usamos en nuestro sistema.

Aquí presentamos el diseño de las pistas para la comunicación entre el PIC y la interfaz Ethernet.

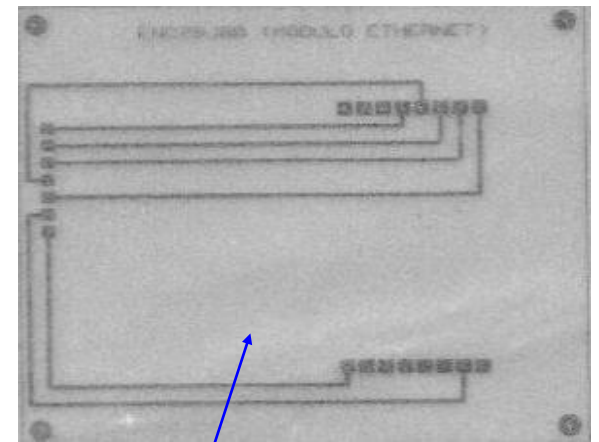
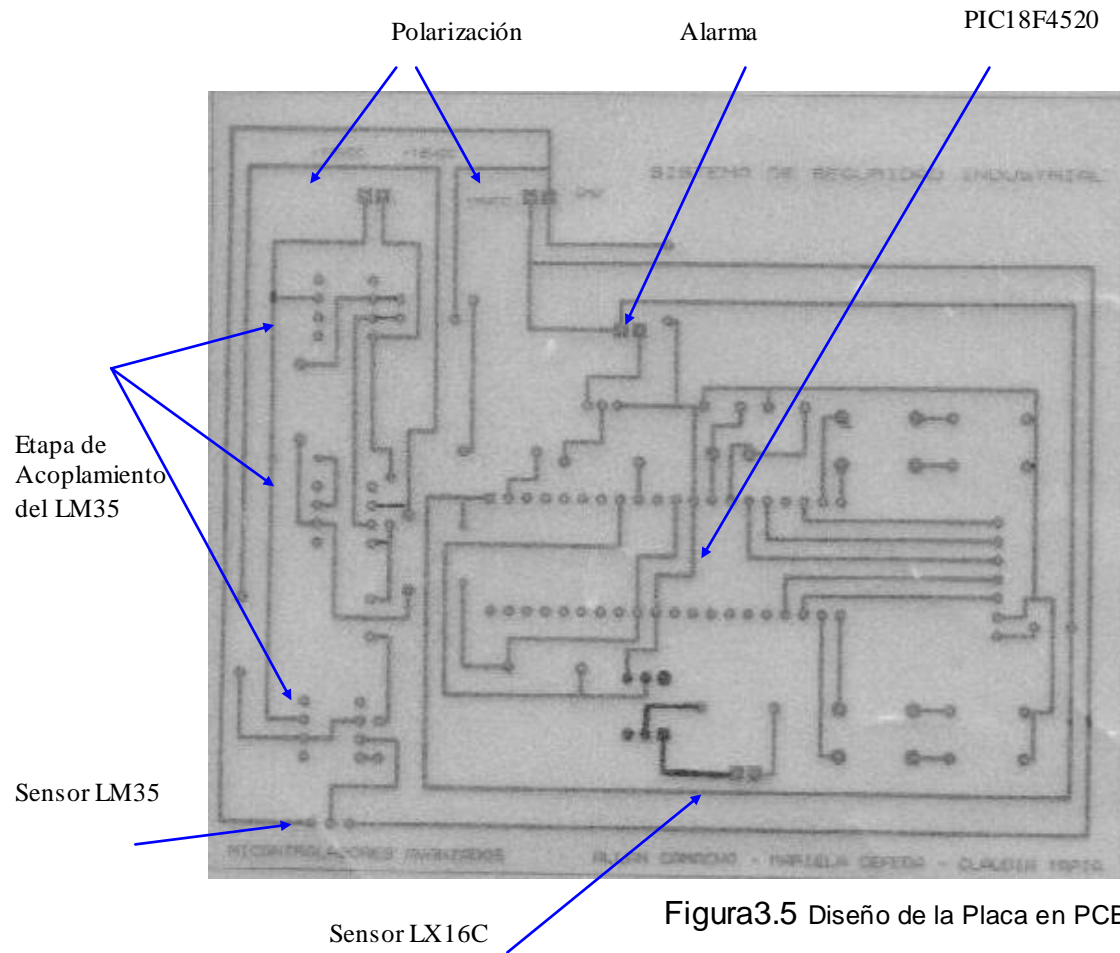


Figura3.5 Diseño de la Placa en PCB

3.3. Diseño del software del Sistema de Seguridad Industrial

3.3.1. Programación del PIC18F4520

El propósito de describir la programación en MikroBasic es dar a conocer cómo el PIC18F4520 es usado para implementar el control del Sistema de Seguridad Industrial.

Bloque1

```
program Sist_Seg_Ind
include "eth_enc28j60"
include "eth_enc28j60_api"

dim mymacaddr as byte [6]
dim myipaddr as byte [4]
dim getRequest as byte[20]
dim dyna as byte[30]
dim txt as string[100]
dim i as byte
dim IpAddr as byte[4]
dim dato as word
dim tmp as string[100]
```

Lo que primero observamos es el nombre con el que está grabado el programa luego encontramos el llamado a las librerías Ethernet que

permiten la comunicación utilizando este protocolo en conjunto con el chip ENC28J60. Lo demás es la declaración de las variables que vamos a utilizar para la programación de nuestro sistema en el PIC.

Bloque 2

```
sub function Spi_Ethernet_UserTCP(dim byref remoteHost as byte[4], dim remotePort, localPort, reqLength as word) as word
    result=0
end sub

sub function Spi_Ethernet_UserUDP(dim byref remoteHost as byte[4], dim remotePort, destPort, reqLength as word) as word
    result = 0
    if destport = 200 then
        for i=0 to 3
            getRequest[i]=spi_ethernet_getbyte()
        next i
    end if
end sub
```

Subrutinas que permiten habilitar la comunicación entre el microcontrolador y el módulo ET-MINI ENC28J60 gracias a la interfaz SPI que permite la transferencia de datos desde el PIC18F4520 hacia la tarjeta Ethernet, y el protocolo UDP que permite la transferencia de información hacia la PC.

Bloque 3

```

main:
  cmcon=7
  trisa=1
  adcon1=$0e
  adcon2=$A4
  trisd=0
  portd=0
  trise.0=1
  trise.1=0
  porte.1=0

```

Ya en el main del programa lo primero que se debe hacer es configurar en el PIC18F4520 los puertos deben habilitarse como entradas o salidas además de especificar también los puertos que van a ser usados como analógicos o digitales.

Bloque 4

```

' 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]=192
myipaddr[1]=168
myipaddr[2]=46
myipaddr[3]=210

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

' Direccion IP de la PC
IpAddr[0] = 192
IpAddr[1] = 168
IpAddr[2] = 46
IpAddr[3] = 203
delay_ms(3000)

```

En esta parte del programa estamos configurando las direcciones IP y MAC que van a ser usadas en el sistema de seguridad industrial, estas direcciones son de gran importancia de modo que tienen que ser bien asignadas para que el sistema funcione correctamente caso contrario no existirá conectividad en la red y no se podrá dar el monitoreo del sistema.

Bloque 5

```

while true
  dato = adc_read(0)
  delay_ms(200)
  wordtostr(dato,txt)
  PORTD = not PORTD
  tmp="temp@"
  Strcat(tmp,txt)
  txt=tmp
  spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt, Strlen(txt))
  delay_ms(200)
  tmp="alarma@"
  if(dato>273) then
    porte.1=1
    txt="1"
    Strcat(tmp,txt)
    txt=tmp
    spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt, Strlen(txt))
    delay_ms(200)
  else
    porte.1=0
    txt="0"
    Strcat(tmp,txt)
    txt=tmp
    spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt, Strlen(txt))
    delay_ms(200)
  end if
  tmp="movimiento@"

```

```
    if(porte.0=1) then
    txt="1"
    Strcat(tmp,txt)
    txt=tmp
    spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt, Strlen(txt))
    delay_ms(200)
    else
    txt="0"
    Strcat(tmp,txt)
    txt=tmp
    spi_ethernet_sendUDP(IpAddr, 10001, 4000, @txt, Strlen(txt))
    delay_ms(200)
    end if
    spi_ethernet_dopacket()
wend
end.
```

Por medio de las funciones presentes en el código de este bloque se está configurando la adquisición, encendido de alarma y envío de lo que están leyendo los sensores.

3.3.2.Programación de LabView

Teniendo implementada la parte de programación el hardware del Sistema de Seguridad Industrial descrita en el literal anterior es necesario usar un software que nos permita hacer una aplicación que permita visualizar los datos que se están enviando desde el hardware a la

persona encargada del monitoreo de la Industria y para ello usamos el software LabView que usa un lenguaje de programación gráfico.

En esta sección vamos a presentar y explicar brevemente el programa que hemos desarrollado.

Bloque de Adquisición de Datos

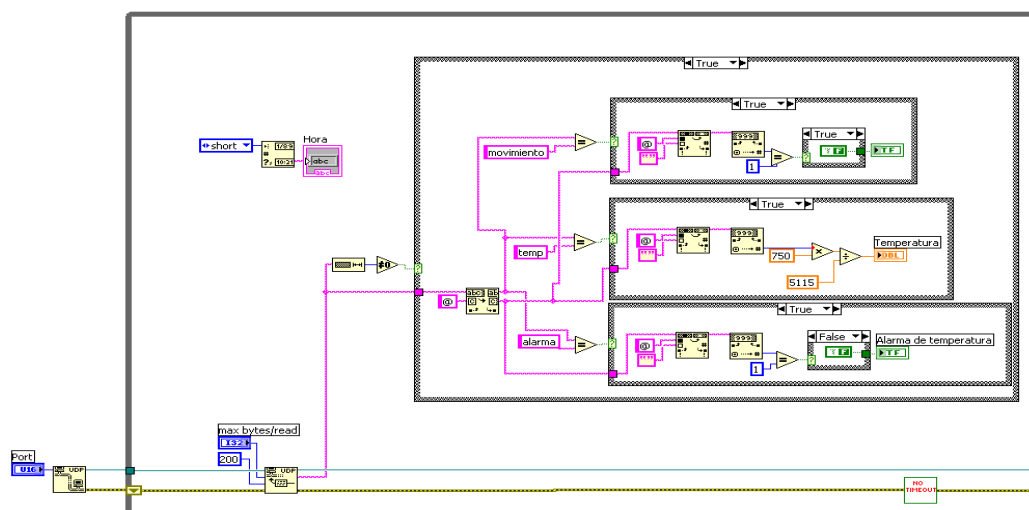


Figura3.6 Bloque de Adquisición de Datos.

Este bloque permite la comunicación Ethernet utilizando el estándar UDP para enviar los datos al PC del PIC.

Para iniciar una sesión UDP se usa como parámetro de entrada la variable de control denominada Puerto, que es un valor ingresado por el usuario a través de la interfaz presentada en el panel frontal de LabVIEW.

Una vez iniciada la sesión UDP se procede a la lectura de los datos utilizando la función UDP Read y a esta función se le ingresa el número máximo de bytes leídos y el retardo en milisegundos, que en este caso específico es de 200ms luego de esto usamos funciones de tipo string para escoger la información que necesitamos de la trama de datos que nos están llegando que puede ser información de la temperatura, movimiento o alarma y así poder ingresar a lazos case para tratar esta información y poder mostrarlas por medio de indicadores en el panel frontal y por último se procede a cerrar la sesión UDP utilizando la función UDP Close.

Bloque de Inicialización y Visualización de las Cámaras

Este bloque corresponde a la inicialización y visualización de las cámaras, como vamos a tener 2 cámaras para ello usamos un selector y

dependiendo de la cámara que se seleccione vamos a entrar a un lazo case para la configuración de cada cámara.

Para ello se inicializa una sesión IMAQ USB, utilizando los controladores USB PC Camera y Modo de Video que son valores especificados por el usuario a través del Front Panel de LabVIEW luego de esto se activa la adquisición de datos de manera continua, esto se logra utilizando la función IMAQ USB Grab Setup.

También encontramos la función IMAQ Create que permite crear o reservar un espacio de memoria temporalmente para almacenar una imagen adquirida usando una cámara Web USB el cual requiere únicamente el ingreso del parámetro Image Name, que indica el nombre que se va a asignar a la imagen almacenada en el espacio de memoria reservado temporalmente.

IMAQ USB Grab Acquire nos permite adquirir una imagen durante una adquisición continua y dicha imagen es presentada a través de un indicador en el panel frontal de LabVIEW.

Por último procedemos a cerrar la sesión IMAQ USB utilizando la función IMAQ USB Close.

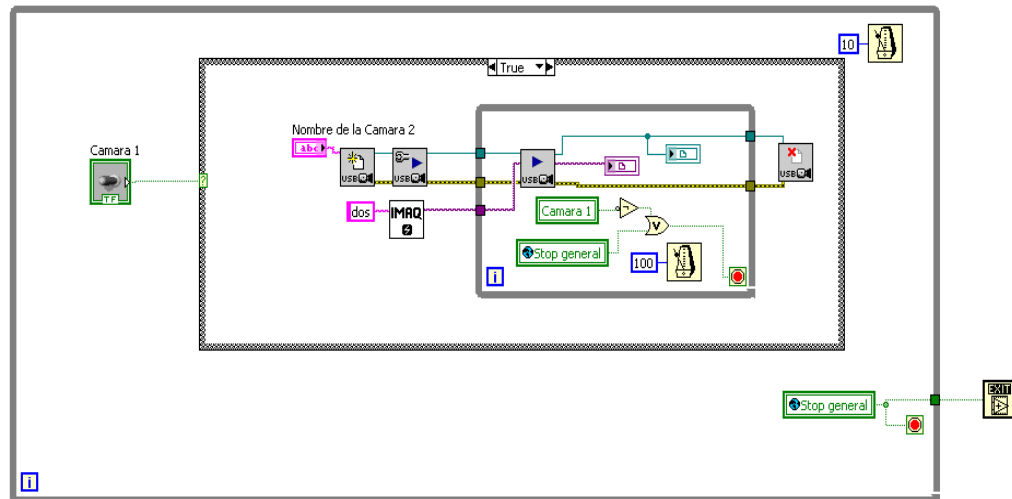


Figura3.7 Inicialización y Visualización de las Cámaras.

Bloque de Captura de Imágenes y Base de Datos

Con respecto a la base de datos creada en empezamos iniciando una conexión con la función DB Tools Open Connection luego se crean un cluster cuyos elementos son la fecha en la que el sistema recibió datos, la ruta de la imagen, cámara, ID y el número de imagen. Toda esta información se ingresa en el sistema de monitoreo manejado por SyBase

a través de la función DB Tools Insert Data. Esta función le indica al sistema la información que debe almacenar y donde hacerlo siempre y cuando la estructura case sea verdadera (TRUE) en caso contrario aunque ningún dato se esté enviando la conexión con SyBase se mantendrá activa.

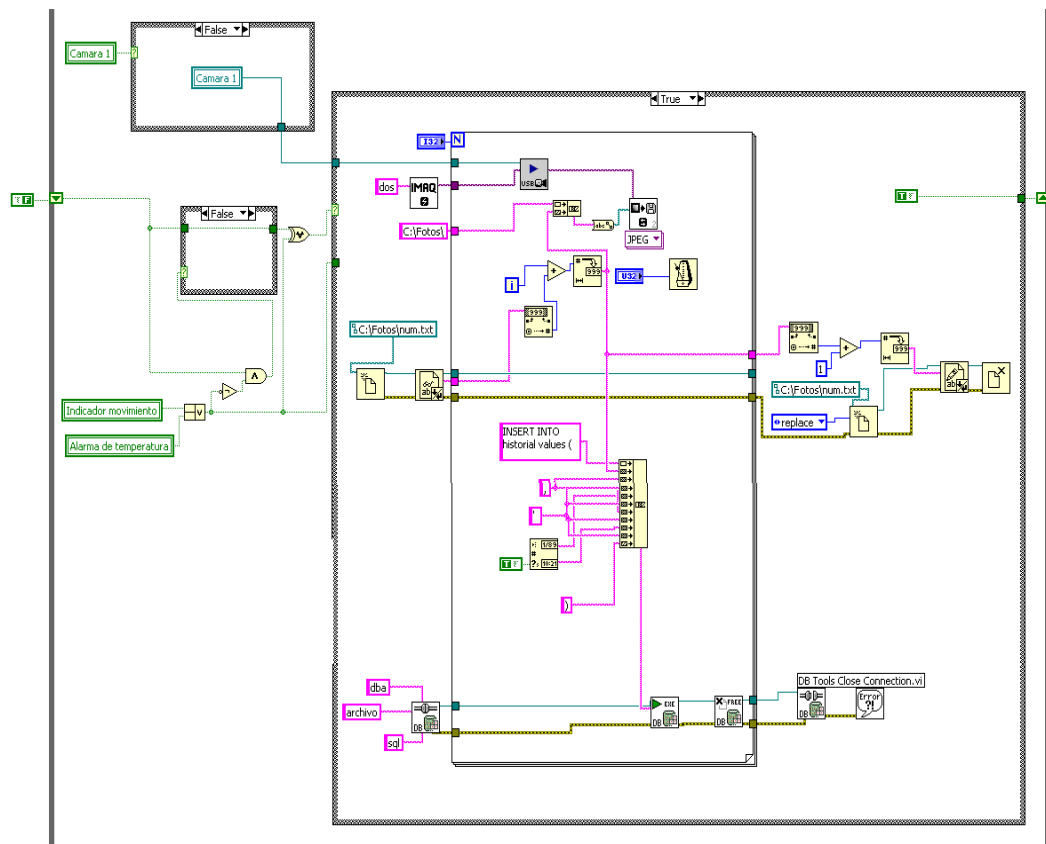


Figura3.8 Etapa de Captura de Imágenes y Base de Datos

Bloque del Panel Frontal del Sistema

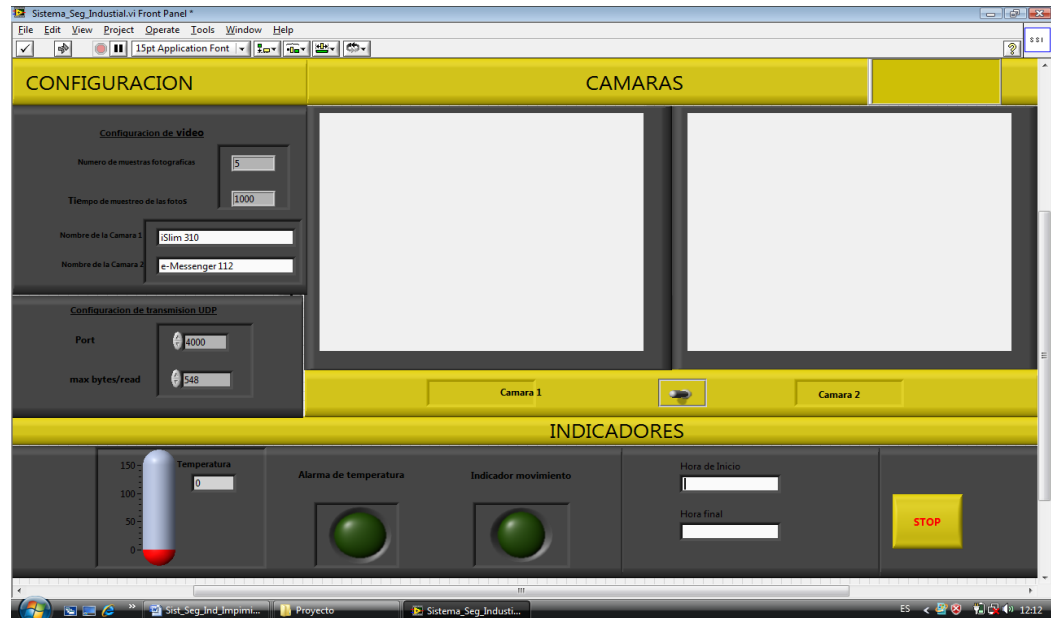


Figura3.9 Panel Frontal del Sistema

Este bloque es el panel frontal en LabView de nuestro Sistema en el cual la persona encargada de monitorear lo que está sucediendo en cada sector de la industria.

Se puede observar que este panel consta de tres partes:

- Configuración: de video y de transmisión UDP

- Cámaras: visualización de cada sector de la industria
- Indicadores: estado de los sensores

3.3.3. Programación en SyBase

En SyBase creamos una base de datos en la cual tenemos una tabla que es la estructura básica donde se almacena la información de acuerdo a las necesidades que queremos registrar en nuestro proyecto como son la fecha y hora en línea en la que ocurran los eventos que está enviando LabView así como también la imagen que se captura en LabView cuando se da un suceso con los sensores.

Hemos desarrollado el sistema de tal manera que el usuario pueda utilizar el panel frontal de LabVIEW como interfaz para enviar datos hacia la bases de datos utilizada por el usuario y poder monitorear constantemente la información adquirida por el sistema por ello describiremos brevemente la programación en SyBase.

Para crear una conexión entre LabView y Sybase debemos hacer lo siguiente:

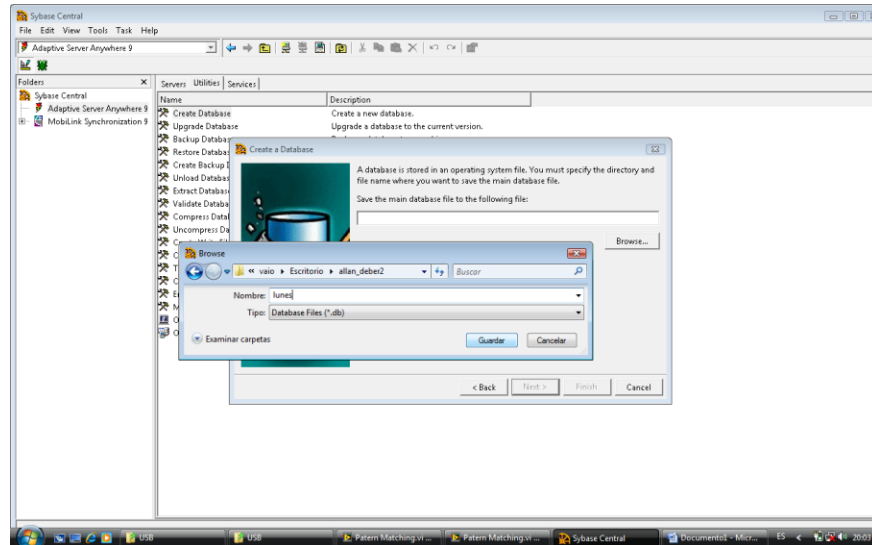


Figura3.10 Creación de la base de datos

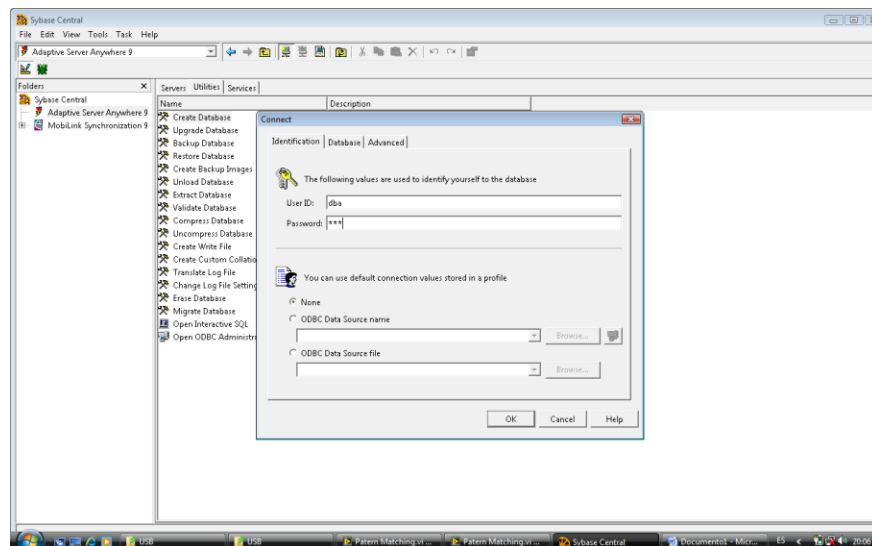


Figura3.11 Ingreso de usuario y contraseña en SYBASE

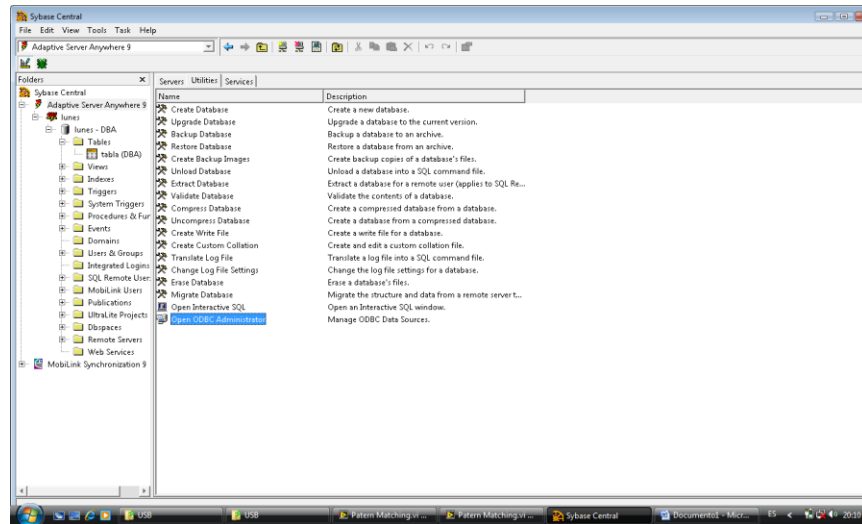


Figura3.12 Conectar con el ODBC

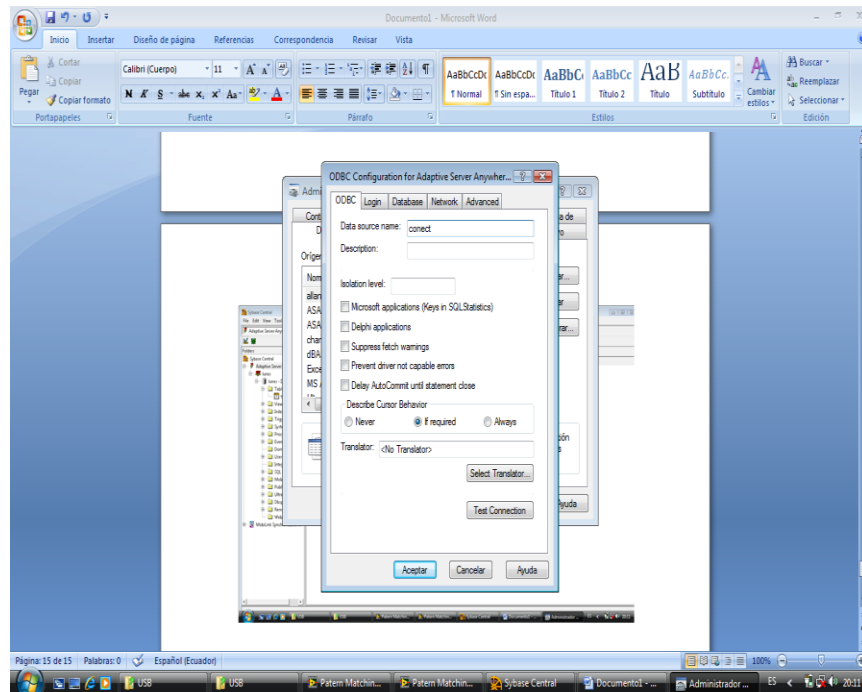


Figura3.13 Configuración del ODBC

El sistema enviará información a esta base de datos, pues en el código de LabVIEW se designó a la conexión dba, en el bloque de adquisición de imágenes y comunicación con Sybase. Entonces cada vez que el sistema lo requiera, enviará información a las tablas correspondientes de esta base de datos.

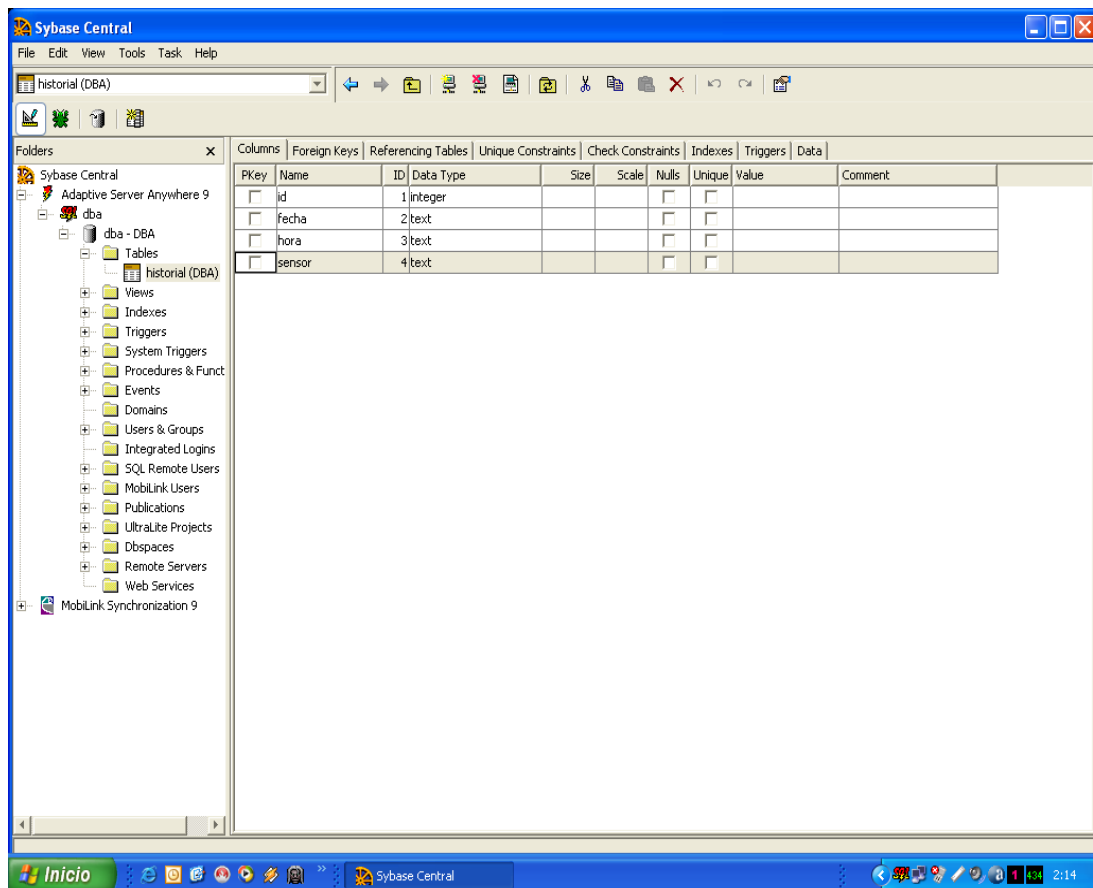


Figura3.14 Base de datos SYBASE

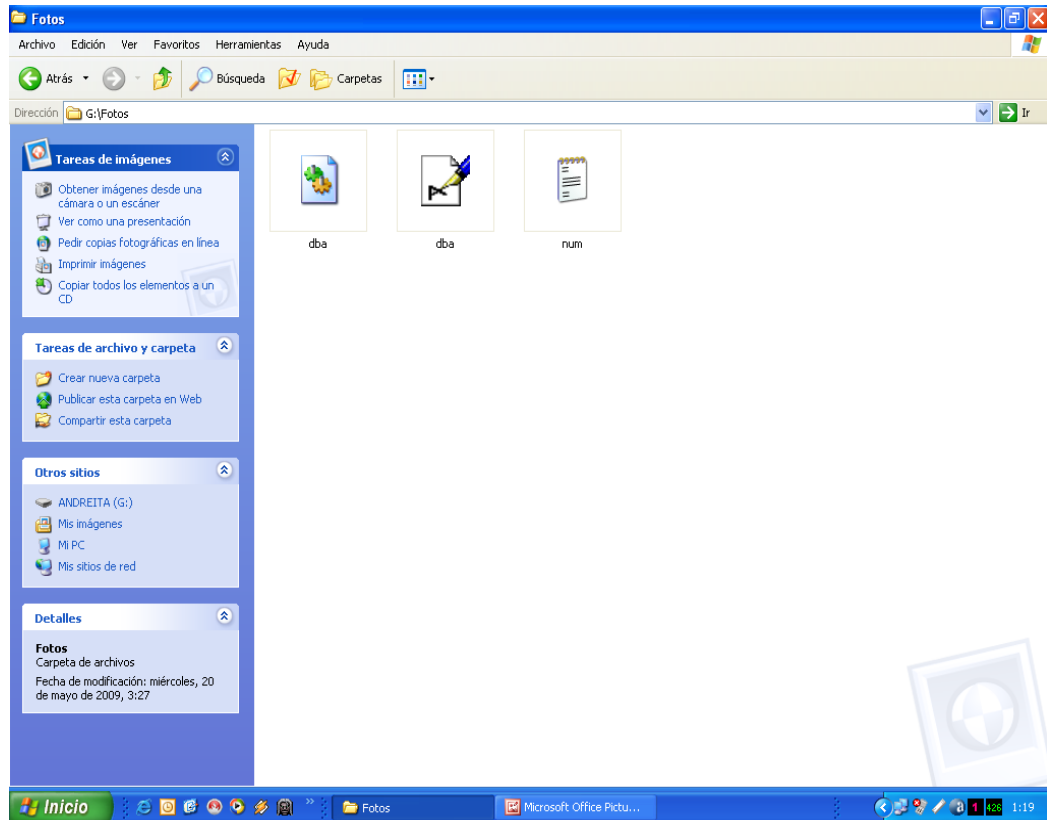


Figura3.15 Carpeta donde se guardan las imágenes capturadas

Como se puede observar en la figura esta base de datos tiene una tabla que permite monitorear continuamente el sistema.

CAPÍTULO 4

4. Implementación y Puesta en Marcha del Sistema de Seguridad Industrial

4.1. Tarjetas del Controlador con elementos ensamblados

Teniendo las tarjetas con las pistas ya diseñadas procedemos a soldar todos los elementos de nuestro sistema obteniendo:



Figura4.1 PCB y elementos del sistema

4.2. Montaje del Sistema

Con las tarjetas ya diseñadas, con todos los elementos soldados en ésta y probada lo que nos queda por hacer es montar el sistema.

Para esto hemos construido en acrílico un pequeño cuarto que simula un sector de la industria, aquí vamos a colocar nuestras tarjetas y en la parte interna el sensor de movimiento y el cable de red conectado a la tarjeta Ethernet además instalar las cámaras en la parte superior del cuarto en la posición que encontremos más adecuada para el monitoreo.

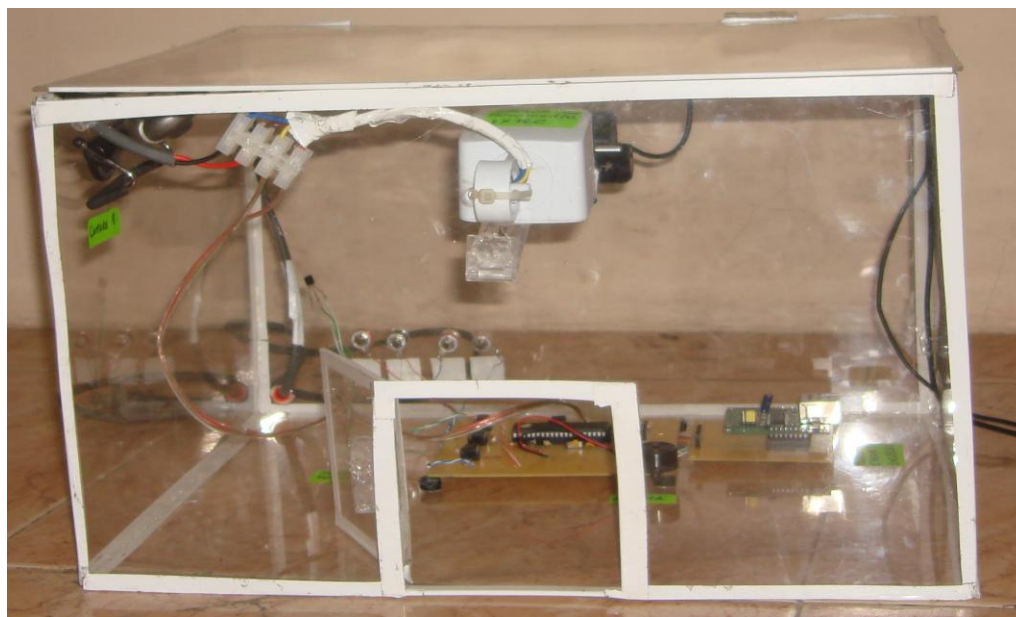


Figura4.2 Vista frontal sistema montado

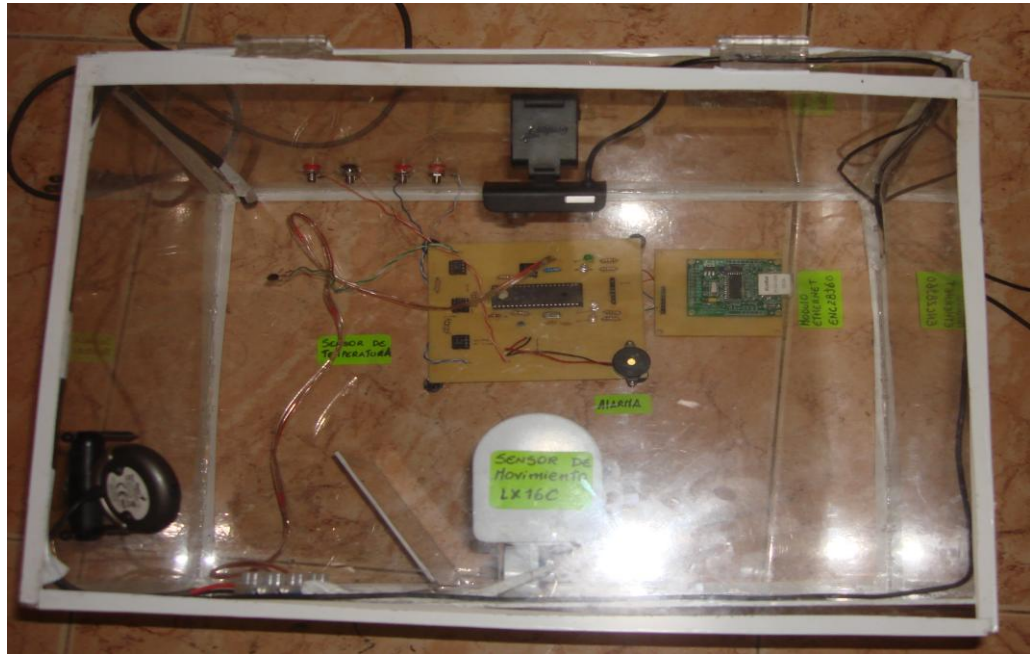


Figura4.3 Vista superior sistema montado

4.3 Puesta en Marcha del Sistema

Con el hardware ya listo es decir instalado correctamente y probado procedemos al manejo de la interfaz hecha en LabView junto con la Base de Datos para la adquisición, monitoreo y registro de datos pero para ello cabe recalcar que debemos tener instalados tanto el software de LabView con todas las librerías utilizadas así como también el SyBase.

Para empezar debemos hacer una prueba de conectividad Ethernet entre en el PIC y la PC para saber que se puede hacer la adquisición de datos por medio de un PING a la dirección IP del microcontrolador, como hacer esto se detallará más en el manual del usuario.

Si vemos que existe respuesta al PING por parte del microcontrolador, entonces tendremos la certeza de que el módulo Ethernet está funcionando correctamente.

Luego de esto ahora si podemos pasar al manejo de la interfaz desarrollada en LabVIEW para a ello abrimos la aplicación que creamos la requiere el ingreso de distintos parámetros por parte del encargado del monitoreo de la industria los cuales se encuentran detallados en el manual del usuario.

Con todos los parámetros necesarios ya ingresados ponemos en marcha la aplicación la cual estará monitoreando de manera continua los sensores y las cámaras.



Figura4.4 Panel Frontal

Si en algún momento la temperatura sobrepasa los 40°C que es lo que hemos configurado en el microcontrolador entonces se encenderá una alarma en el sector donde se ha sentido el problema y a la vez esta señal de alarma puede ser visualizada por medio de un indicador de alerta y de un indicador de nivel de temperatura que tenemos en el panel frontal de LabView para que la persona que esté monitoreando sea capaz de tomar una decisión inmediata con respecto a lo que está sucediendo.

Al llegar la señal de alarma a LabView gracias a las librerías Database contamos con que el suceso quedará registrado en la base de datos SyBase con la fecha y hora que ocurrió así como también de una imagen.



Figura4.5 Sensor de temperatura activado

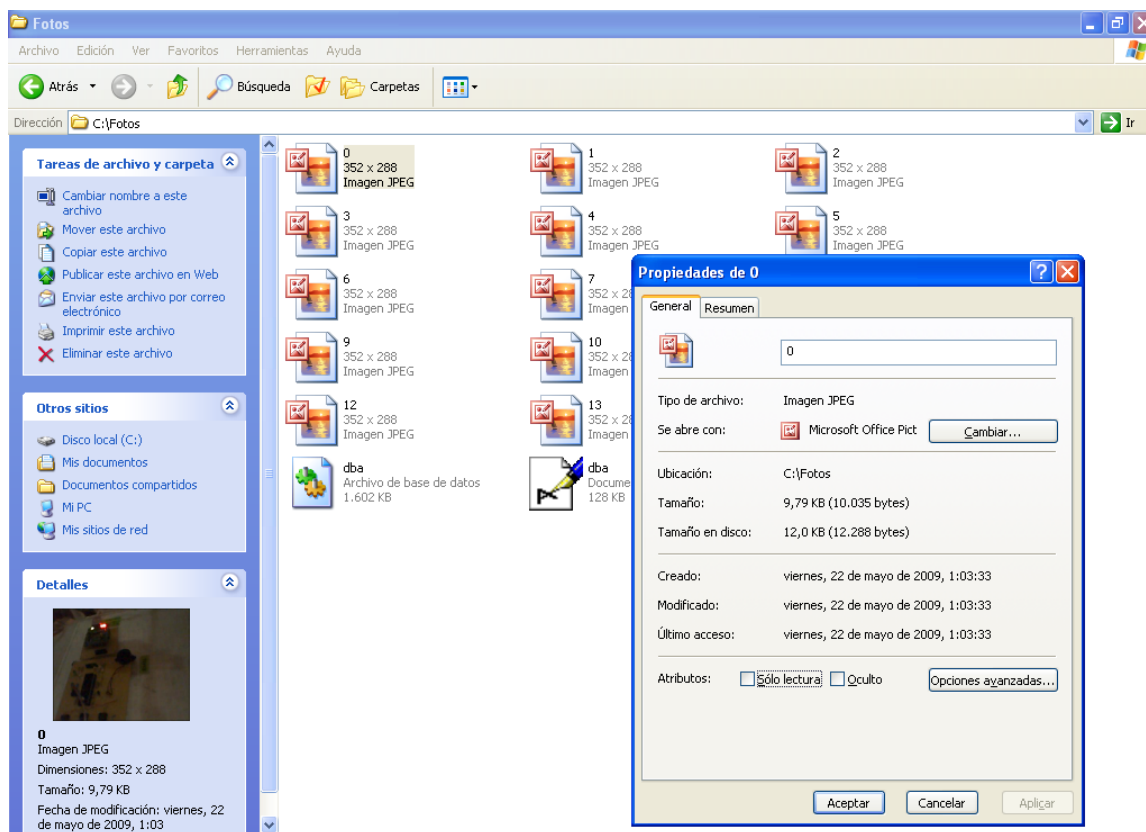


Figura4.6 Carpeta donde se guardan las fotos tomadas

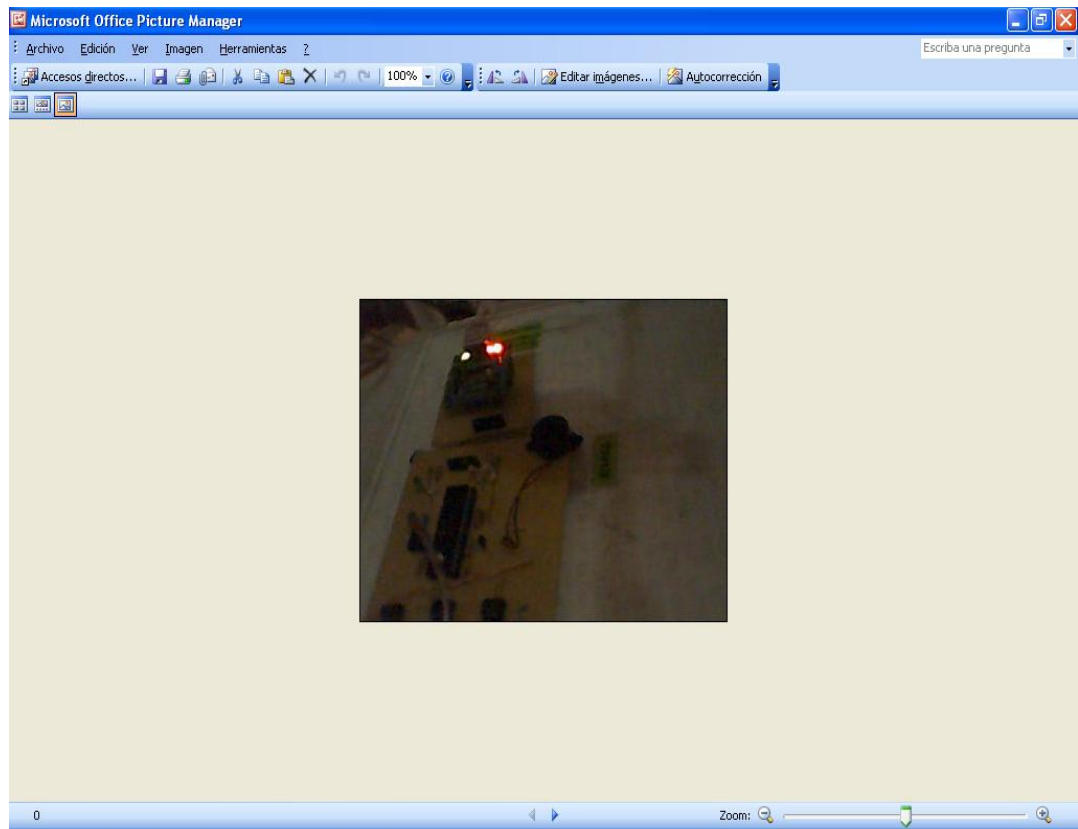
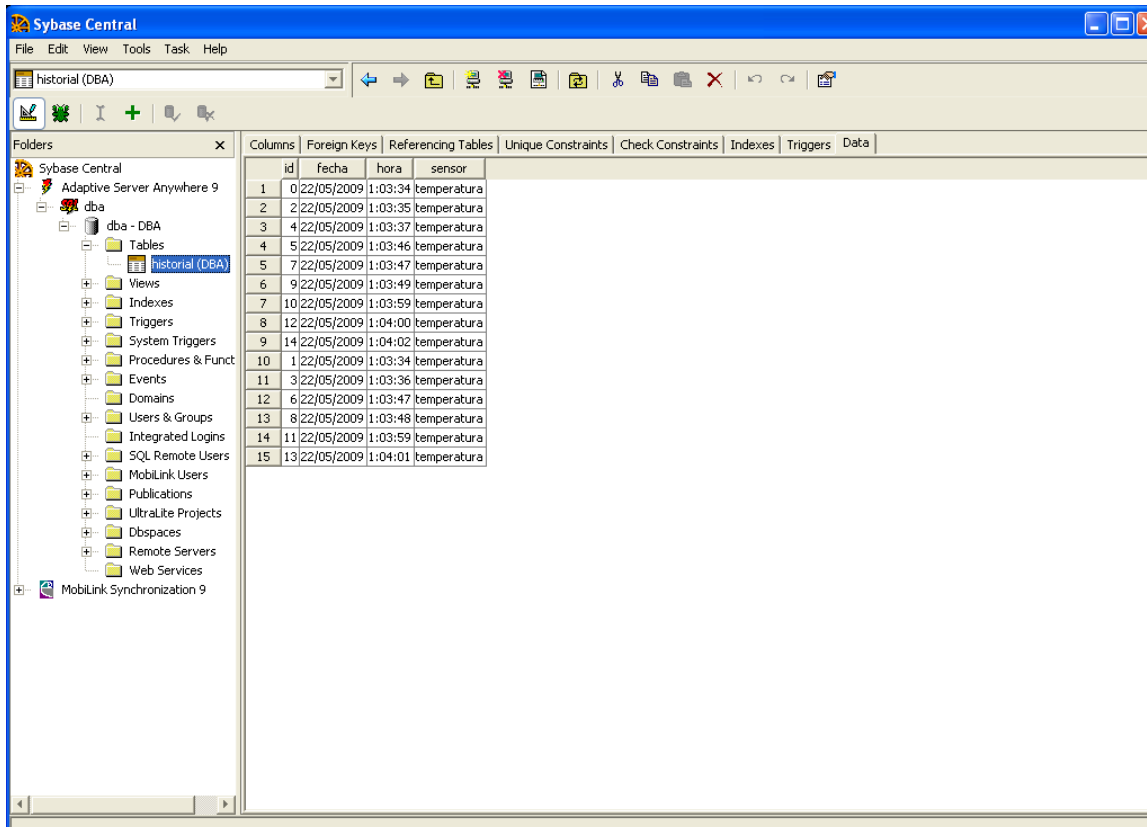


Figura4.7 Una de las fotos tomadas



id	fecha	hora	sensor
1	02/22/05/2009	1:03:34	temperatura
2	22/05/2009	1:03:35	temperatura
3	42/05/2009	1:03:37	temperatura
4	52/05/2009	1:03:46	temperatura
5	72/05/2009	1:03:47	temperatura
6	92/05/2009	1:03:49	temperatura
7	102/05/2009	1:03:59	temperatura
8	122/05/2009	1:04:00	temperatura
9	142/05/2009	1:04:02	temperatura
10	122/05/2009	1:03:34	temperatura
11	322/05/2009	1:03:36	temperatura
12	622/05/2009	1:03:47	temperatura
13	822/05/2009	1:03:48	temperatura
14	1122/05/2009	1:03:59	temperatura
15	1322/05/2009	1:04:01	temperatura

Figura4.8 Tabla de la base de datos registrando temperatura

En el caso de que se esté sensando la presencia de alguien entonces en el panel frontal de LabView se encenderá un indicador de alerta además de que por medio de las cámaras podemos estar visualizando el sector de la industria donde se esté presentando el suceso, a la vez nuestro sistema captura imágenes en ese instante y las guarda en una base de datos cabe recalcar que el número de imágenes que capture así como el tiempo de muestreo va a depender de la configuración que se haya

configurado en el panel frontal de LabView, además el suceso queda registrado en la base de datos con la fecha y hora.



Figura4.9 Sensor de movimiento activado

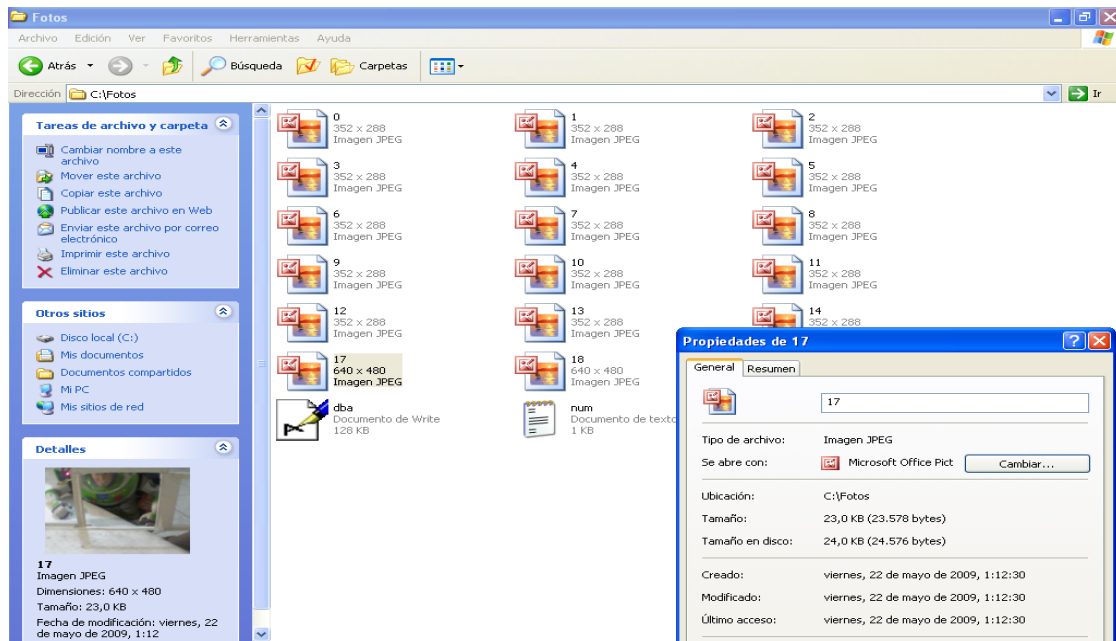


Figura4.10 Fotos tomadas cuando se activo el sensor de movimiento.

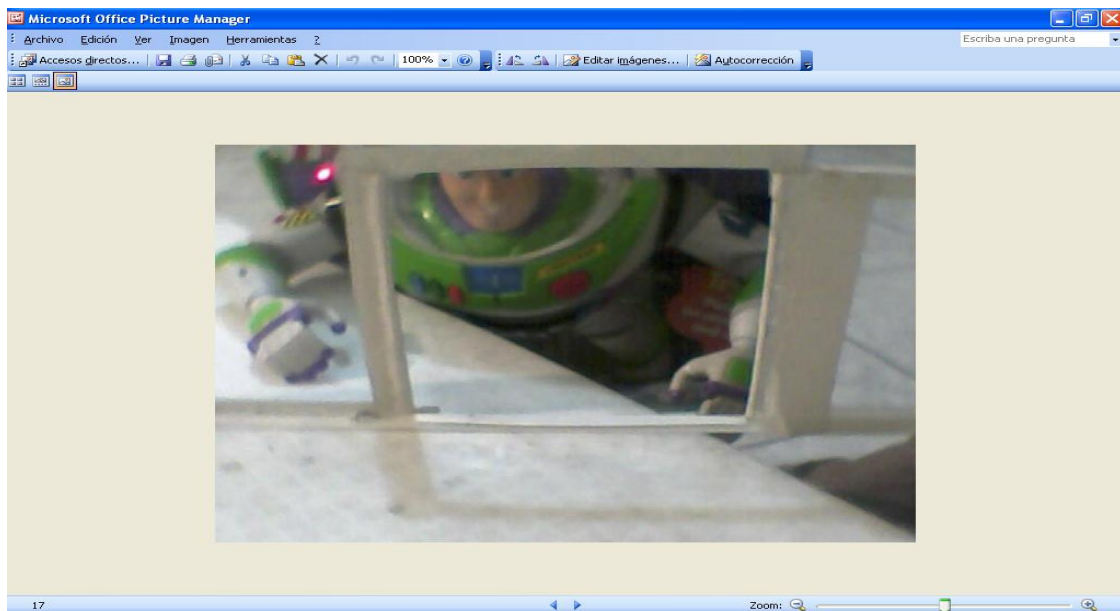
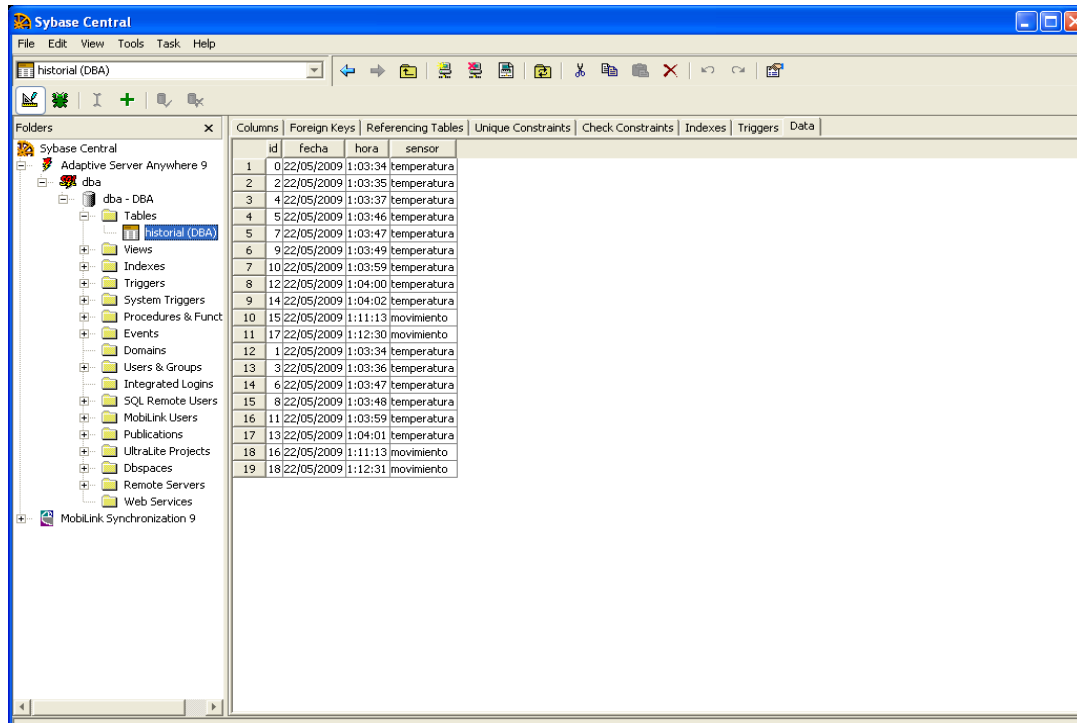


Figura4.11 Una de las fotos tomadas cuando se activo el sensor de movimiento



The screenshot shows the Sybase Central interface. The left pane displays a folder tree with 'historial (DBA)' selected under 'Tables'. The main pane shows a table with the following data:

id	fecha	hora	sensor
1	0	22/05/2009 1:03:34	temperatura
2	2	22/05/2009 1:03:35	temperatura
3	4	22/05/2009 1:03:37	temperatura
4	5	22/05/2009 1:03:46	temperatura
5	7	22/05/2009 1:03:47	temperatura
6	9	22/05/2009 1:03:49	temperatura
7	10	22/05/2009 1:03:59	temperatura
8	12	22/05/2009 1:04:00	temperatura
9	14	22/05/2009 1:04:02	temperatura
10	15	22/05/2009 1:11:13	movimiento
11	17	22/05/2009 1:12:30	movimiento
12	1	22/05/2009 1:03:34	temperatura
13	3	22/05/2009 1:03:36	temperatura
14	6	22/05/2009 1:03:47	temperatura
15	8	22/05/2009 1:03:48	temperatura
16	11	22/05/2009 1:03:59	temperatura
17	13	22/05/2009 1:04:01	temperatura
18	16	22/05/2009 1:11:13	movimiento
19	18	22/05/2009 1:12:31	movimiento

Figura4.12 Tabla de la base de datos registrando movimiento

CONCLUSIONES Y RECOMENDACIONES

- 1) Con nuestro proyecto se pretende dar un ejemplo del sinnúmero de aplicaciones que se pueden realizar utilizando LABVIEW y su aplicación IMAQ VISION. Además este trabajo representa un estudio básico en el que se demuestra que se puede llegar a implementar proyectos interesantes con aplicaciones comerciales en nuestro país, es decir que se pueden generar soluciones propias y no depender de soluciones importadas.
- 2) Los sensores de: temperatura LM35 y de movimiento LX16C fueron escogidos para la implementación del presente proyecto debido a las características y facilidades que presentan. La detección de movimiento mediante el sensor LX16C con tecnología de rayos infrarrojos es capaz de detectar patrones cambiantes en un rango de 2-11m a 180°, lo que lo hace interesante para esta aplicación. Igualmente el sensor de temperatura con su rango desde -55°C a +150°C es más que suficiente para el prototipo planteado. Añadiendo a todo esto el bajo costo de los mencionados sensores, esto hace posible la realización práctica de este proyecto con limitados recursos económicos.
- 3) Como protocolo de comunicación se utiliza el UDP para la comunicación entre los dispositivos que estén en red tanto para el control de

temperatura como para la entrada - salida de personal a un lugar determinado. Se utilizaron cámaras WEB e IP en el manejo de imágenes. El microcontrolador 18F4520 es el que permite el envío de señales de los sensores a la computadora por medio de Ethernet utilizando el módulo ENC28J60.

- 4) La ventaja de usar la red Ethernet es que no tiene limitante de número máximo de dispositivos en la red lo que si ocurre con los protocolos RS-232 y el RS485. Si se requiere agregar más dispositivos, la solución sería agregar un punto de red donde se lo colocará. Lo que se deberá controlar será la cantidad de tráfico que los equipos de red soporten.
- 5) El software para la interfaz del sistema fue LabVIEW 8.5, porque permite que el desarrollador se concentre en la solución que está implementando y no en los detalles de programación, además su interfaz gráfica lo convierte en una muy buena opción para la elaboración del sistema. La desventaja de este programa es que para lograr comunicarse con tarjetas o equipos externos es necesario adquirir librerías que no vienen con el programa original.
- 6) El sistema cuenta con una base de datos desarrollada en SYBASE que permite tener información precisa y actualizada, lo cual nos proporciona

un control centralizado de los datos para ser compartidos y evitar la redundancia.

- 7) Para realizar la adquisición de imágenes con las cámaras IP debemos utilizar controles ActiveX enlazados con las librerías DLL (Dynamic Link Library) de la cámara y a su vez esta debe contener métodos como Picture, Snapshot, Size, format, etc. Para la cámara IP Genius la librería "MULTVIEW" que es la DLL de la cámara no contiene métodos para enlazarse con LabView por ello para la realización de este proyecto usamos cámaras Genius que nos brindan las características necesarias para la fácil captación de las imágenes que se desea para la simulación del programa.
- 8) Para una aplicación más robusta y de mayor precisión es recomendable utilizar el protocolo TCP/IP para el envío de datos que en el caso de existir colisiones o errores en la transmisión, sus capas se encargan de solicitar la retransmisión de paquetes lo que no ocurre con el protocolo UDP.
- 9) Para el sensor de temperatura utilizar un acondicionamiento de señal para que trabaje a su máximo rango además verificar que los puertos con los que estemos trabajando estén habilitados.

ANEXO A

MANUAL DE USUARIO

1. INTRODUCCIÓN

La operación y puesta en marcha del sistema serán descritos a continuación para una correcta utilización a fin de evitar un mal funcionamiento que pueda afectar al control del sistema.

2. ANTES DEL ENCENDIDO DEL SISTEMA

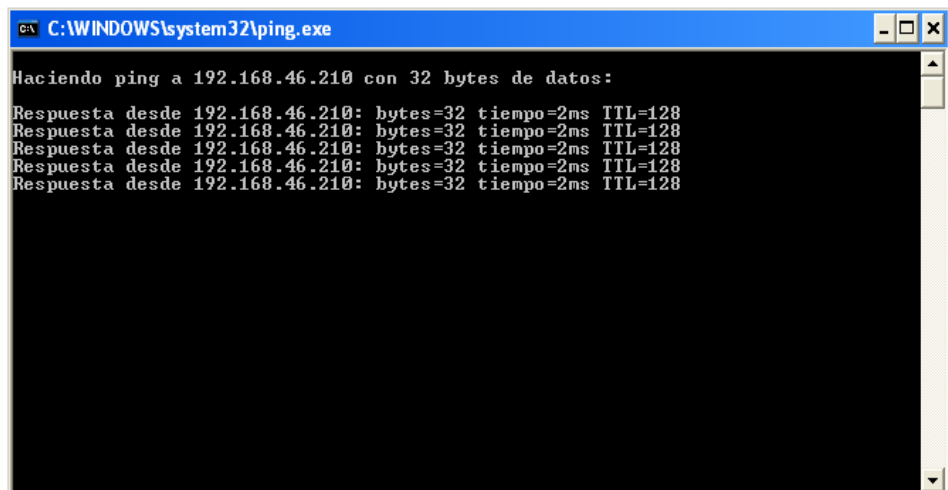
Verificar la correcta alimentación y polaridades con las que se va a alimentar la tarjeta controladora.

Poner en red el PIC con la PC configurándoles direcciones IP dentro del mismo rango para que pueda existir comunicación entre estos.

3. FUNCIONAMIENTO Y PUESTA EN MARCHA

- Luego de alimentar la tarjeta se procederá a verificar que los cuatro leds de la tarjeta de interfaz Ethernet estén titilando para saber que hay comunicación.
- Hacer ping para verificar la conexión entre el PIC y la red de la siguiente manera:

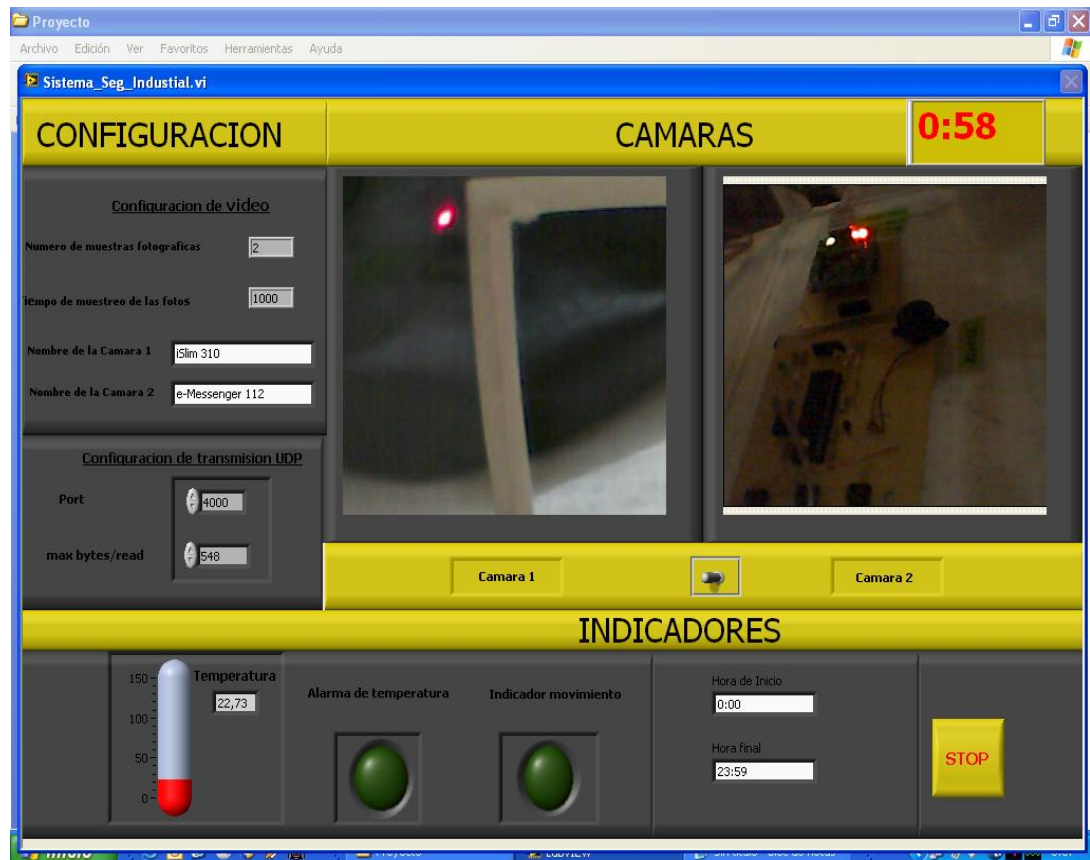
INICIO > Accesorios > Herramientas de sistema > ping
Dirección IP del PIC



```
C:\WINDOWS\system32\ping.exe
Haciendo ping a 192.168.46.210 con 32 bytes de datos:
Respuesta desde 192.168.46.210: bytes=32 tiempo=2ms TTL=128
Respuesta desde 192.168.46.210: bytes=32 tiempo=2ms TTL=128
Respuesta desde 192.168.46.210: bytes=32 tiempo=2ms TTL=128
Respuesta desde 192.168.46.210: bytes=32 tiempo=2ms TTL=128
```

- Asegurarse que las cámaras estén conectadas.

- Abrir el software de aplicación en LabView *Sistema_Seg_Ind.*
Para poder observar los valores que están captando los sensores realice lo siguientes pasos de configuración en panel frontal de dicha aplicación:



1. En la sección de configuración de transmisión UDP seleccione el puerto de comunicación Ethernet por donde quiere adquirir los datos que provienen de la

tarjeta controladora y también el número máximo de bits a leer.

2. En la sección de configuración de video ponga el número de muestras de fotos a tomarse y el tiempo de muestreo de cada foto que usted desee para cuando se active algún sensor.

3. También ponga los nombres de las cámaras a usar en la sección de configuración de video.

- Escriba la hora de inicio y hora final para configurar el tiempo en el que usted desee que esté corriendo la aplicación.
- Abrir la tabla llamada *historial* de Base de Datos en Sybase y conectar el .dba para que exista la conexión con LabView.
- Correr el programa en LabView para visualizar lo que está ocurriendo.
- A partir de este momento el sistema está listo para hacer el correcto monitoreo de los sensores en cada sector de la industria.

ANEXO B

CODIGO FUENTE

```
program sist_seg_ind
include "eth_enc28j60"
include "eth_enc28j60_api"

dim mymacaddr as byte [6]
dim myipaddr as byte [4]
dim getRequest as byte[20] ' 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 dato as word
dim tmp as string[100]

sub function Spi_Ethernet_UserTCP(dim byref remoteHost as byte[4], dim remotePort, localPort, reqLength as word) as word
result=0
end sub

sub function Spi_Ethernet_UserUDP(dim byref remoteHost as byte[4], dim remotePort, destPort, reqLength as word) as word
result = 0
if destport = 200 then
for i=0 to 3
getRequest[i]=spi_ethernet_getbyte()
next i
```

```

        while(reqLength <> 0)
            spi_ethernet_putbyte(spi_ethernet_getByte())
            reqLength = reqLength - 1
        wend
    end if
end sub

main:
    cmcon=7
    trisa=1
    adcon1=$0e
    trisd=0
    portd=0
    trise.0=1
    trise.1=0
    porte.1=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]=192
myipaddr[1]=168
myipaddr[2]=46
myipaddr[3]=210

IpAddr[0] = 192
IpAddr[1] = 168
IpAddr[2] = 46
IpAddr[3] = 203
delay_ms(1000)
spi_init()
delay_ms(3000)
spi_ethernet_init(portc,0,portc,1,mymacaddr,myipaddr,1)
delay_ms(3000)
while true
    spi_ethernet_dopacket()
    dato = adc_read(0)
    delay_ms(50)
    wordtostr(dato,txt)
    PORTD = not PORTD
    tmp="temp@"

```

```

Strcat(tmp,txt)
txt=tmp
spi_ethernet_dopacket()
spi_ethernet_sendUDP(IPAddr, 10001, 4000, @txt, Strlen(txt))
delay_ms(50)
tmp="alarma@"
if(dato>273) then
porte.1=1
txt="1"
Strcat(tmp,txt)
txt=tmp
spi_ethernet_dopacket()
spi_ethernet_sendUDP(IPAddr, 10001, 4000, @txt, Strlen(txt))
delay_ms(50)
else
porte.1=0
txt="0"
Strcat(tmp,txt)
txt=tmp

```

```

spi_ethernet_dopacket()
spi_ethernet_sendUDP(IPAddr, 10001, 4000, @txt, Strlen(txt))
delay_ms(50)
end if
tmp="movimiento@"
if(porte.0=1) then
txt="1"
Strcat(tmp,txt)
txt=tmp
spi_ethernet_dopacket()
spi_ethernet_sendUDP(IPAddr, 10001, 4000, @txt, Strlen(txt))
delay_ms(50)
else
txt="0"
Strcat(tmp,txt)
txt=tmp
spi_ethernet_dopacket()
spi_ethernet_sendUDP(IPAddr, 10001, 4000, @txt, Strlen(txt))
delay_ms(500)
end if
wend
end.

```

BIBLIOGRAFÍA

1. Guía para elaborar una Tesis de Grado
<http://www.cib.espol.edu.ec>
2. Datasheets de los integrados utilizados en la implementación del proyecto.
<http://www.datasheetcatalog.com>
3. FABRICIO ECHEVERRIA, Tutorial de aplicación a la base de datos 1º Edición; ESPOL; Guayaquil-Ecuador, 2008.
<http://blog.espol.edu.ec/pechever/manual-de-lab-view-con-sql/>
4. ANGULO USATEGUI JOSÉ, Microcontroladores avanzados dsPIC 1º Edición; Thomson; Madrid-España, 2006; pp 3-495.
5. MALONEY J. TIMOTHY, Electrónica industrial moderna 3º Edición; Pretince Hall; Naulcalpan de Juárez-México, 2007; pp 526-530.
6. GARY W. JOHNSON AND RICHARD JENNINGS, Programación gráfica con LabVIEW .
7. Hojas de fabricante de especificación del PIC18F4520
<http://www.microchip.com>
8. Manual del Módulo ET-MINI ENC28J60 (Ethernet Controller). Versión 1, septiembre 2007.
9. AQUILINO RODRIGUEZ PENIN, Comunicaciones Industriales, MARCOMBO S.A, 1ª edición.
10. <http://www.national.com/mpf/LM/LM35.html>
11. <http://www.lxpir.com/instructions/Lx16C.htm>

12. JOSÉ RAFAEL LAJARA Y JOSÉ PELEGRÍ, Labview: Entorno Gráfico De Programación, Febrero 2007.
13. Guía de Introducción de LABVIEW en 6 horas por National Instrument.