



Universidad del Azuay

Facultad de Ciencia y Tecnología

Escuela de Ingeniería Electrónica

“Diseño e Implementación de un Sistema de Seguridad Doméstico con Capacidad de Monitoreo y Control Vía Web”

Trabajo de Graduación previo a la obtención del título de Ingeniero Electrónico

Autores

Patricio Araujo

Daniel Capelo

Director

Lcdo. Leopoldo Vásquez

Cuenca, Ecuador

2009

DEDICATORIA

Este trabajo está dedicado a nuestras familias quienes han sido en todo momento un pilar fundamental, a través de su apoyo y aliento, para llegar a esta etapa en nuestras vidas.

De igual forma dedicamos este proyecto a todas aquellas personas que han formado parte de nuestras vidas brindándonos su amistad, consejos y enseñanzas.

AGRADECIMIENTO

Agradecemos infinitamente a nuestras familias, amigos y profesores que han contribuido enormemente en nuestra formación personal y académica, durante estos años, y que seguramente lo harán durante el resto de nuestras vidas.

Expresamos de manera especial nuestro agradecimiento a nuestro profesor y director de tesis el Lcdo. Leopoldo Vásquez y al Ing. Leonel Pérez sin quienes la realización de este proyecto no habría sido posible.

RESUMEN

El presente trabajo pretende el uso de elementos no convencionales dentro de los sistemas de seguridad, a fin de extender el rango de los métodos de alertar, acerca de anomalías, dentro del ambiente del domicilio. Mediante alertas telefónicas hacia líneas fijas ó móviles se logra notificar al propietario del inmueble, dentro o fuera del país. El control telefónico es full dúplex para recibir el estatus (monitoreo) y actuar sobre el sistema.

El monitoreo y el control del dispositivo se puede lograr desde el Instrumento Virtual publicado en la Web. La información del monitoreo desde los sensores, cámaras web y estado de los actuadores es tomada con seguridad garantizada. La seguridad es administrada por el propietario, pero también se pueden autorizar otros administradores.

ABSTRACT

This work aims to extend the use of unconventional elements within the security systems in order to expand the range of methods to warn about the anomalies in the home environment. By phone alerts to fixed and mobile lines is achieved, to notify to the propertie's owner, either inside or outside the country. The phone control is full duplex because is capable of receive status (monitoring) and to "act" over the system.

Acting or monitoring actions can be achieved by a Web Published Virtual Instrument. Monitoring information is taken from sensors, Webcams and actuator's status with all security guarantee. Main security administration is given by the owner, but other people can be involved in administration too.

Índice de Contenidos

Dedicatoria.....	ii
Agradecimientos.....	iii
Resumen.....	iv
Abstract.....	v
Índice de Contenidos.....	vi
Índice de Cuadros e Ilustraciones.....	ix
Índice de Anexos.....	xii
INTRODUCCION.....	1
CAPITULO 1. MÓDULO DE ADQUISICION (PIC)	
1.1. Introducción.....	4
1.2. Estructura Interna del PIC.....	5
1.2.1. Procesador o UCP.....	5
1.2.2. Memoria de Programa.....	6
1.2.3. Puntero de Programa.....	6
1.2.4. Pila o Stack.....	6
1.2.5. Memoria de Datos.....	7
1.2.6. Memoria EEPROM.....	9
1.2.7. Reloj del Sistema.....	9
1.3. Puertos de Entrada y Salida.....	11
1.4. Circuitos Auxiliares e Interfaces.....	12
1.4.1. Fuente de Alimentación.....	12
1.4.2. Pines del Microcontrolador.....	13
1.4.3. Circuitos Hacia los Actuadores.....	14
1.4.4. Interfaz del Puerto Serial.....	15
1.4.5. Conexión del LCD.....	16
1.5. Comunicación Serial.....	17
1.5.1. Velocidad de Transmisión Serial.....	18
1.5.2. Trama de Dato Serial.....	18
1.6. Software del Micro controlador.....	19

1.6.1. Set de Instrucciones.....	20
1.6.2. Diagrama de Flujo.....	23
1.6.3. Funcionamiento.....	24
CAPITULO 2. LÍNEA TELEFÓNICA (DTMF)	
2.1. Introducción.....	25
2.2. Recepción con el MT8870.....	27
2.2.1. Función.....	28
2.2.2. Estructura Interna.....	30
2.2.2.1. Resumen de Pines.....	31
2.2.3. Interface con la Línea Telefónica.....	32
2.2.4. Interacción con el PIC.....	32
2.3. Transmisión con el ISD25XX.....	33
2.3.1. Función.....	34
2.3.1.1. Modos de Funcionamiento.....	35
2.3.2. Estructura Interna.....	35
2.3.2.1. Resumen de Pines.....	37
2.3.3. Circuito Auxiliar de Grabación – Reproducción.....	38
2.3.4. Tarjeta de Pruebas del ISD.....	41
CAPITULO 3. DISEÑO Y MONTAJE DEL HARDWARE	
3.1. Sensores.....	44
3.1.1. Sensores de Movimiento.....	47
3.1.2. Sensores Magnéticos.....	49
3.2. Cámara WEB.....	50
3.3. Diseño del PCB, Prótel 99 SE.....	51
3.4. Montaje del PCB.....	57
3.5. Pruebas de Funcionamiento.....	60
CAPITULO 4. SOFTWARE SOBRE LABVIEW	
4.1. Introducción al ambiente LABVIEW.....	63
4.1.1. Requerimientos para la Instalación.....	66
4.2. Modulo de Adquisición.....	67
4.3. Software sobre LABVIEW (lenguaje G).....	68

4.3.1. Diagrama de Flujo.....	68
4.3.1.1. Funcionamiento.....	69
4.3.2. Diagrama de Bloques.....	70
4.3.3. Panel Frontal.....	78
4.4. Comunicación con el PIC.....	79
4.5. Comunicación con la WEB.....	80
4.6. Seguridad de páginas web.....	88
4.6.1. Firma Digital.....	89
CONCLUSIONES.....	90
BIBLIOGRAFIA.....	91
ANEXOS.....	93

Índice de Cuadros e Ilustraciones

Tabla 1.1.a:	Capacitores según Frecuencia del Cristal a usar.....	10
Tabla 1.1.b:	Cristales Utilizados.....	10
Tabla 1.2:	Instrucciones Orientadas a Registros.....	21
Tabla 1.3:	Instrucciones Orientadas a bits.....	21
Tabla 1.4:	Instrucciones Orientadas a Literales.....	22
Tabla 2.1:	Matriz DTMF.....	26
Tabla 2.2:	Posibilidades de los Pines del MT8870.....	29
Tabla 2.3:	Frecuencia del Cristal Según ISD Utilizado.....	37
Tabla 3.1.a:	Tipos de Sensores.....	45
Tabla 3.1.b:	Tipos de Sensores.....	46
Tabla A.1:	Identificación de Errores de Adquisición de Video.....	103
Figura 1.1:	Memoria de Programa, Puntero y Pila.....	7
Figura 1.2:	memoria de Datos.....	8
Figura 1.3:	Configuración de Oscilador Externo.....	9
Figura 1.4:	Conexión RC a la Entrada del Oscilador.....	11
Figura 1.5:	Fuente de Alimentación.....	12
Figura 1.6:	Pines del PIC.....	14
Figura 1.7:	Disparo de los Relés.....	15
Figura 1.8:	Borneras para los Actuadores.....	15
Figura 1.9:	Interfaz Serial.....	16
Figura 1.10:	Conexión LCD.....	17
Figura 1.11:	Diagrama de Flujo Para el PIC.....	23
Figura 2.1:	Conmutador de la Línea Telefónica.....	27
Figura 2.2:	Indicador del "Ring".....	28
Figura 2.3:	Diagrama de Bloques del MT8870.....	30
Figura 2.4:	MT8870 Encapsulado de 18 Pines.....	31
Figura 2.5:	Interfaz con la Línea Telefónica.....	32
Figura 2.6:	Pinaje del ISD2560.....	36
Figura 2.7:	Diagrama de Bloques del ISD2560.....	37
Figura 2.8:	Configuración Push Buttom.....	39
Figura 2.9:	Configuración de Reproducción Controlada por el PIC.....	40

Figura 2.10:	Circuito que Conmuta la Salida del ISD hacia la Línea.....	40
Figura 2.11:	Dip-Switch de Control del ISD2560.....	41
Figura 2.12:	Circuito del ISD.....	42
Figura 2.13:	Controles de la Tarjeta Universal.....	43
Figura 2.14:	Circuito de la Tarjeta Universal ISD.....	43
Figura 3.1:	Esquema de un Sensor de Óptica Alineada.....	47
Figura 3.2:	Esquema de un Sensor de Óptica Reflexiva.....	48
Figura 3.3:	Sensor de Movimiento.....	49
Figura 3.4:	Sensor Magnético.....	49
Figura 3.5:	Cámara Web.....	51
Figura 3.6:	Representación Esquemática y Huella de Un DB9.....	52
Figura 3.7.a:	Cara Superior del PCB del Equipo.....	54
Figura 3.7.b:	Cara Superior del PCB de la Tarjeta Universal del ISD.....	55
Figura 3.8.a:	Cara Posterior del PCB del Equipo.....	56
Figura 3.8.b:	Cara Posterior del PCB de la Tarjeta Universal del ISD.....	57
Figura 3.9.a:	Esquema para el Montaje de Componentes del Equipo.....	58
Figura 3.9.b:	Esquema para el Montaje de Componentes de la Tarjeta del ISD.....	59
Figura 3.10:	Montaje Terminado del PCB.....	60
Figura 3.11:	Conector DB9 en Configuración “Loopback”.....	61
Figura 4.1:	Ejemplo de un Instrumento Virtual en su Etapa de Bloques.....	64
Figura 4.2:	Ejemplo de un Panel Frontal.....	65
Figura 4.3:	Arquitectura de Adquisición de Video.....	66
Figura 4.4:	Diagrama de Flujo del Proceso.....	68
Figura 4.5:	Inicialización del Puerto Serial.....	70
Figura 4.6:	Validación de un “Username” y un “Password”.....	71
Figura 4.7:	Diagrama de Bloques de la Adquisición, continua, de dos Webcams...	72
Figura 4.8:	Sub Instrumento Virtual OR6.....	73
Figura 4.9:	Transmisión Serial.....	74
Figura 4.10:	Recepción Serial.....	75
Figura 4.11:	Sub Instrumento Virtual “mantiene_constante”.....	76
Figura 4.12:	Diagrama de Bloques Funcional.....	77
Figura 4.13:	Panel Frontal del Instrumento.....	78
Figura 4.14:	Ejemplo de un Instrumento Virtual.....	81
Figura 4.15:	Habilitando Web Server.....	82

Figura 4.16:	Propiedades de la Pagina Web.....	83
Figura 4.17:	Titulo, Encabezado y Pie del Documento Web.....	83
Figura 4.18:	Establecimiento del Modo de Visualización.....	84
Figura 4.19:	Creación del Pagina HTML para el Panel Remoto.....	85
Figura 4.20:	Panel Remoto Abierto desde el Navegador de Internet.....	86
Figura 4.21:	Botones del Panel Remoto.....	86
Figura 4.22:	Instrumento Virtual Controlado desde el Navegador de Internet.....	87
Figura 4.23:	Interacción Red Privada y la Web.....	88
Figura A.1:	Indicación de Entradas y Salidas en el PCB terminado.....	93
Figura A.2:	Ejemplo de Conexión de un Actuador de Corriente Continua.....	96
Figura A.3:	Ejemplo de Conexión de un Actuador de Corriente Alterna.....	96
Figura A.4:	Partes Relevantes del Instrumento Virtual.....	98
Figura A.5:	Sección del Instrumento que comanda los Actuadores.....	99

Índice de Anexos

Anexo A:	Manual del Usuario.....	93
Anexo B.	Lista de Materiales.....	104
Anexo C.	Subrutina que Arma el Sistema.....	107
Anexo D.	Subrutinas de Configuración del Sistema.....	109
Anexo E.	Subrutina de Cambio de Estado de los Actuadores.....	111
Anexo F.	Subrutina de Alarma.....	113
Anexo G.	Subrutina de Reproducción con el ISD2560.....	115
Anexo H.	Subrutina de Identificación de Tonos.....	116
Anexo I.	Subrutinas de Comunicación Serial.....	117

Araujo Parra Iván Patricio

Capelo Ramón Daniel Santiago

Trabajo de Graduación

Lcdo. Leopoldo Vásquez

Septiembre del 2009

Diseño e Implementación de un Sistema de Seguridad Doméstico con Capacidad de Monitoreo y Control Vía Web

Introducción

Lo que se pretende es eliminar ciertas limitaciones de los sistemas de seguridad estándar, como agregar la posibilidad de controlar tipo on/off de forma remota ciertas áreas, con el propósito de simular presencias humanas. El objetivo es realizar el control y el monitoreo, desde cualquier lugar del globo, valiéndose de la web y de la red telefónica existente. Además se aspira a implementar un método de alarma telefónica automática, la cual previene de áreas violentadas hacia una línea determinada, preferiblemente celular, aunque es posible programar dos números a los cuales generar las notificaciones, sean estas fijas o móviles.

Todo el sistema es administrado desde un ordenador y a través un instrumento virtual generado en LABVIEW. Dicho instrumento genera indicadores del estado de los sensores y las líneas de control (actuadores), un área de la pantalla contiene el video en tiempo real de las zonas de vigilancia, mientras que el resto presenta los controles del equipo. LABVIEW estará configurado como servidor Web, valiéndose del protocolo TCP/IP, para que todos los resultados del sistema estén publicados en el internet a través de una página web que lleva el mismo nombre del archivo en LABVIEW (extensión .vi).

Para acceder a al monitoreo y el control del Instrumento de forma remota (desde el internet) a mas de conocer la dirección electrónica también se debe pertenecer a una lista de usuarios permitidos por el administrador de la pagina, es decir el propietario del inmueble.

Los sensores necesarios para llevar a cabo un test óptimo del estado del domicilio son: sensores magnéticos que alerten de las principales puertas y ventanas que se abran, sensores de movimiento que perciban alguna presencia dentro de las principales zonas de vigilancia y por último las cámaras web captarán de manera inequívoca todos los sucesos dentro de la casa.

Para comunicar los sensores y actuadores del dispositivo se plantea la adquisición y transmisión serial mediante los puertos de entrada y salida de un PIC, esto a fin de, evitar la inclusión de un DAQ *National Instruments* que es muy fiable pero bastante costoso.

A partir del modulo de adquisición se utilizan sus puertos de salida como líneas de control; esto es activar o desactivar, mediante relees, como mínimo 4 actuadores, que pueden ser desde un *breaker* general hasta las luces en diferentes habitaciones, todos ellos trabajan de forma individual, pero también pueden alternar su encendido y apagado temporizado como parte de una secuencia que simula la presencia dentro del inmueble.

Para tareas remotas simples se puede optar por utilizar el acceso telefónico del sistema. La recepción telefónica se realiza a través un decodificador de tonos para obtener control remoto (actuadores) solamente después de ingresar una clave de validación. En caso de presentarse una alerta se establecerá una llamada cuyos dígitos se generan desde un ISD pregrabado, de ser el caso, también se reproducirá un mensaje de voz una vez que la llamada sea atendida. Toda la parte electrónica será comandada por el PIC que adicionalmente registra, a través de un LCD, el estado del sistema.

Como es lógico una alerta enviada hacia un correo electrónico no es muy eficiente así como pretender la activación y desactivación del equipo desde la web; por ello y para

simplificar estos procesos lo más indicado es utilizar la línea telefónica para estas acciones remotas.

En caso de interrumpirse la energía convencional se plantea una fuente de poder adicional y recargable; la cual entra en funcionamiento hasta que la electricidad del domicilio se reponga.

Para la correcta instalación y funcionamiento del equipo se incluye un manual del usuario en el cual se explican detalles que facilitan un correcto montaje y operación de todos los elementos que componen el equipo.

El sistema completo se pone a prueba sobre una maqueta en la que se simulan todos los estados posibles del dispositivo, así como el funcionamiento de forma remota, tanto desde el teléfono como desde un navegador de Internet.

Existe una gran ventaja al explotar una red tan extendida como el internet ya que no importa el lugar del mundo en el que se encuentre el propietario del domicilio para conocer el estatus del sistema y más importante todavía puede llevar a cabo tareas remotas, pero de no contar con el servicio de internet o si se opta por un proceso más dinámico puede realizar las mismas tareas desde su teléfono celular. Estas dos opciones le permiten al usuario interactuar con su domicilio sin ninguna limitación de movilidad.

CAPITULO I

MODULO DE ADQUISICION (PIC)

1.1. Introducción

Típicamente en aplicaciones en las que se necesita adquirir señales hacia la computadora y usar esta información en un software conocido, lo mejor es utilizar el hardware recomendado por el fabricante del programa, sin embargo, si se conoce el protocolo, a través del cual se transmiten los datos, es posible crear un interfaz que funcione de manera eficiente. De esta forma la información no se ve comprometida y se pueden eliminar recursos innecesarios para el proceso específico que se persigue.

El objetivo de la adquisición de datos es medir una magnitud física y eléctrica como voltaje, corriente, temperatura, presión o sonido. Cuando la adquisición se basa en PC se combina hardware modular, software de aplicación y la PC para realizar los registros de medida. Todos los sistemas de adquisición captan los datos, los analizan y presentan la información. Las señales que se necesitan adquirir y procesar, en este caso, son indicaciones de tipo on/off (señales digitales), tanto en los sensores como en los actuadores, por lo tanto no necesitan tratamientos como linealización, escalado o la conversión. Por esta razón el modulo de adquisición solo capta las diferentes señales, las ordena y las traslada a la computadora.

Por tratarse de señales digitales, la comunicación serial es óptima para llevar a cabo el proceso de adquisición hacia la PC. Para aplicaciones analógicas o de mayor velocidad la transmisión serial no sería suficiente por sus limitaciones en cuanto a velocidad. Para llevar a cabo el control de los actuadores es necesario que la comunicación serial sea *full-dúplex*, es decir que el modulo de adquisición debe recibir y enviar datos.

Con un micro controlador se pueden cubrir los requerimientos en cuanto a numerosas entradas digitales y además se cuenta con la capacidad de comunicación serial en el mismo dispositivo; si además se toma en cuenta que un 16F87X incluye memoria no volátil se vuelve la elección precisa para el desarrollo del presente proyecto. De hecho hay varias características que se pueden resaltar con respecto a esta familia de PICs.

- 33 pines de propósito general.
- Capaz de trabajar con velocidades de hasta 20 Mhz.
- La memoria de Programa es de tipo Flash de hasta 2K (2048) instrucciones.
- La memoria de datos tiene capacidad de hasta 128 bytes.
- Posee una memoria adicional de tipo EEPROM de hasta 64 bytes.
- Puede trabajar con tensiones desde 2V hasta 5.5V de alimentación.
- Cada uno de sus pines puede proveer hasta 25mA de corriente.
- Posee tres temporizadores o *Timers* programables.
- Tiene un Conversor Analógico Digital de 8 canales con resolución de 10 bits.
- Posee un modulo CCP (Compara Captura PWM).
- Tiene un puerto de comunicaciones seriales de tipo USART.

1.2. Estructura Interna del Pic.

Un micro controlador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria, para lo cual consta de varias partes funcionales que cumplen con misiones específicas. Un micro es comparable con un computador por tener sus tres unidades funcionales esto es: UCP, memoria y unidades de entrada salida, pero además de estas hay partes adicionales o módulos que cumplen funciones especiales.

1.2.1. Procesador O UCP.

Es el elemento más importante del Integrado ya que se encarga de direccionar la memoria de instrucciones, recibe el código de la instrucción en curso, la codifica y luego la ejecuta. También debe cumplir con otras tareas directamente relacionadas

con las instrucciones como buscar los operandos, necesarios para ejecutarlas, y luego almacena el resultado de la operación.

1.2.2. Memoria de Programa.

Es la memoria de instrucciones, es decir que aquí es donde se graban las órdenes que el PIC debe ejecutar y por ello deben guardarse en orden. Es tan importante la instrucción por sí misma como por su posición (dirección) dentro de la memoria. Cuando comienza un programa, la instrucción que se ejecuta es la ubicada en la dirección 0x00. A esta dirección se le llama vector de RESET. También existe el vector de Interrupción que se encuentra en la dirección 0x04 y cuya instrucción solo se ejecuta si se produce una interrupción, por cualquiera de sus fuentes. La memoria de programa es variable según el micro. En la familia media 16F87X la memoria tiene la capacidad de 2048 instrucciones y es de tipo Flash lo que quiere decir que es rápidamente reprogramable.

1.2.3. Puntero de Programa.

Es un registro de 13 bits que almacena la dirección de la memoria de programa que se está por ejecutar. Cuando se produce un *Reset* el puntero se limpia, es decir que pasa a 0x00 de la memoria de programa, se extrae la instrucción de dicha dirección y luego el puntero se incrementa, este proceso se repite sucesivamente.

1.2.4. Pila O Stack.

En el transcurso normal del programa el puntero se incrementa con cada instrucción hasta que se produce un salto de programa, causado por una llamada a subrutina o por una interrupción. Antes de producirse el salto, hacia otro lugar del programa, se almacena la dirección del puntero para que concluida la subrutina el programa siga justo en la instrucción en la que se produjo el salto.

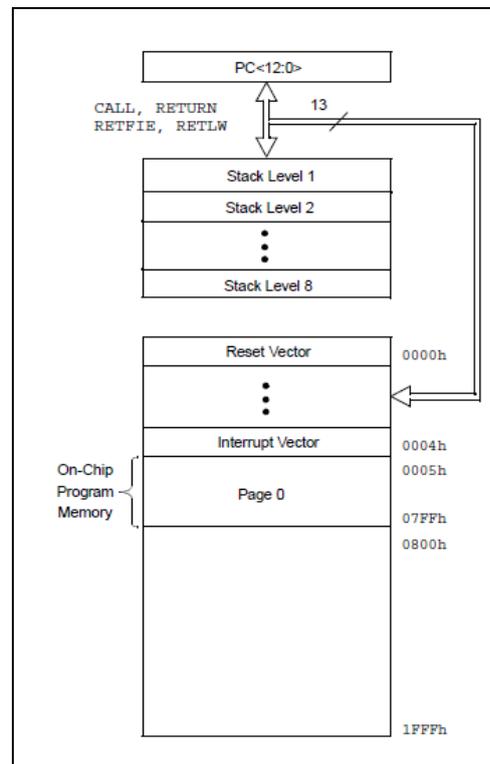


Figura 1.1. Memoria de programa, Puntero y Pila.

Fuente: Hoja de Datos 16F871.

1.2.5. Memoria de Datos.

Esta memoria está destinada a almacenar datos con los que operar durante la ejecución del programa. La memoria está particionada en 4 bancos que contienen registros de propósito general y también registros para funciones especiales, dispuestas según se muestra en la figura 1.2. Los registros de propósito general guardan valores como variables, contadores o banderas, datos de los cuales se vale el programador para conseguir la funcionalidad del programa. Los registros para funciones especiales contienen resultados o sirven para controlar módulos del micro como, por ejemplo, el convertor analógico-digital o el puerto serial. Estos registros tienen nombres especiales según a que elemento del PIC correspondan, el valor del TIMER 1 está en el registro TMR1, se configura la interrupción a través del INTCON, etc.

File Address		File Address		File Address		File Address			
Indirect addr. ⁽¹⁾	00h	Indirect addr. ⁽¹⁾	80h	Indirect addr. ⁽¹⁾	100h	Indirect addr. ⁽¹⁾	180h		
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h		
PCL	02h	PCL	82h	PCL	102h	PCL	182h		
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h		
FSR	04h	FSR	84h	FSR	104h	FSR	184h		
PORTA	05h	TRISA	85h		105h		185h		
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h		
PORTC	07h	TRISC	87h		107h		187h		
PORTD ⁽²⁾	08h	TRISD ⁽²⁾	88h		108h		188h		
PORTE ⁽²⁾	09h	TRISE ⁽²⁾	89h		109h		189h		
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah		
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh		
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch		
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh		
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽¹⁾	18Eh		
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽¹⁾	18Fh		
T1CON	10h		90h		110h		190h		
TMR2	11h		91h						
T2CON	12h	PR2	92h						
	13h		93h						
	14h		94h						
CCPR1L	15h		95h						
CCPR1H	16h		96h						
CCP1CON	17h		97h						
RCSTA	18h	TXSTA	98h						
TXREG	19h	SPBRG	99h						
RCREG	1Ah		9Ah						
	1Bh		9Bh						
	1Ch		9Ch						
	1Dh		9Dh						
ADRESH	1Eh	ADRESL	9Eh						
ADCON0	1Fh	ADCON1	9Fh						
General Purpose Register 96 Bytes	20h	General Purpose Register 32 Bytes	A0h	accesses 20h-7Fh	120h	accesses A0h - BFh	1A0h		
			BFh						1BFh
			C0h						1C0h
			EFh	accesses 70h-7Fh	16Fh	accesses 70h-7Fh	1EFh		
			F0h					170h	1F0h
	7Fh	accesses 70h-7Fh	FFh		17Fh		1FFh		
Bank 0		Bank 1		Bank 2		Bank 3			

Figura 1.2. Memoria de Datos.

Fuente: Hoja de Datos 16F871.

1.2.6. Memoria EEPROM.

Se define como una memoria solo de lectura programable y borrrable solo eléctricamente. Cuando se retira la alimentación del PIC los datos almacenados en este espacio de memoria no se pierden. Se dispone de 64 bytes de EEPROM para guardar datos estáticos (constantes) para todos los procesos que el micro tenga que ejecutar. Este espacio es muy importante cuando se tiene que guardar datos que no dependan del avance cronológico del programa por ejemplo una clave de acceso o un número telefónico constante.

1.2.7. Reloj del Sistema.

Para funcionar el micro necesita un reloj que marque el ritmo o la velocidad de proceso, define el tiempo específico en el que se realizan todas las instrucciones. Este reloj tiene varios modos de funcionamiento. Tres modos incluyen una configuración con cristal o un resonador cerámico conectado entre los pines OSC1 y OCS2, además de dos capacitores tal como se muestra en la figura 1.3.

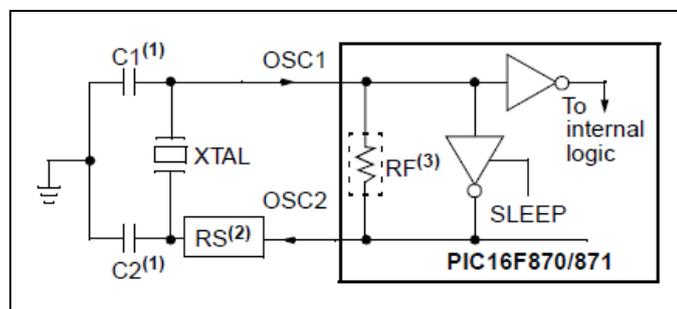


Figura 1.3. Configuración del Oscilador externo.

Fuente: Hoja de Datos 16F871.

Para *Low Power Cristal* (LP), *Cristal Resonator* (XT) y *High Speed* (HS) el fabricante recomienda los valores de capacitores según las frecuencias de los cristales que se deseen utilizar, estos valores se muestran en la Tabla 1.1.a y los Cristales en la Tabla 1.1.b.

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

Tabla 1.1. a. Capacitores según la Frecuencia de los Cristales.

Fuente: Hoja de Datos 16F871.

Crystals Used		
32 kHz	Epson C-001R32.768K-A	± 20 PPM
200 kHz	STD XTL 200.000 kHz	± 20 PPM
1 MHz	ECS ECS-10-13-1	± 50 PPM
4 MHz	ECS ECS-40-20-1	± 50 PPM
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM

Tabla 1.1.b. Cristales Utilizados.

Fuente: Hoja de Datos 16F871.

El último modo de funcionamiento incluye una configuración tipo RC que se muestra en la figura 1.4. El ciclo de reloj está definido por la combinación de Resistencia y Capacitancia, también es función de la fuente de alimentación y la temperatura.

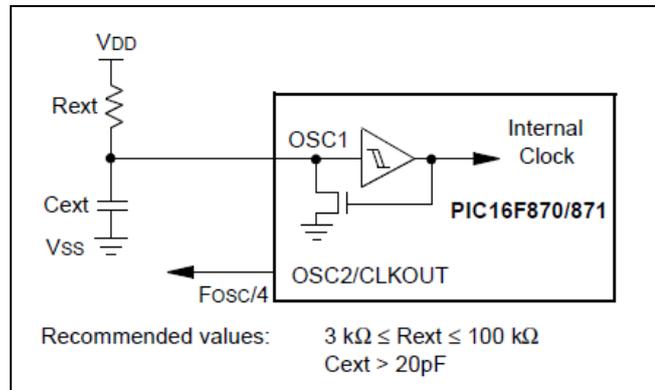


Figura 1.4. Conexión RC a la entrada del Oscilador.

Fuente: Hoja de Datos 16F871.

1.3. Puertos de Entrada y Salida.

Varios de los pines del micro son puertos de entrada salida, pero adicionalmente algunos están multiplexados con funciones especiales de los periféricos del dispositivo. Un puerto se configura a través de sus registros TRIS, por ejemplo, si el TRISA es llevado a 1 todo el puerto se programa como entrada y viceversa.

El puerto A tiene 6 bits bidireccionales (entrada o salida), RA0, RA1, RA2, RA3 y RA5 son además entradas del modulo de conversión analógico digital y el pin RA4 esta multiplexado con el *TIMER 0*. El puerto B tiene 8 entradas o salidas digitales, además RB0 esta multiplexado con la entrada de la interrupción externa. El puerto C está compuesto por 8 entradas o salidas digitales, RC0 y RC1 son también las entradas del reloj externo, RC2 es una entrada o salida del modulo CCP, RC6 y RC7 son la salida y la entrada del modulo serial, respectivamente. RD0 a RD7 son entradas y salidas digitales.

1.4. Circuitos Auxiliares e Interfaces.

Para un funcionamiento óptimo de todos los módulos del PIC es necesario que todos ellos tengan el interfaz adecuado, para ello es muy importante tener en cuenta las recomendaciones del fabricante, en este caso Microchip, así como la aplicación específica en la que utilizarán en el presente trabajo.

1.4.1. Fuente de Alimentación.

En la figura 1.5. Se muestra la fuente de alimentación conmutable propuesta, el punto etiquetado VC es de donde proviene la tensión, por ello este punto tiene dos posibles alimentadores, seleccionados, a través de un relé. La conmutación se realiza entre un adaptador, nombrado ADP, y una batería, BAT, que serán conectados a los contactos normalmente abierto y normalmente cerrado, respectivamente. La bobina del relé está conectada directamente a la alimentación del adaptador. Al no funcionar el adaptador, si fuera el caso, tampoco lo hará el relé por lo que la fuente estará alimentada desde la batería.

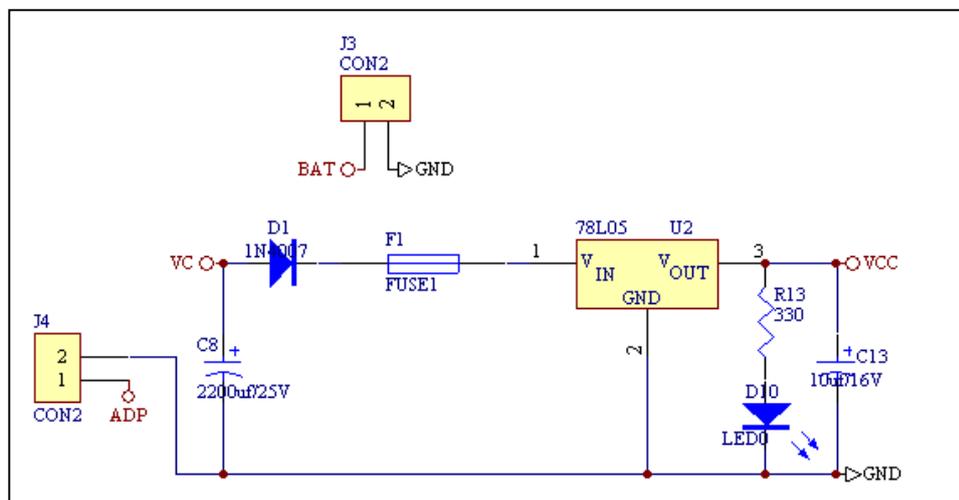


Figura 1.5. Fuente de alimentación.

Fuente: Autores.

1.4.2. Pines Del Microcontrolador.

Como se indicó en la figura 1.3. El oscilador externo se conecta a un cristal de 16Mhz que da la señal de reloj al PIC entre los pines 13 y 14 y establece un ciclo de maquina igual a 62.5 nano segundos. Los pines 11 y 32 son entradas de alimentación positiva así como 12 y 31 son de referencia o tierra. El pin número 1 está conectado a través de un pulsante a un uno lógico que resetea los procesos.

Los pines 2, 3 y 4 son salidas que sirven para comandar al ISD (figura 1.6). En el capitulo siguiente se considerará todo acerca del funcionamiento del ISD2560 al igual que su interacción con el Micro y las líneas de control necesarias. Los pines 5 y 40 son salidas que conmutan la línea telefónica hacia el Decodificador o desde el ISD, a través de relés dobles, debido a que son dos líneas las que se deben de cambiar.

Los pines 6, 7 y 15 se utilizan para controlar el LCD. Los pines 33 al 38 (TR1 – TR6 según la figura 1.6.) son salidas que disparan relés que pueden controlar diferentes actuadores. Las patillas 8 y 9 son salidas que encienden un LED de dos diferentes colores según el estado del sistema. El pin número 10 sirve de salida que encenderá una sirena en caso de alarma. Los pines 19 al 22 y 27 al 30 son salidas digitales que llevan el dato a visualizarse en el LCD. Los pines 25 y 26 son la entrada y salida del puerto serial y están conectados con el conector DB9. Los pines 16, 17, 18, 23 y 24 sirven para interactuar con el decodificador MT8870.

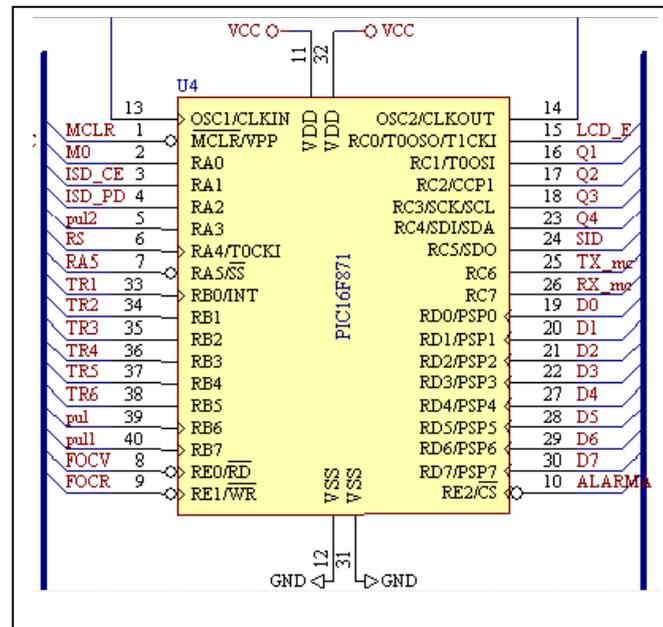


Figura 1.6. Pines del PIC.

Fuente: Autores.

1.4.3. Circuitos Hacia los Actuadores.

Seis salidas digitales del micro controlador están conectadas, a través de una resistencia, con la base de un transistor que dispara en su colector a un relé tal como se muestra en la figura 1.7. Cabe recalcar que los contactos auxiliares de los relés son independientes a sus bobinas, por lo tanto los actuadores que se encienden a través de ellos pueden estar alimentados por CC o CA. La toma común y la normalmente abierta están conectadas con borneras para ser cableadas fácilmente. Figura 1.8.

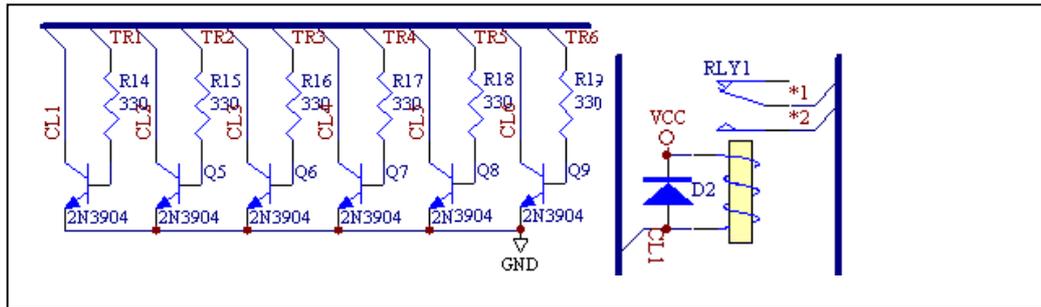


Figura 1.7. Disparo de los Relés.

Fuente: Autores.

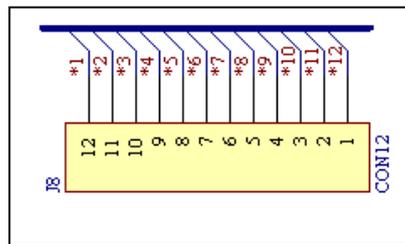


Figura 1.8. Bornera para los Actuadores

Fuente: Autores.

1.4.4. Interfaz del Puerto Serial.

Las líneas del PIC destinadas a la comunicación serial no son compatibles con la PC debido a la diferencia de la tensión con la que ambos dispositivos trabajan. El circuito mostrado en la figura 1.9. Se utiliza para convertir el voltaje de las líneas del PIC en voltaje compatible con la computadora y está basado en un MAX232 que adecúa la señal para la interacción. Las líneas de transmisión-recepción están conectadas a un conector DB9, típicamente utilizado para comunicación serial, presente en casi cualquier computadora convencional.

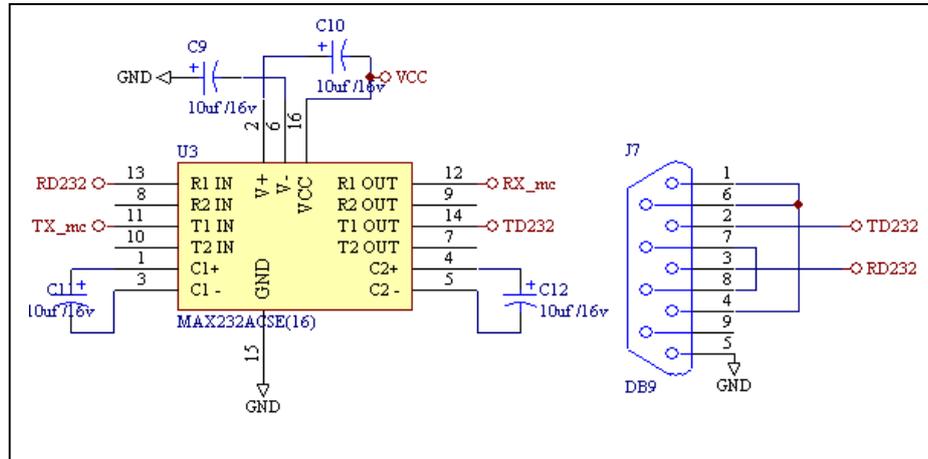


Figura 1.9. Interfaz Serial.

Fuente: Autores.

1.4.5. Conexión del LCD.

Los pines 1 y 2 del LCD son la alimentación negativa y positiva, respectivamente. El pin 3, VL, es una entrada de voltaje variable que controla el contraste del indicador y para el cual se utilizará un potenciómetro conectado directamente entre el voltaje 5 y 0V. El pin número 4 es el control del modo dato ó modo comando y está conectado con el PIC para ser manejado desde el programa. El quinto pin es una entrada digital que controla la lectura o escritura del dispositivo. Para los fines que se persiguen, no es necesario leer el LCD por lo que siempre se mantiene en escritura. Para cada acción del *display* es necesario un pulso que habilite la instrucción, esta es la función del pin 6. Los pines 7 al 14 son las entradas que contienen dato a visualizarse en el LCD. Los pines restantes, 15 y 16 son ánodo y cátodo de la luz de fondo del indicador (*backlight*).

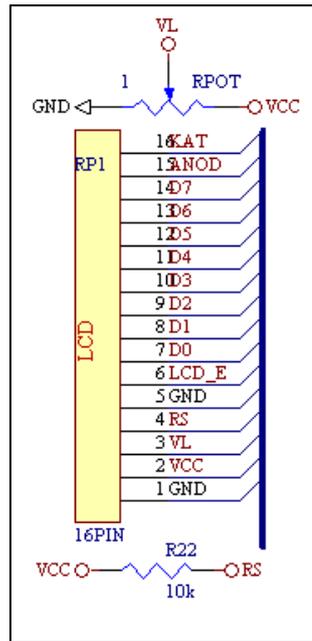


Figura 1.10. Conexión LCD

Fuente: Autores.

1.5. Comunicación Serial.

Para realizar la comunicación entre dos dispositivos es muy común y conocido el protocolo serial. Actualmente se incluye un puerto serial en casi todas las computadoras de escritorio y el estándar utilizado es el RS-232. La transmisión es muy sencilla, se realiza enviando y recibiendo bytes un bit a la vez, y aún cuando esto significa menor velocidad que los protocolos que transmiten un byte completo por vez también significa mayor fiabilidad cuando se comunican dispositivos a mayores distancias, la transmisión en paralelo establece no más de 20 metros para el cable, con no más de 2 metros entre dos dispositivos mientras que un cable serial puede llegar a 1200 metros.

Típicamente, el protocolo serial se utiliza para transmitir datos en formato ASCII. Para establecer la comunicación se necesitan 3 líneas: tierra, línea de transmisión y línea de

recepción. Debido a que la comunicación se realiza de manera asincrónica es posible recibir por una línea mientras se transmite por otra.

1.5.1. Velocidad de Transmisión Serial.

La velocidad de transmisión es uno de los parámetros más importantes para establecer la comunicación entre dos dispositivos. La velocidad se refiere a la cantidad de información que se puede transmitir por segundo, esta velocidad se mide en baudios, por ejemplo, la velocidad máxima en un computador normal son 115200 baudios. Los ordenadores tienen establecidas varias velocidades seleccionables, pero un micro controlador debe configurar la comunicación a través de un registro (SPBRG) de 8 bits, tal como lo describe la Ecuación 1.1.

$$\text{Baud rate} = \frac{F_{osc}}{16(x + 1)}$$

Ecuación 1.1.

Dentro de los límites que establece la expresión se puede elegir la velocidad que uno desee, esto si se comunican dos dispositivos iguales, si se quiere transmitir hacia una computadora se tiene que seleccionar la velocidad más aproximada a la elegida en el computador.

Para un correcto funcionamiento del sistema no es necesario alta velocidad de transmisión por lo que se utilizará una baja velocidad para garantizar la menor cantidad de errores posible esto es 9600 baudios.

1.5.2. Trama del Dato Serial.

Cada dato transmitido o recibido a través del protocolo RS-232 se compone de la siguiente forma, un bit de *start* que marca el inicio de cada trama de datos. A continuación viene la información en 5, 7 ó 8 bits. El número de datos que se transfiere

depende del tipo de información que se maneja, por ejemplo, el ASCII estándar tiene un rango de 0 a 127, es decir, 7 bits; para ASCII extendido son 0 a 255 por lo que son necesarios 8 bits. Después de la información están los bit de stop que indican el fin de un paquete de la trama y puede ser 1, 1.5. o 2 bits de stop. Como cada dispositivo tiene su reloj propio es posible que no estén sincronizados por lo que los bits de stop pueden darle cierta tolerancia a esta falta de sincronía. Mientras más sean los bits de stop mayor será la tolerancia, pero también más lenta será la comunicación.

La comunicación se configurará con 1 bit de *start* 7 bits de datos y 1 bit de *stop*.

1.6. Software del Microcontrolador.

Para desarrollar el *software* tal y como se necesita existe una herramienta, adecuada para la tarea, es MPLAB y es gratuito a través de la pagina web de Microchip, es un programa con editor de texto para programar en lenguaje ensamblador, adicionalmente tiene compilador de código y simulador (no en tiempo real). Utilizando la ventana del simulador se puede ver la evolución de los registros y variables del PIC según avanzan las líneas de programa.

Además de las ventajas para desarrollar *software*, MPLAB se desenvuelve en ambiente *Windows* y por tanto presenta las clásicas barras de programa, de menú, de herramientas, de estado etc. El programa también es capaz de interactuar con el hardware Microchip de grabación, *PICSTART*. Para lograr comenzar y grabar un programa en un chip se tienen que seguir los siguientes pasos:

1. Crear un archivo *.ASM* que contendrá el código en lenguaje ensamblador.
2. Crear un proyecto nuevo eligiendo un nombre y ubicación.
3. Agregar el fichero *.ASM* como un *SOURCE FILE*.
4. Elegir el micro a utilizar desde *SELECT DEVICE* del menú *CONFIGURE*.

A partir de este punto se comienza a desarrollar el programa en lenguaje ensamblador, en el espacio del texto del archivo *.ASM*.

1.6.1. Set de Instrucciones.

La familia media 16F de micro controladores son de tipo RISC (*Reduced Instructions Set Computer*) esto se debe a que para programar el dispositivo se tienen disponibles únicamente 35 comandos en total divididos según su función. Existen instrucciones orientados a afectar registros enteros, también hay los que actúan sobre bits específicos dentro de los registros y los que están se usan en relación a los literales o banderas que declara el programador.

En la tabla 1.2. Se muestran la sintaxis de los comandos orientados a registros seguidos de su explicación y los operandos que utiliza. En la tercera columna se muestra el bit que se altera dependiendo del resultado de la instrucción ejecutada.

El comando NOP es simplemente una demora que tarda un ciclo de máquina (62.5 nano segundos) y existe debido a que algunos procesos necesitan de tal demora para poder ejecutarse de manera correcta. No modifican ningún tipo de registro ni parcial ni totalmente por lo que en realidad no se debe encasillar dentro de los tres tipos de instrucciones.

OPERACIONES ORIENTADAS A REGISTROS		
Sintaxis	Operación	Estados afectados
ADDWF f,d	Sumar W y f	C,DC,Z
ANDWF f,d	AND entre W y f	Z
CLRF F	Limpiar f	Z
CLRW	Limpiar W	Z
COMF f,d	Complementar f	Z
DECF f,d	Decrementar f	Z
DECFSZ f,d	Decrementar f, saltar si cero	
INCF f,d	Incrementar f	Z
INCFSZ f,d	Incrementar f, saltar si cero	
IORWF f,d	OR entre W y f	Z
MOVF f,d	Mover f	Z
MOVWF F	Mover W a F	
NOP	No Operación	
RLF f,d	Rotar a la izquierda a través del carry	C
RRF f,d	Rotar a la derecha a través del carry	C
SUBWF f,d	Restar W de f	C,DC,Z
SWAPF f,d	Intercambiar nibbles de f	
XORWF f,d	OR exclusiva entre W y f	Z

Tabla 1.2. Instrucciones orientadas a Registros

Fuente: r-luis.xbot.es

En la tabla 1.3. Se muestran las operaciones que se utilizan para cambiar un registro, pero bit a bit, se especifica su expresión y la acción que realiza. Algunos de ellos pueden utilizarse para hacer selecciones, dependiendo del estado del bit que se esté evaluando.

OPERACIONES ORIENTADAS A BITS		
Sintaxis	Operación	Estados afectados
BCF f,b	Limpiar bit b de f	
BSF f,b	Activar bit b de f	
BTFSC f,b	Probar bit b de f, saltar si cero	
BTFSS f,b	Probar bit b de f, saltar si uno	

Tabla 1.3. Instrucciones orientadas a Bits

Fuente: r-luis.xbot.es

En la tabla 1.4. Se ilustran los comandos que se relacionan con las banderas o registros de propósito general dentro del programa. Además de alterar el contenido de los literales también sirven para desplazarse dentro del programa y entre subrutinas.

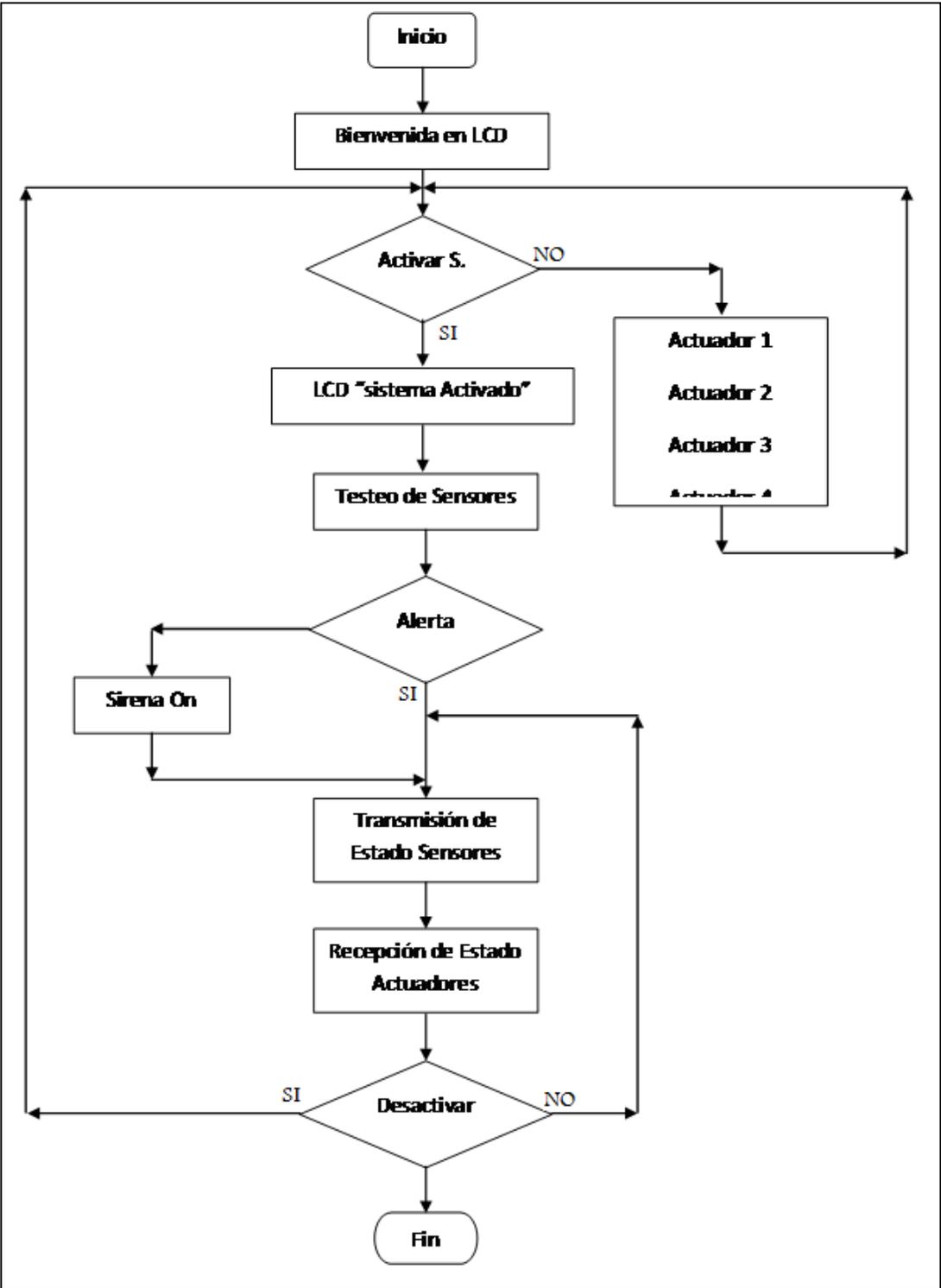
OPERACIONES ORIENTADAS A LITERALES			
Nemotécnicos		Operación	Estados afectados
ADDLW	k	Sumar literal k a W	C,DC,Z
ANDLW	k	AND entre k y W	Z
CALL	k	Llamar subrutina	
CLRWDT		Limpiar WDT	-TO,-TD
GOTO	K	Salta a dirección k	
IORLW	K	OR entre k y W	Z
MOVLW	K	Cargar W con literal k	
RETFIE		Retornar de interrupción	
RETLW	K	Retornar y cargar W con k	
RETURN		Retornar de subrutina	
SLEEP		Ir al modo de bajo consumo	-TO,-TD
SUBLW	K	Restarle k a W	C,DC,Z
XORLW	K	OR exclusiva entre k y W	Z

Tabla 1.4. Instrucciones orientadas a Literales.

Fuente: r-luis.xbot.es

En el anexo C se muestra el código completo para el micro controlador que administra el sistema y que se configura según las especificaciones de hardware que se describen en este capítulo.

1.6.2. Diagrama de Flujo.



Fuente: Autores.

1.6.3. Funcionamiento.

En principio el sistema debe presentar un mensaje de bienvenida a través del LCD y al mismo tiempo se encuentra en estatus de *stand by*, esperando por un cambio de estado, la siguiente opción debe ser: activar alarma, cambio de estado de actuador 1, actuador 2, actuador 3 actuador 4 ó cambiar clave. Controlar los actuadores o cambiar la clave se maneja con menús sucesivos activados a través del teclado del teléfono, habilitado después de descolgar el auricular. Si se elige cambiar la clave se ingresa en un menú que solicita el ingreso de la clave actual, la cual está grabada en la EEPROM del PIC, de realizarse el cambio la nueva clave también se registra en la memoria no volátil. Después de cualquiera de estas opciones el sistema vuelve al *stand by*.

Si se elige activar alarma el primer proceso es indicar el estado en el *Display* a continuación se comienza un testeo de los sensores, después de periódicamente el PIC debe enviar, vía puerto serial, un informe de estado de los sensores y también de los actuadores. Terminada la transmisión se debe recibir el dato de un posible cambio de estado, en los actuadores, desde la computadora. Luego de estos procesos se testea la línea telefónica, pues desde ella se pueden comandar cambios en los actuadores, inicio de secuencias ó desactivación del sistema. Luego se vuelve a testear los sensores, es decir se vuelve un lazo cerrado.

De producirse una alarma se activa el relé que enciende la sirena. También se inicia una secuencia de acciones que primero: busca en la memoria EEPROM una combinación de números, luego los codifica y los pone en la línea telefónica a través del ISD2560, espera por la respuesta y al ser contestada la llamada reproduce el mensaje de alerta, finalmente cierra la línea telefónica, para esperar la llamada que desactive la alarma.

CAPITULO II

LÍNEA TELEFÓNICA (DTMF)

2.1. Introducción.

Debido a la amplia difusión de las redes telefónicas en nuestro medio, y en general a nivel mundial, este se vuelve un elemento ideal para lograr transmisiones de información con alcance global. Pero para aprovechar debidamente el potencial de la línea es necesario conocer bien las características eléctricas y sus modos de funcionamiento.

En primer lugar se debe mencionar que el par telefónico, eléctricamente, se comporta de la siguiente forma; en reposo, es decir sin que ninguna llamada este en proceso, el circuito tiene una tensión presente de entre 48 y 52 voltios de corriente continua permitiendo el paso de apenas 50uA, cuando se presenta una petición de llamada, cuando se ha discado un número y se produce el ring, se genera una tensión alterna sobrepuesta a la continua con un valor aproximado de 100 voltios pico-pico a una frecuencia de entre 25 y 50 Hz. Si se atiende la llamada el aparato telefónico cierra el circuito con su contraparte y las características cambian; la tensión continua se reduce a un rango de 5 a 7 voltios con una corriente similar a 20 mA y la corriente alterna desaparece por completo.

Si se toma en cuenta las condiciones iniciales es evidente que la exposición de un PIC, y por consiguiente de cualquier circuito electrónico TTL (tensión de funcionamiento de 5 voltios), a los niveles de señal de la línea telefónica en reposo producirá seguramente el desgaste de los componentes y que posteriormente se estropeen. Para lograr un funcionamiento adecuado con el DAQ es primordial mantener aisladas todas las partes del circuito de las corrientes nocivas al mismo. Por su puesto lo anterior se traduce en la conmutación del acceso a la línea telefónica para evitar el contacto con la misma siempre que no haya una llama establecida.

Una vez que se logren controlar los excesos de alimentación y corrientes parasitas se debe lidiar con la interacción entre los circuitos y el canal telefónico, esto se refiere tanto a la generación de llamadas hacia otros terminales como a atender las llamadas entrantes. Estas acciones se logran a través de la técnica llamada DTMF por su definición en ingles *DUAL TONE MULTI FREQUENCY*.

DTMF codifica los elementos de un teclado matricial de 4x4 alfanumérico utilizando una señal compuesta por dos tonos de diferente frecuencia. Una fila de 4 tonos, de alta frecuencia, combinada con una columna de 4 elementos, de baja frecuencia, forman una configuración DTMF como se muestra en la tabla 2.1. De esta forma en la línea se introduce solo una señal a la vez y puede ser identificada plenamente, sin error entre caracteres.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Tabla 2.1. Matriz DTMF.

Fuente: wikipedia.com

Este método se utiliza tanto en la generación del código de marcado, para llamar, como en la identificación de los dígitos que se reciben, esto a fin de validar una clave de comprobación que de acceso al control de todo el sistema de forma remota. Las dos funciones, recepción y transmisión, se realizan por diferentes elementos dentro del circuito y aplican el mismo principio, pero se valen de medios muy diferentes uno del otro.

2.2. Recepción con el MT 8870.

Para realizar la recepción de las llamadas, en principio, es necesario que el dispositivo se encargue de conmutar los canales que conectan la línea telefónica con el ingreso de señal al decodificador

En la figura 2.1. Se muestra el circuito que realiza el cambio para acceder a la línea telefónica. Desde una salida del micro controlador, etiquetado pul1, con un estado alto se dispara el transistor Q2 mismo que energiza a la bobina de K1. Los contactos auxiliares 4 y 1 cierran el circuito poniendo en paralelo a la línea telefónica (puntos "line" y "line1") con una resistencia de 560 ohmios, componente ideal para simular un teléfono descolgado. Esta acción engaña a la central telefónica haciéndole creer que hay un teléfono, ya sea contestando u originando una llamada telefónica.

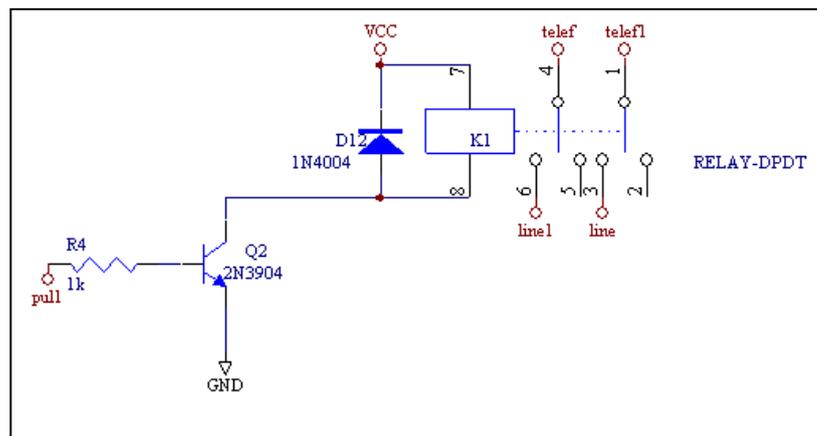


Figura 2.1. Conmutador de la línea telefónica.

Fuente: Autores.

La conmutación se realiza cuando el PIC cuenta cierto número de *rings*. Cada vez que se produce una llamada, por cada timbre, el opto acoplador U1 (figura 2.2.) manda un 1 lógico hacia un LED y al mismo tiempo hacia un puerto del Micro configurado como entrada. Dependiendo de el tiempo que se desee que el sistema espere, se programa el número de *rings* 3, 4 5, etc. Una vez que la llamada se establece, el canal se cierra entre los dos terminales telefónicos, por lo tanto cualquier número presionado en

cualquiera de los dos aparatos telefónicos ingresa directamente a los puntos de ingreso de señal del decodificador.

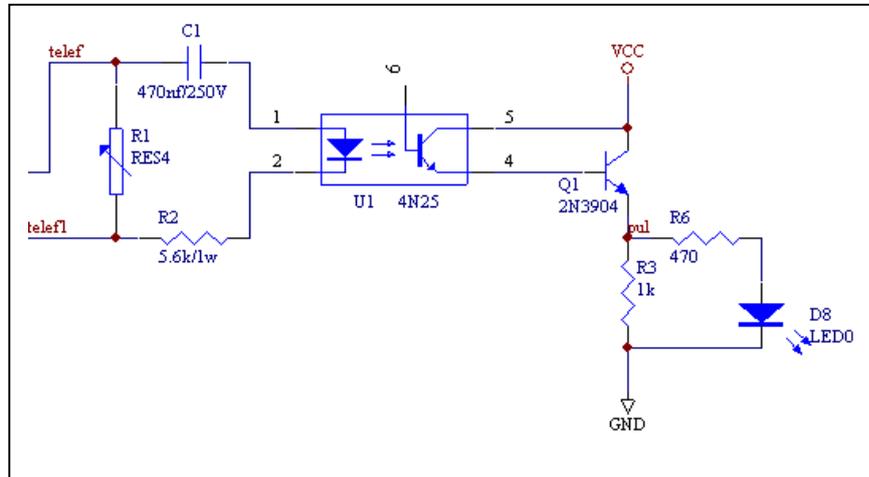


Figura 2.2. Indicador del Ring.

Fuente: Autores.

2.2.1. Función.

Cualquier señal recibida por el decodificador es descompuesta en los tonos que se combinaron para formarla. El símbolo presionado que dio origen a la combinación es identificado y convertido a formato binario de 4 bits, salidas Q1 a Q4. Las salidas funcionan en combinación con los pines TOE, INH y ESt según se indica en la tabla 2.2. En caso de encontrarse INH en estado alto y ESt bajo quedan bloqueadas las teclas que simbolizan las letras A, B, C y D.

Digit	TOE	INH	ESt	Q ₄	Q ₃	Q ₂	Q ₁
ANY	L	X	H	Z	Z	Z	Z
1	H	X	H	0	0	0	1
2	H	X	H	0	0	1	0
3	H	X	H	0	0	1	1
4	H	X	H	0	1	0	0
5	H	X	H	0	1	0	1
6	H	X	H	0	1	1	0
7	H	X	H	0	1	1	1
8	H	X	H	1	0	0	0
9	H	X	H	1	0	0	1
0	H	X	H	1	0	1	0
*	H	X	H	1	0	1	1
#	H	X	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0
A	H	H	L	undetected, the output code will remain the same as the previous detected code			
B	H	H	L				
C	H	H	L				
D	H	H	L				

Tabla 2.2. Posibilidades de los pines del MT8870.

Fuente: Hoja de Datos T8870

La cifra binaria es depositada en el *buffer* de ingreso del PIC formado por 4 puertos de entrada. Una vez en el micro haya recogido todos los números que se hayan marcado los compara con la variable *password* guardada en su memoria EEPROM. En caso de empatar exactamente la clave, el sistema da acceso al control de los actuadores a través de las primeras teclas del teléfono; esto quiere decir tecla 1 enciende el actuador número 1, tecla 2 actuador 2, etc.

2.2.2. Estructura Interna.

En el diagrama de bloques de la figura 2.3. Se ilustra el orden en el que se realizan los procesos necesarios para la decodificación de las señales dentro del 8870. Absolutamente necesarios son la alimentación y la señal de reloj, pero estos bloques no están relacionados con la decodificación, sino que son requisitos de funcionamiento.

A continuación del ingreso, la señal pasa a través de un grupo de filtros que la dividen en dos tonos, uno de alta y otro de baja frecuencia. Luego ambos tonos atraviesan un bloque en el que se detecta los cruces con cero, en el siguiente bloque los tonos entran al algoritmo digital que identifica el número discado, posteriormente se necesita convertir el número a formato digital y enviarlo a los 4 *buffers*, por su puesto la salida solo estará presente si se habilitan los *buffers* (pin TOE).

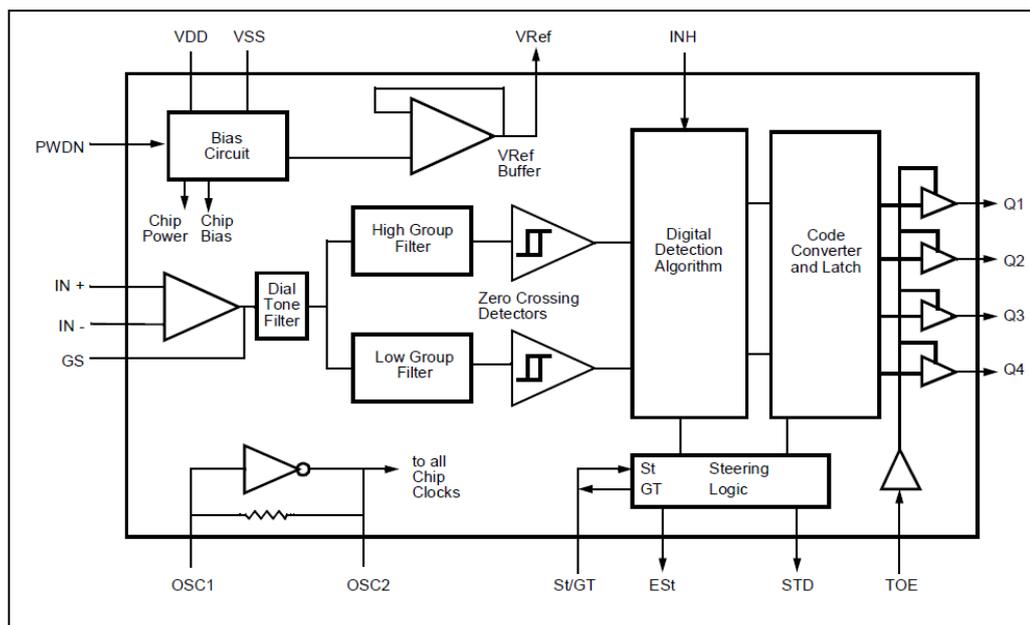


Figura 2.3. Diagrama de Bloques MT8870.

Fuente: Hoja de Datos MT8870.

Todos los pines del encapsulado de 18 pines se muestran en la figura 2.4. y corresponden a la tecnología *through hole*. Imagen rescatada de la hoja de datos que publica el fabricante del *chip*.

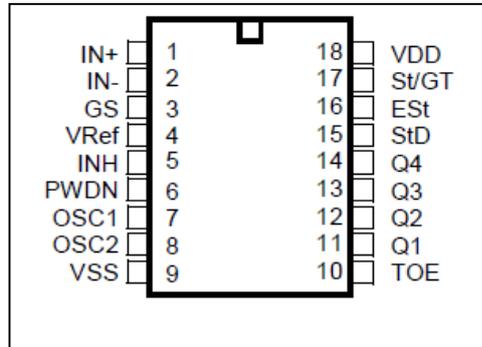


Figura 2.4. MT8870 Encapsulado 18 pines.

Fuente: Hoja de Datos MT8870.

2.2.2.1. Resumen de Pines.

IN+ Entrada no inversora del amplificador instrumental interno.

IN- Entrada inversora del amplificador instrumental interno.

GS control de la ganancia del amplificador instrumental,

VRef Voltaje de referencia del Circuito.

INH Inhibidor de detección de teclado alfanumérico.

PWDN Activa el modo *Power Down* de bajo consumo (chip inactivo).

OSC1 Entrada 1 de reloj externo.

OSC2 Entrada 2 de reloj externo.

VSS Alimentación Negativa-

TOE Habilita las 4 salidas Q1 a Q4.

Q1 – Q4 Salidas digitales correspondientes a los tonos recibidos.

StD Salida que indica cuando se ha decodificado un par de tonos.

Est Salida que indica cuando se ha detectado un par de tonos valido.

St/GT Un voltaje alto registra un par de tonos y actualiza la salida. Un voltaje bajo libera al chip para aceptar un nuevo par de tonos.

VDD Alimentación positiva.

2.2.3. Interfaz con la Línea Telefónica.

El circuito mostrado en la figura 2.5. Muestra la conexión de todos los pines del 8870 junto con los componentes auxiliares necesarios. Los valores de resistencia y capacitancia de los componentes son típicos y recomendados por el fabricante del Integrado. Como se indica en el grafico el cristal es de baja frecuencia con respecto al cristal del PIC esto se debe a que son menos los procesos que realiza el decodificador, no así el micro debe desempeñar muchas más tareas incluso relacionadas con la línea telefónica. Anteriormente se hizo referencia a las figuras 2.1. y 2.2. que ilustran los circuitos complementarios del decodificador y se explica el funcionamiento de los mismos.

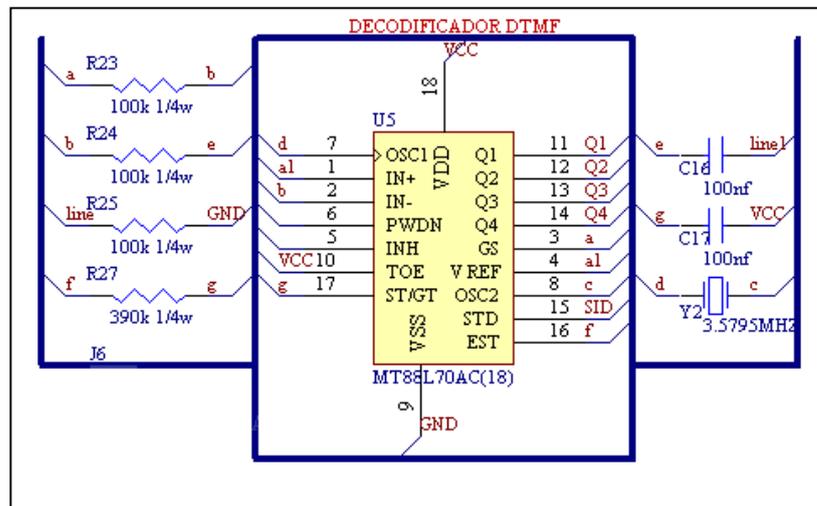


Figura 2.5. Interfaz con la Línea Telefónica.

Fuente: Autores.

2.2.4. Interacción con el PIC.

Hay varios pines con los que el micro puede valerse para saber cuál es el estado del decodificador, pero el realmente importante es StD. La información que se envía al PIC es precedida de un uno lógico en el pin StD que llega directamente a un puerto de entrada digital del micro y paralelamente a un indicador luminoso que indica de cuantos

datos ingresan. Cada vez que el micro reciba el dato binario desde Q1-Q4 debe guardarlo en una variable y estar listo a recibir el siguiente aviso de dato.

Los otros pines que pueden alertar del estado del 8870 son ESt y St/Gt, la primera es la salida que indica con un estado alto que se ha recibido un par de tonos valido para la conversión y finalmente St/Gt es una salida con dos estados posibles. Estado alto: quiere decir que un par de tonos ingresó y el que se encontrada procesándose debe ser llevado a las salidas. Estado bajo: es indicación de que el decodificador está listo para recibir el siguiente par de tonos.

2.3. Transmisión con el ISD.

Una vez que se ha producido una alerta en el sistema, una de las tareas que se deben realizar es establecer una llamada telefónica. Lo ideal para realizar el enlace sería usar un generador DTMF de la familia MT88XX, pero actualmente no se cuenta con ese recurso tecnológico por lo que se reemplaza dicho generador con un circuito integrado grabador-reproductor en el cual se hayan cargado previamente los tonos correspondientes a las teclas de un teléfono, para esto se debe grabar la señal desde la línea telefónica usando un teléfono.

Los chips de este tipo, disponibles en el mercado local, tienen un rango de tiempo de grabación de 60 hasta 120 segundos. Evidentemente los 10 números almacenados no ocupan todo el espacio de memoria por lo que este espacio se aprovecha para guardar los mensajes que se van a transmitir por la línea, por ejemplo, "alerta", "alarma violentada" o quizás "Zona segura".

Para cada tarea tal como grabar o reproducir en sus diferentes modos de funcionamiento un ISD2560 requiere de un circuito auxiliar por lo que hace falta una configuración universal. Como un dispositivo independiente (tarjeta de pruebas del ISD), se plantea un circuito que se puede utilizar para grabar el Integrado con una entrada de micrófono o con una entrada de señal desde un conector tipo audífono. Adicionalmente el circuito debe ser capaz de funcionar de los 6 modos posibles y de los cuales se tratara más adelante. Aunque el uso de este chip es específico dentro

del presente proyecto, es muy versátil por sus características, algunas de las cuales se describen a continuación.

- Fácil grabación/reproducción.
- Alta calidad de reproducción de voz/audio.
- Operación manual o mediante un Micro controlador.
- Duración máxima 60 segundos.
- Operación en modo de bajo consumo.
- Almacenamiento en memoria no volátil.
- Direccional para manejar múltiples mensajes.
- 100 años de retención en la memoria.
- Fuente de reloj interno.
- Rango de funcionamiento TTL.

2.3.1. Función.

Inicialmente se debe conmutar la línea telefónica para que su acceso se conecte con la salida del ISD. Dentro del circuito del DAQ estará incluida una configuración de reproducción a través de direccionado de *slots*, esto quiere decir que salta entre los espacios de memoria para armar un mensaje con direcciones no sucesivas. Al reproducir los tonos grabados correspondientes a los números guardados en la EEPROM del micro se produce la llamada hacia el o los terminales dispuesto en la programación del PIC. El PIC a través de sus puertos comanda al ISD.

Después de establecer la petición de llamada o ring el sistema espera por la atención a la misma. El mensaje que se libera en la línea telefónica debe ser coherente con la condición del sistema. Una vez informado el estatus del dispositivo se termina la llamada y se espera por la siguiente que posiblemente desactive la alarma o cambie el estado de los actuadores.

2.3.1.1. Modos de Funcionamiento.

M0 *Message Cueing* permite saltar entre los mensajes sin la necesidad de saber la dirección física de cada uno. Funciona con cada 0 enviado al pin 23 CE.

M1 *Delete EOM Markers* este modo de funcionamiento sirve para grabar secuencialmente varios mensajes y dejar una sola marca de fin de mensaje, precisamente en el último. Al momento de reproducir todos los mensajes se unen como uno solo continuo.

M2 *Unused* este modo no está disponible.

M3 *Message Looping* cuando se elige este modo de funcionamiento se reproduce repetidamente el mensaje ubicado en el primer espacio de la memoria.

M4 *Consecutive Addressing* permite grabar y reproducir mensajes independientemente debido a que al final de cada uno deja una marca de fin de mensaje.

M5 *CE-Level activated* sirve para reproducir un mensaje repetidamente, pero con la posibilidad de hacer una pausa. El mensaje se reproduce mientras CE está en 0 cuando se lleva a 1 se pausa hasta que vuelva a estar en 0.

M6 *Push Button* este modo permite seleccionar la grabación y la reproducción a través de un *Switch* y el tiempo de cada uno con pulsantes.

2.3.2. Estructura Interna.

En todo el proyecto se utiliza la tecnología *through hold*, en la figura 2.6. Se ve el encapsulado para el ISD compatible con este tipo de montaje. En la tecnología superficial se nota diferencias en dimensiones, distribución de pines, etc.

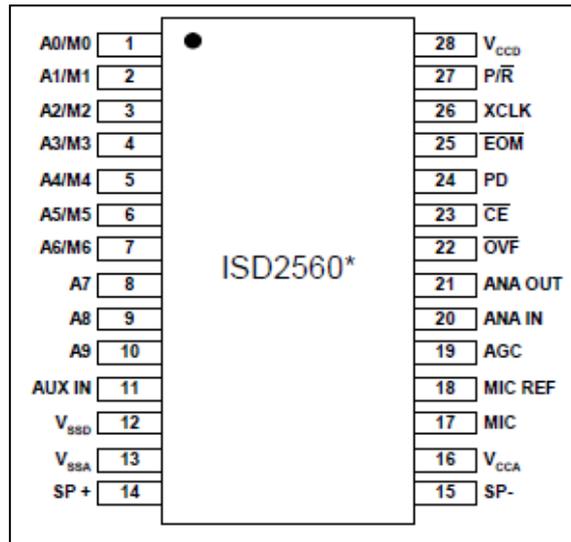


Figura 2.6. Pinaje del ISD2560.

Fuente: Hoja de Datos ISD2560

En la figura 2.7. se ilustra el diagrama de bloques del ISD e indica las partes funcionales del chip. Conectado en la entrada de la señal se encuentra un preamplificador para el uso de un micrófono y a través de una resistencia externa se puede controlar su ganancia. A continuación la señal pasa por un filtro activo anti alias, la señal llega a un arreglo de 480K de memoria no volátil, este arreglo es direccionable externamente. Al salir la señal pasa por un filtro que la suaviza. Por último antes de salir la señal pasa por un amplificador que la adecua para reproducirse en un parlante.

También están los bloques de alimentación que están conectados a los pines de tensiones positivas, y referencias. El bloque de reloj interno que da el ritmo muestreo de la señal también dirige la grabación dentro de la memoria. Si se desea utilizar el reloj externo se debe regir a los datos expuestos en la tabla 2.3. Finalmente el bloque en el que están todos los pines de control para todas las funciones del ISD.

Part Number	Sample Rate	Required Clock
ISD2560	8.0 kHz	1024 kHz
ISD2575	6.4 kHz	819.2 kHz
ISD2590	5.3 kHz	682.7 kHz
ISD25120	4.0 kHz	512 kHz

Tabla 2.3. Frecuencia del Cristal Según ISD utilizado.

Fuente: Hoja de Datos ISD2560

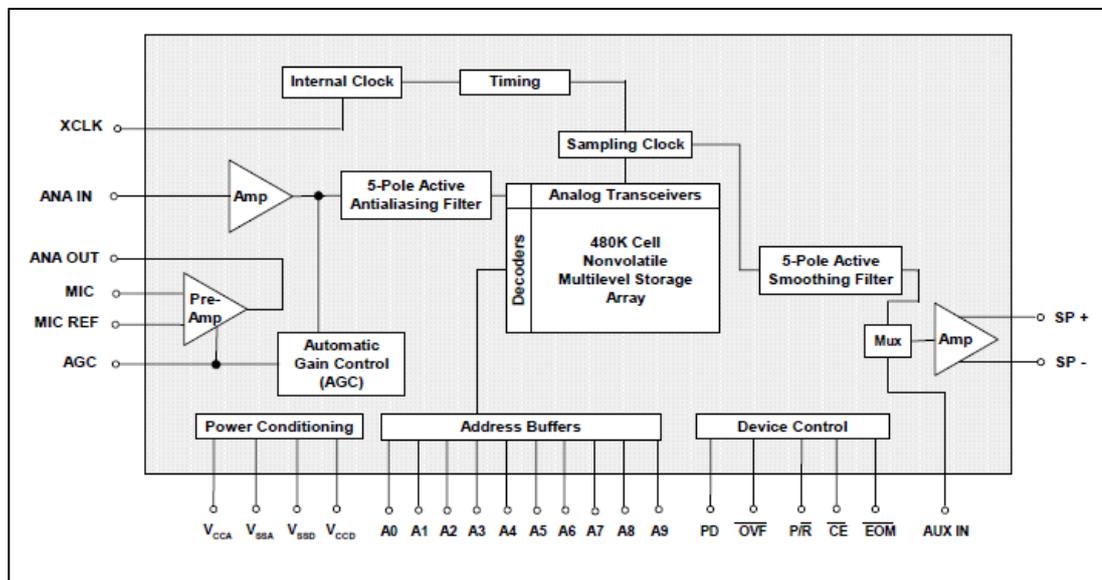


Figura 2.7. Diagrama de Bloques del ISD2560.

Fuente. Hoja de Datos ISD2560.

2.3.2.1. Resumen de Pines.

A0/M0 – A6/M6 estos bits seleccionan el modo de funcionamiento.

A7 junto con A0 –A6 funcionan para direccionar la memoria en caso de utilizar M4.

AUX IN entrada se utiliza para conectar múltiples ISDs en cascada sirve como señal de reproducción del siguiente chip a la salida de parlante del anterior.

VSSA y VSSD el ISD utiliza un bus de tierra digital y uno de analógica separadamente. SP+/SP- salidas positiva y negativa hacia el parlante. Salida máxima de 50mW (Parlante de 16 ohmios).

VCCA y VCCD para minimizar el ruido se utilizan buses separados para la alimentación digital y la analógica.

MIC es la entrada positiva hacia el preamplificador integrado en el ISD.

MIC REF entrada negativa para el micrófono y hacia la referencia del preamplificador.

AGC es el pin de control automático de ganancia (según nivel del micrófono). Los componentes externos sirven de limitación únicamente.

ANA IN transfiere la señal analógica de ingreso.

ANA OUT provee la salida del preamplificador.

OVF *Over Flow* esta salida manda un pulso bajo en caso de que la memoria este llena.

CE *Chip Enable* sensible a los pulsos bajos inicia los procesos de grabado y reproducción, según sea el caso.

PD control del *Power Down* o modo de bajo consumo de energía.

EOM cada vez que se termina un mensaje se inserta una marca en la memoria y al mismo tiempo se provoca un 0 lógico en esta salida.

XCLK es el pin de entrada si se deseara utilizar el reloj externo. Si no se utiliza se debe conectar a tierra.

P/R con un estado alto se selecciona la función reproducción. Con un cero lógico se elige la función grabación

2.3.3. Circuito Auxiliar de Grabación-Reproducción.

El fabricante del chip, por medio de su hoja de datos, propone y recomienda un circuito (figura 2.8.) que sirve para grabar y reproducir de modo *Push-Button* y consta de un *switch* para seleccionar la función y dos pulsantes para controlar los procesos *start/pause* y *stop/reset*. Los valores utilizados en el esquema funcionan de manera eficiente por lo que no hay motivo por el cual se deban cambiar estos valores o esta configuración en un circuito funcional.

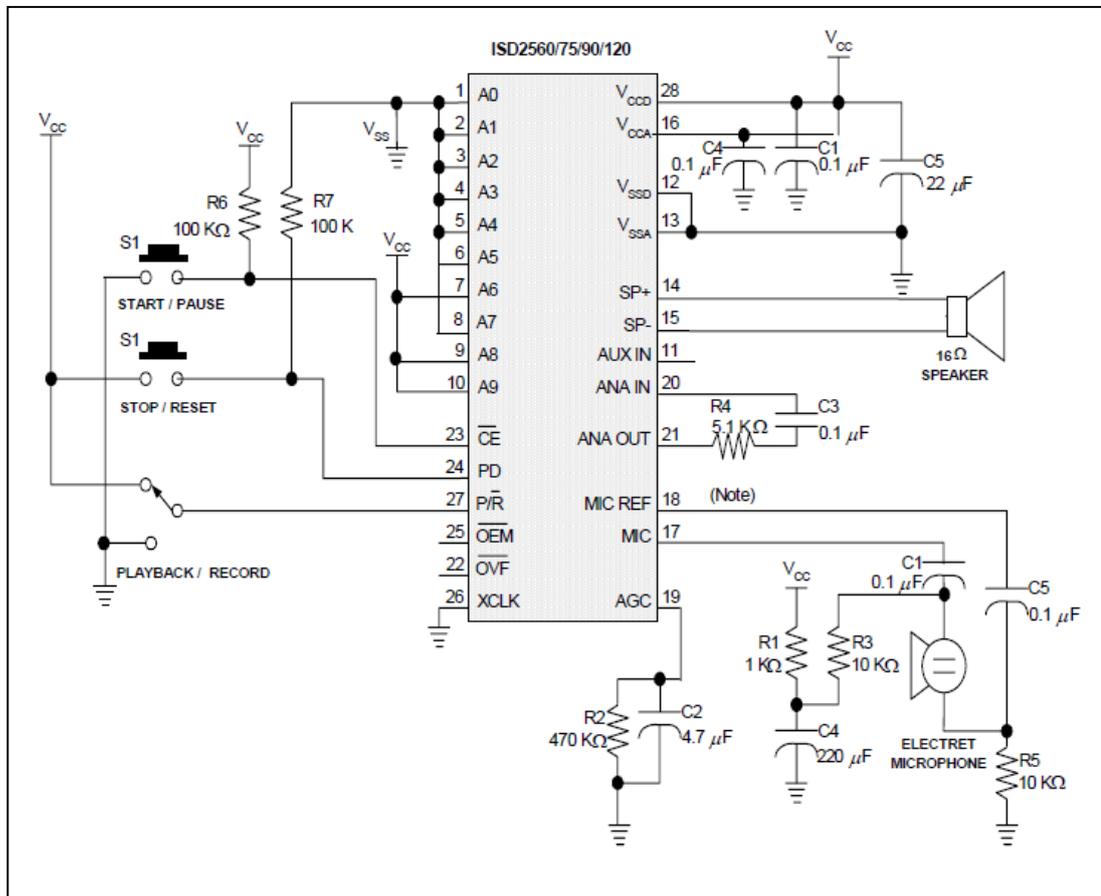


Figura 2.8. Configuración *Push Button*.

Fuente: Hoja de Datos ISD2560.

Para el circuito del DAQ se debe utilizar el modo *Push Button*, pero con algunos cambios como en la figura 2.9. Los controles no deben ser externos vía pulsantes, sino programables con los puertos del micro controlador. Ya no se necesita grabar por lo que no se necesita del *switch* para seleccionar la función y se suprime por completo el micrófono y los componentes auxiliares para su funcionamiento. En esta configuración se necesita utilizar la indicación de EOM. Y utilizar la salida hacia el parlante conectada directamente en la línea telefónica.

La selección del modo funcional y el direccionamiento de la memoria también son comandados por los puertos de salida del micro controlador.

2.3.4. Tarjeta de Pruebas del ISD.

Es posible utilizar un ISD de para reproducir de 6 formas diferentes utilizando el mismo circuito y los mismos valores de componentes. Para la grabación de la memoria del ISD, con los números y también con los mensajes, se debe utilizar la tarjeta de pruebas, misma en la que se puede grabar una y otra vez hasta estar conforme con el volumen, timbre y tono de los mensajes grabados.

La tarjeta se puede configurar de cualquiera de los 6 modos funcionales debido a que cuenta con un *dip-switch* que setea las entradas a voluntad grafico 2.11. Dentro del modo *push-button* se puede elegir la grabación o reproducción con un selector del mismo tipo. En el anexo que contiene el manual de funcionamiento se detalla las posiciones según la función que se desee dar al dispositivo y el modo en el que se quiere trabajar.

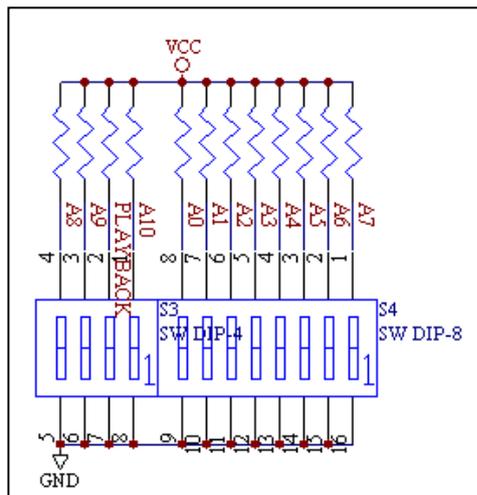


Figura 2.11. Dip –Switch de control del ISD2560

Fuente: Autores.

En la figura 2.12. Se ilustra la conexión del Integrado y los componentes necesarios para lograr una configuración universal así como las salida que tienen la posibilidad de una conexión externa a través de borneras o conectores. El voltaje de alimentación puede ingresar a través de un conector típico de fuente de poder (adaptador) y también cableando desde otro dispositivo mediante borneras.

Las entradas desde el micrófono tienen conectados filtros RC típicos para eliminar la mayor cantidad posible de ruido y por su puesto se accede a él mediante una bornera que también da la posibilidad de conectar cualquier tipo de fuente de audio. Cabe aclarar que el componente idóneo para este circuito es un mini micrófono (*electret*). El resto de los capacitores presentes en el circuito se deben a un filtrado extra a la tensión de alimentación. La tarjeta tiene acceso para conectar la fuente de audio desde la línea telefónica para grabar pares de tonos, ya codificados DTMF, valiéndonos de cualquier teléfono. El diodo D11 es un indicador de funcionamiento de mensaje, permanece encendido solo mientras este reproduciéndose o grabándose un mensaje.

La tarjeta también da la posibilidad de conectarse, mediante borneras, con un PIC para ser controlado a través de él. Se incluyen borneras para conectar un parlante de reproducción o cablear hacia otro dispositivo para el cual el circuito será la fuente de audio.

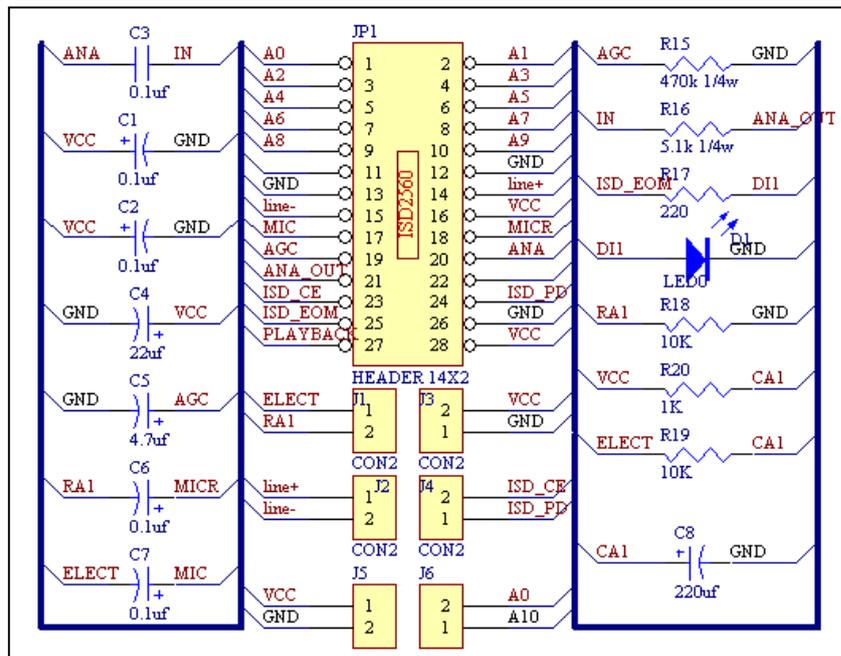


Figura 2.12. Circuito del ISD.

Fuente: Autores.

Los pulsantes mostrados en la figura 2.13. Muestran la forma de conectar resistencias de protección a los controles, de esta forma se pueden obtener ambos estados, alto y bajo. Debido a su condición de resistencias de protección no es relevante el valor que se utilice para ellas, solo se debe considerar un valor bajo para que no se consideren carga el resto del circuito. La misma consideración se debe tener para *Switchs*, selectores e incluso el *dip-switchs* como del que se trato en párrafos anteriores.

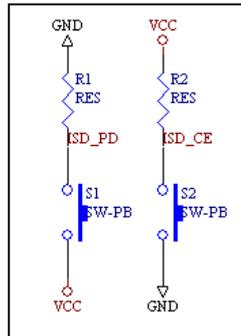


Figura 2.13. Controles de la tarjeta universal.

Fuente: Autores.

En la grafica 2.14. Se indica el circuito completo de la placa de pruebas para el ISD, que simplemente es la combinación de todos los interfaces antes mencionados. Todos los módulos están habilitados, su funcionalidad depende solamente de su configuración.

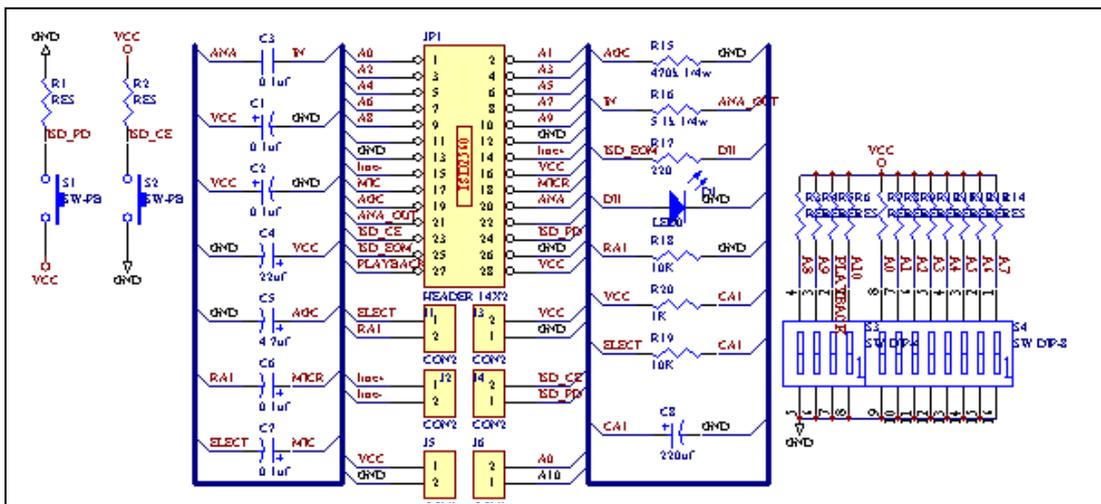


Figura 2.14. Circuito de la Tarjeta Universal ISD.

Fuente: Autores.

CAPITULO III DISEÑO Y MONTAJE DEL HARDWARE

3.1. Sensores.

Un sensor es un dispositivo que capta alguna magnitud o condición física (también llamada variable de instrumentación) y la transforma en una magnitud eléctrica, medible a través de un circuito electrónico, el cual debe ser capaz de tolerar cualquier condición de dicho sensor, esto se refiere a las condiciones extremas del componente. Normalmente un sensor no es capaz de indicar la medida por sí solo, es necesario el uso de algún interfaz para acondicionar la señal que proviene de él. También es típico usar una configuración que convierta la señal en una indicación numérica y esto puede basarse en el manejo de LEDs, *displays* de 7 segmentos o LCDs o incluso mostrar los datos por medio de un *software* en un computador.

La variación de la magnitud física causa la variación, en forma proporcional, por ejemplo, de resistencia, corriente, capacidad o voltaje. El circuito que recibe la señal del sensor debe ser sensible a este cambio para traducirlo en resultados visibles y coherentes para quien recibe los datos. A veces es necesario tomar la señal y linealizar el efecto del sensor o convertir la variación a rangos soportados por la configuración electrónica que se esté utilizando.

Los sensores se pueden clasificar según tres criterios:

Clasificación en activos o pasivos quiere decir que se diferencian dos tipos de dispositivos. Los sensores activos no requieren de una fuente de alimentación para derivar una señal en función de su medición, por el contrario uno pasivo no es capaz de medir si no está conectado a su tensión de funcionamiento.

También se pueden clasificar los sensores según la señal de salida de los mismos. Están los dispositivos que solo tienen dos estados posibles, alto o bajo, abierto o cerrado, etc. Las dos condiciones están separadas por un valor de umbral de la variable monitoreada. También existen componentes cuya salida está compuesta por varios canales digitales en arreglos de tipo binario, BCD, etc. Por último, los sensores que producen una señal analógica de voltaje o corriente cuyo valor es cambiante según especificaciones del sensor.

El último criterio que se utiliza para encasillar a los sensores es el tipo de magnitud física que pueden monitorizar. Según se indica en la tabla 3.1.a. y 3.1.b. En las que se ven algunos ejemplos de sensores según la característica física que manejan y el tipo de salida que producen.

Magnitud	Transductor	Característica
Posición lineal o angular	Potenciómetro	Analógica
	Encoder	Digital
Desplazamiento y deformación	Transformador diferencial	Analógica
	Galga extensiométrica	Analógica
Velocidad lineal y angular	Dinamo tacométrica	Analógica
	Encoder	Digital
	Detector inductivo	Digital
Aceleración	Acelerómetro	Analógico
Fuerza y par (deformación)	Galga extensiométrica	Analógico
Presión	Membranas	Analógica
	Piezoeléctricos	Analógica

Tabla 3.1.a. Tipos de sensores.

Fuente: wikipedia.com

Magnitud	Transductor	Característica
Caudal	Turbina	Analógica
	Magnético	Analógica
Temperatura	Termopar	Analógica
	RTD	Analógica
	Termistor NTC	Analógica
	Termistor PTC	Analógica
	Bimetal	I/O
Sensores de presencia	Inductivos	I/O
	Capacitivos	I/O
	Ópticos	I/O y Analógica
Sensores táctiles	Matriz de contactos	I/O
	Piel artificial	Analógica
Visión artificial	Cámaras de video	Procesamiento digital
	Cámaras CCD o CMOS	Procesamiento digital
Sensor de proximidad	Sensor final de carrera	
	Sensor capacitivo	
	Sensor inductivo	
	Sensor fotoeléctrico	
Sensor acústico (presión sonora)	micrófono	
Sensores de acidez	ISFET	
Sensor de luz	fotodiodo	
	Fotorresistencia	
	Fototransistor	
Sensores captura de movimiento	Sensores inerciales	

Tabla 3.1.b. Tipos de sensores.

Fuente: wikipedia.com

Para el presente proyecto son necesarios tres tipos de sensores. Para detectar una posible presencia dentro de las zonas de seguridad se deben utilizar sensores de movimiento. Para advertir de una puerta o ventana violentada se necesitan sensores magnéticos. La visión del equipo se efectúa mediante cámaras web que sustituyen a las típicas utilizadas para monitoreo.

3.1.1. Sensores de Movimiento.

Para realizar la tarea de encontrar cualquier cambio en el ambiente dentro del predio que se está protegiendo se ha optado por un detector de presencia óptico. Estos elementos se basan en la emisión y recepción de un haz de luz, por lo general infrarroja, y reaccionan cuando cualquier objeto interrumpe dicho haz. La luz utilizada puede estar dentro del espectro visible ó invisible. La señal puede tener modulación en el emisor y demodulación en el receptor, esto con el fin de evitar accionamientos intempestivos, que pueden ser causados por reflejos de la luz visible.

Según la alineación entre receptor y emisor se conocen dos tipos de sensores ópticos. Los sensores de óptica alineada necesariamente deben estar uno frente al otro como se indica en la figura 3.1. Para su implementación se debe tener en cuenta que sus lentes deben ser de muy buena calidad y se complementan con filtros para evitar accionamientos inesperados. Permiten cubrir grandes distancias, aproximadamente 100 metros, pero mientras mayor es la distancia será más la dificultad para alinearlos y mantenerlos así.

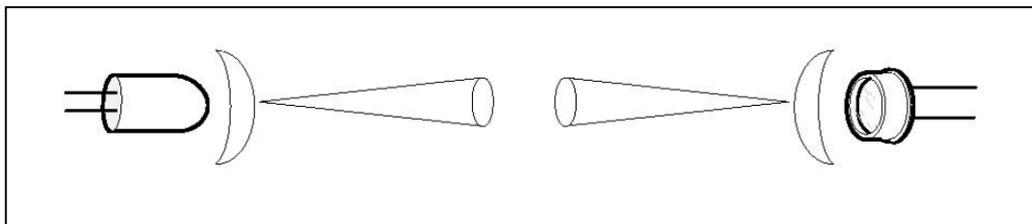


Figura 3.1. Esquema de un Sensor de Óptica Alineada.

Fuente: wikipedia.com

Los sensores de óptica reflexiva tienen el receptor y el emisor en el mismo dispositivo y buscan el reflejo de la luz sobre el objeto que se quiere detectar, el esquema de la figura 3.2. Muestra la forma en la que se debe realizar la reflexión de la luz. Estos sensores se deben calibrar en función de la distancia que deben cubrir y los objetos que deben detectar.

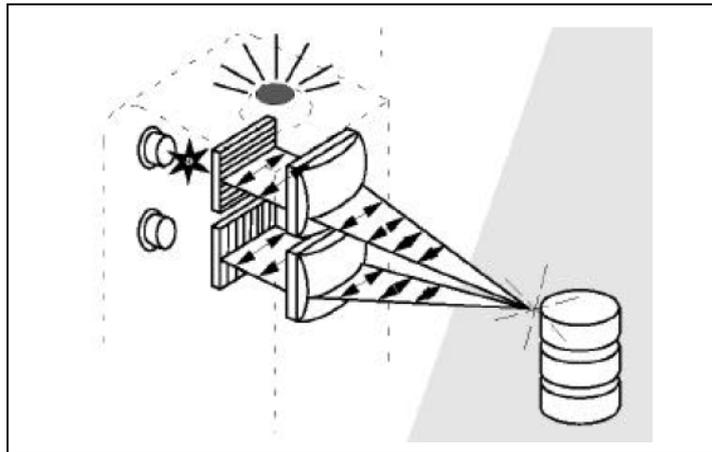


Figura 3.2. Esquema de un Sensor de Óptica Reflexiva.

Fuente: wikipedia.com

El sensor tiene dos terminales para alimentación (12V) y 4 entradas que dan acceso a dos contactos normalmente cerrados. Si una entrada del PIC no se conecta el micro interpreta el ingreso como un estado alto; el sensor se debe conectar de modo que el contacto cierre el canal entre tierra y la entrada al equipo. Al cambiar el estado para el que fue calibrado, el sensor abre los contactos normalmente cerrados, lo que produce un 1 virtual lo que indica al equipo que existe una presencia dentro de la zona de protección.



Figura 3.3. Sensor de Movimiento.

Fuente: Autores

3.1.2. Sensores Magnéticos.

Un sensor magnético es un dispositivo relativamente simple. Se puede explicar su funcionamiento haciendo una analogía con un interruptor común y corriente que al cambiar su estado interrumpe el paso de la tensión. El componente consta de dos partes metálicas revestidas de un material aislante (plástico), las cuales tienen potencial magnético contrario, por lo que se atraen entre sí. Cuando las dos partes están unidas el interruptor está cerrado, los dos terminales están eléctricamente unidos (conducen). Cuando las dos plaquetas magnéticas son separadas el interruptor se abre y la señal que atraviesa el componente es cortada.



Figura 3.4. Sensor Magnético.

Fuente: portaldelaindustria.com

En la imagen 3.4. Se muestra un sensor magnético. Típicamente las dos plaquetas se colocan una en la parte fija y otra en la móvil, ya sea de una puerta o de una ventana, al abrirse las dos partes del dispositivo se separan causando el cambio de estado de la señal inyectada al equipo por el sensor. Por las características del componente no es necesario adecuar la señal, el PIC tolera la señal directamente por uno de sus puertos de entrada salida.

3.2. Cámaras WEB.

Una *webcam* es, actualmente, un dispositivo altamente difundido por sus aplicaciones en videoconferencia a través de internet. Este componente es muy parecido a una cámara fotográfica digital, pero controlada desde un *software*, típicamente, a través del puerto USB. El software captura un fotograma o una trama desde la cámara cada cierto tiempo y la envía a otro punto para ser visualizada. Para construir una imagen de video sin saltos de imagen se debe alcanzar una tasa de transferencia entre 15 a 30 tramas por segundo.



Figura 3.5. Cámara Web.

Fuente: hardwarezone.com

Normalmente una cámara web está constituida por un lente, un sensor de imagen y la circuitería para controlar estos dos elementos. La parte electrónica es la encargada de leer la imagen del sensor y transmitirla a la computadora. Antes de poder utilizar la cámara web se debe instalar el *software* como si se fuera a utilizar con el programa del

fabricante, esto a fin de que se instalen los controladores del dispositivo que son necesarios sea cual sea la aplicación en la que se vaya a usar.

Para que la cámara web sea compatible y controlable desde LABVIEW, que es el programa de computadora que se utilizará, hace falta que la cámara web tenga ciertos controladores, como *DIRECT SHOW* que, por desgracia, no se indica entre las propiedades de las cámaras aunque las tengan. Para saber si una cámara es compatible hace falta instalar el software propio del componente y elaborar, sobre LABVIEW, un instrumento virtual que detecte todos los componentes de video conectados al equipo. Si el instrumento reconoce al dispositivo lo etiqueta. Este nombre es necesario para, posteriormente utilizar la cámara en cualquier otro instrumento virtual.

3.3. Diseño del PCB.

Un circuito impreso o PCB (por *Printed Circuit Board*) es un medio físico para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de hilos o pistas de material conductor, típicamente cobre, grabados a partir de una lámina del conductor adherida a un sustrato no conductor. Aunque existen tecnologías para elaborar tarjetas de múltiples capas, lo más usado localmente es tecnología doble cara.

Para el desarrollo del diseño de la tarjeta de circuito impreso se necesita un programa de Diseño Asistido por Computador (CAD), especializado en este tipo de tareas. Protel99 brinda un amplio rango de herramientas de desarrollo de circuitos electrónicos en su fase esquemática y de circuito impreso, el software también permite la programación y simulación. Es recomendable, de ser necesario simular el circuito, hacerlo de forma modular, cada parte funcional por separado.

Este programa cuenta con varios tipos de bases de datos que asocian componentes con varias características, por ello se pueden generar archivos de diferente tipo. A partir de un archivo de tipo esquemático (.sch), en el que especifican el formato físico asignado a cada componente (huella), se puede generar la lista de materiales y dar las

especificaciones que concluyen con la generación del diseño del circuito impreso (archivo .pcb).

Se pueden crear componentes y sus huellas en caso de no existir tales dentro de alguna de las bibliotecas incluidas en este software. Un buen método para crear una huella es partir de un bosquejo dibujado sobre papel cuadrículado, esto a fin de tener precisión en las dimensiones y por su puesto la forma del componente. En la figura 3.6. Se muestra un ejemplo de un elemento y junto a él su huella, necesaria antes de crear el PCB.

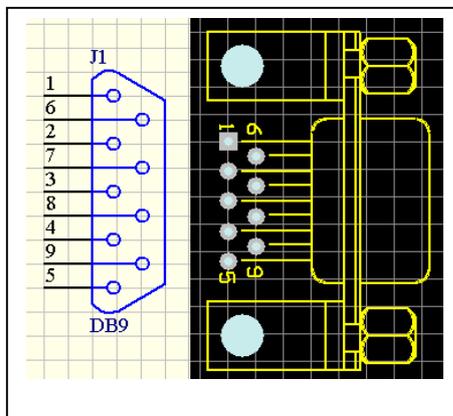


Figura 3.6. Representación Esquemática y Huella de un DB9 (Puerto Serial).

Fuente: Autores.

En capítulos anteriores se han descrito los circuitos de control e interfaces necesarios para el correcto funcionamiento de cada parte del proyecto. Debido a que se han realizado las pruebas correspondientes, sobre un tablero, y se comprobó la funcionalidad de todos los circuitos, no es preciso que se simule ninguno de ellos. Luego de armar el circuito completo se debe especificar la huella que da la forma y las dimensiones de cada componente. El siguiente paso es generar una lista de nodos en la cual se detallan los puntos de conexión entre los diferentes componentes.

Una vez concluida la etapa de esquema se crea una tarjeta virtual para el PCB con dimensiones tentativas. En este paso se deben elegir varias características que pueden variar según la aplicación como, por ejemplo la dimensión máxima y mínima de cada hilo conductor, separación entre ellos, tecnología *through hole* o superficial,

número de capas, etc. Con el nuevo archivo PCB en blanco se carga la lista de nodos con lo aparecen todos los componentes del circuito. De no haber errores, se inicia el posicionado de todos los elementos.

La primera consideración de posicionado se relaciona con todos las borneras o conectores de entrada y salida, mismos que deben ubicarse en los bordes de la tarjeta. También se debe tener en cuenta separar lo más posible los componentes que manejan potencia de los que sirven de comunicación y que por tanto pueden verse afectados, por interferencias. Siguiendo con las comunicaciones, es recomendable fijar las líneas de transmisión y recepción lo mas paralelas posible para el reducir el ruido inducido por otros componentes.

Al terminar el posicionado de los materiales comienza el ruteo. En la figura 3.7. Se muestra el diseño terminado de la cara superior del PCB, tanto del equipo de seguridad como de la tarjeta universal de pruebas del ISD. Al ruteo se le adiciona un *Poligon Plane*, que es una herramienta mediante la cual se coloca una capa de conductor conectada a todos los puntos unidos a algún nodo y se usa para hacer más robusta una parte de la placa, por ejemplo, en esta capa se puede ver un *Poligon Plane* con respecto a los puntos en contacto con VCC, también llamado plano de alimentación. Es posible combinar la superficie de la placa con varios planos referidos a diferentes conexiones, pero no es necesario cuando las pistas son del grosor suficiente, por ello solo se ven este tipo de planos en referencia a la alimentación del equipo.

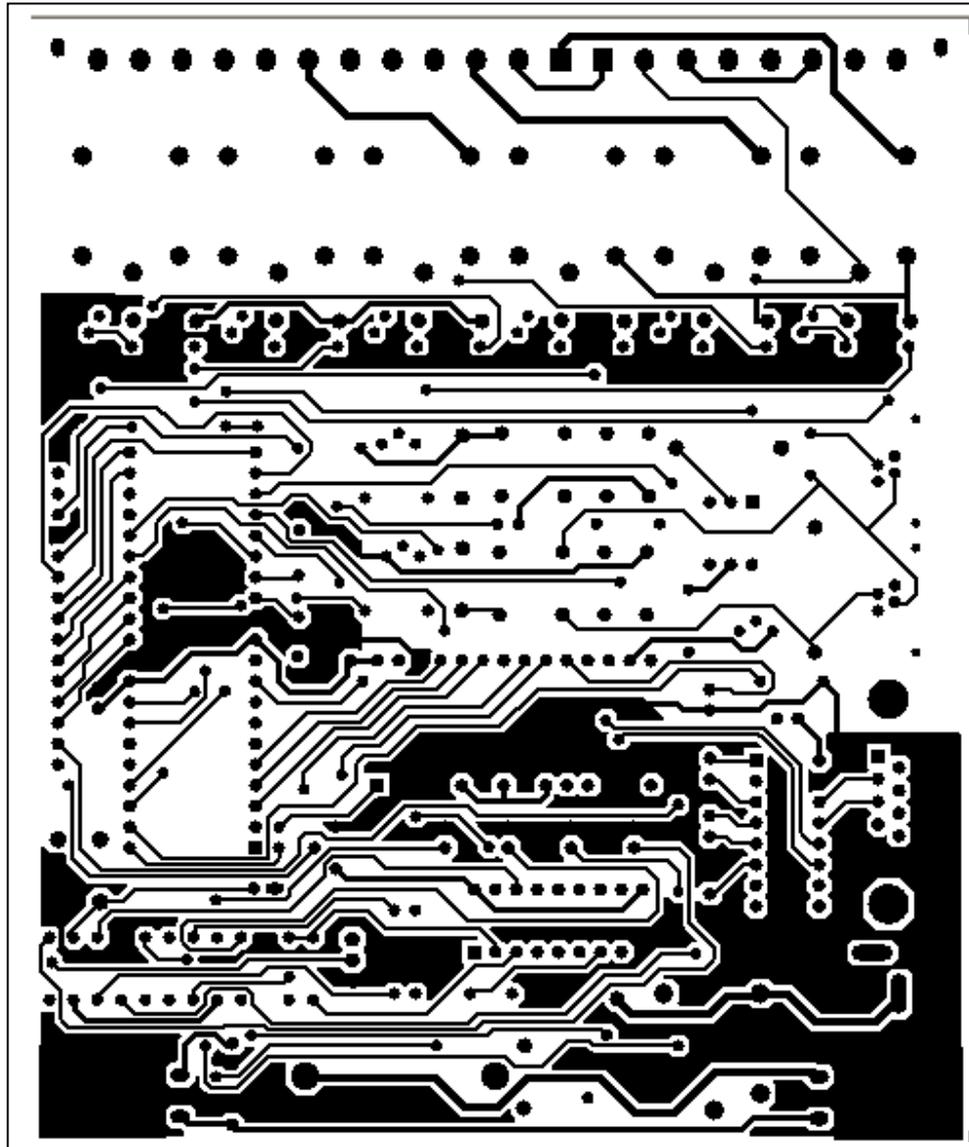


Figura 3.7a. Cara superior del PCB del equipo.

Fuente: Autores.

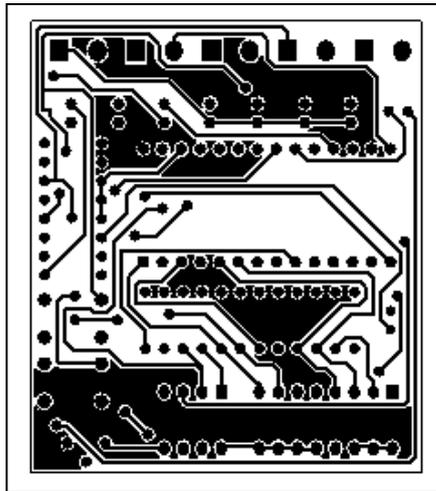


Figura 3.7b. Cara superior del PCB de la Tarjeta Universal del ISD.

Fuente: Autores.

También en la cara posterior se ha colocado un *Poligon Plane* referido a tierra, mismo que se puede ver en la figura 3.8. Para separar los elementos que posiblemente manejen corriente alterna del resto de los componentes, sensibles a estas señales, es imperativo no ubicar planos que sean comunes para ambos o simplemente no colocarlos. La creación de los planos es el detalle constructivo final por lo cual el diseño de PCB queda terminado y listo para su elaboración.

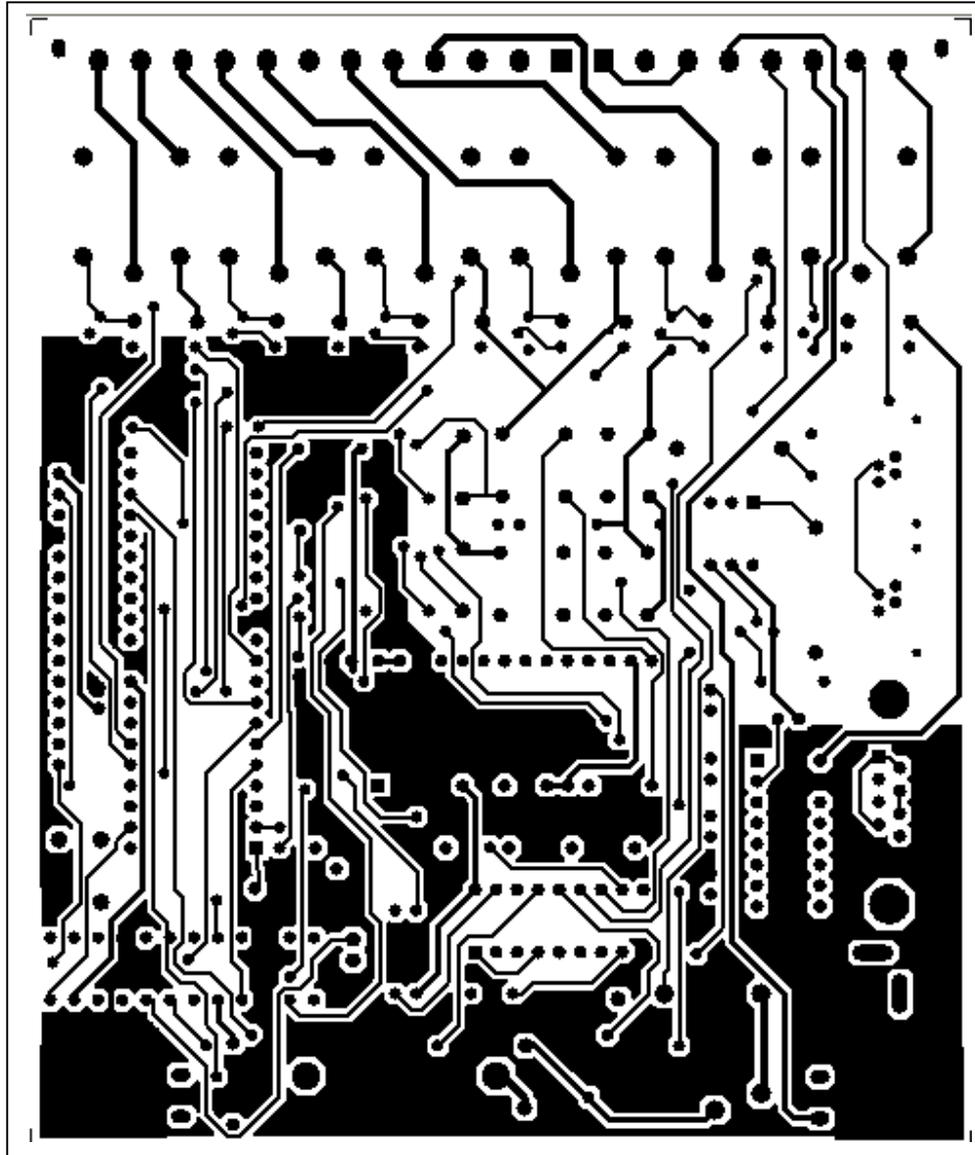


Figura 3.8a. Cara posterior del PCB del Equipo.

Fuente: Autores.

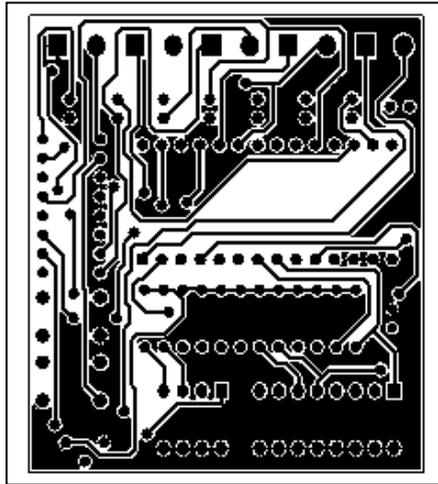


Figura 3.8.b. Cara Posterior del PCB de la Tarjeta Universal del ISD

Fuente: Autores.

3.4. Montaje del PCB.

Protel99 también genera la guía para el montaje del PCB, a través de la vista de impresión (figuras 3.9a. y 3.9b.), que muestra todos los componentes y sus etiquetas en su posición final. Esta vista junto con la lista de materiales (anexo 3) es esencial para el montaje de los componentes de forma correcta, esto se refiere a orientación, polaridad, valor y hasta tamaño de todos los elementos.

Es muy recomendable aprovechar la ventaja que tienen algunos componentes de conectarse a la tarjeta mediante adaptadores o *sockets*, tal es el caso de varios de los integrados que se incluyen en el circuito, así como el LCD que se conecta mediante un conector tipo peineta macho. Se pueden colocar conectores peineta hembra para el cristal y para algunos elementos, esto da la posibilidad de ser reemplazados. Para el PIC existe un socket especial tipo "ZIP" que se ajusta mediante una palanca para colocar y retirar el *chip* sin forzar ni dañar sus pines. Fuera de estas consideraciones todos los componentes se sueldan directamente.

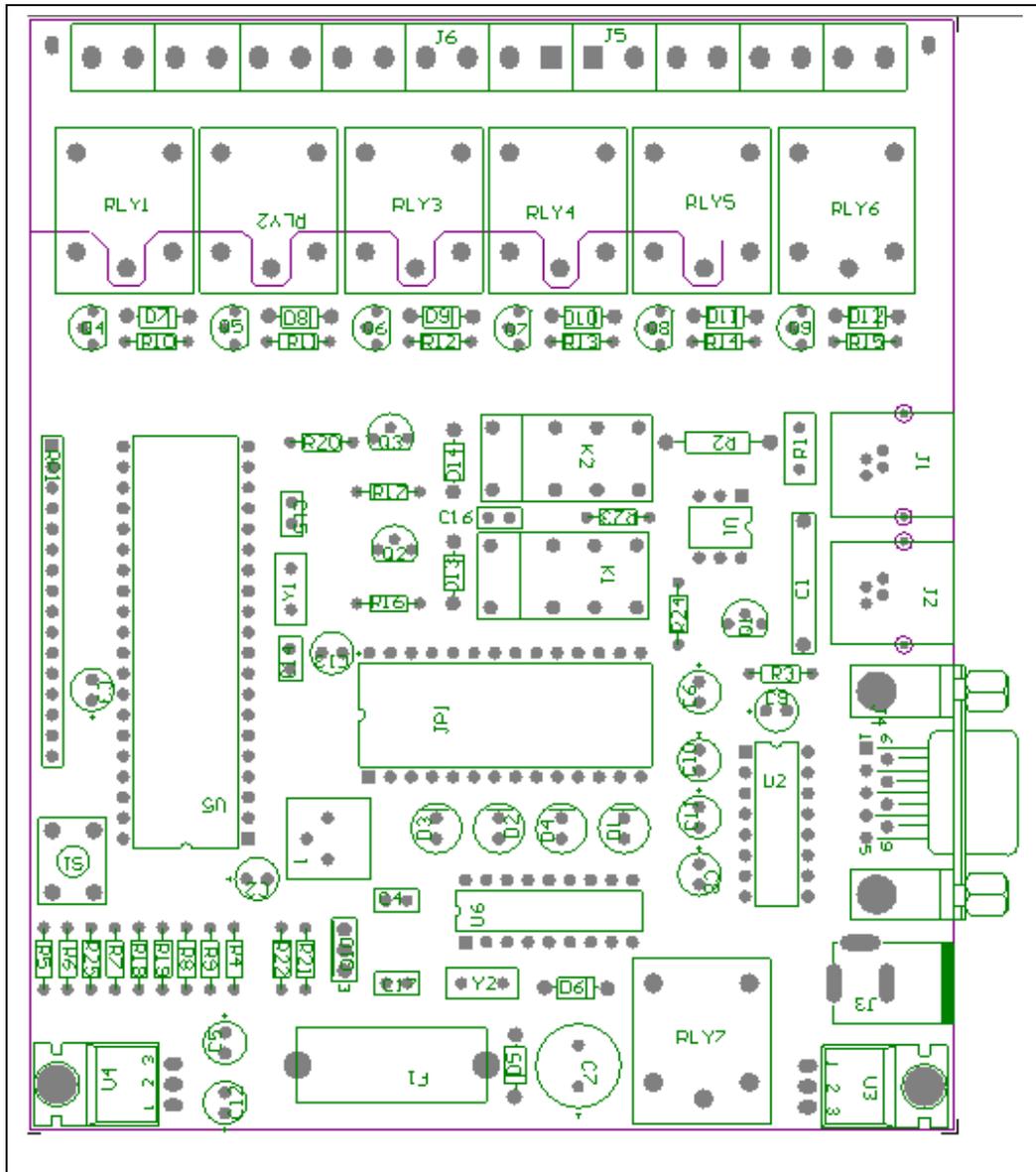


Figura 3.9a. Esquema para montaje de Componentes del Equipo.

Fuente: Autores.

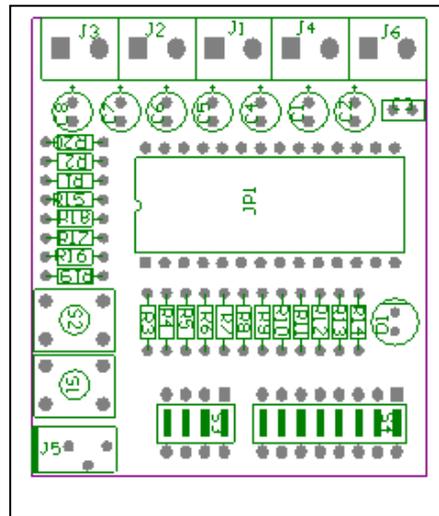


Figura 3.9b. Esquema para el montaje de la Tarjeta Universal del ISD.

Fuente: Autores.

Para evitar usar cajetines telefónicos dobles el dispositivo está equipado con dos terminales tipo RJ11 en paralelo, esto da la posibilidad de conectar tanto la línea telefónica como el terminal telefónico sin interrumpir la conexión entre ellos, tal como un modem de computador.

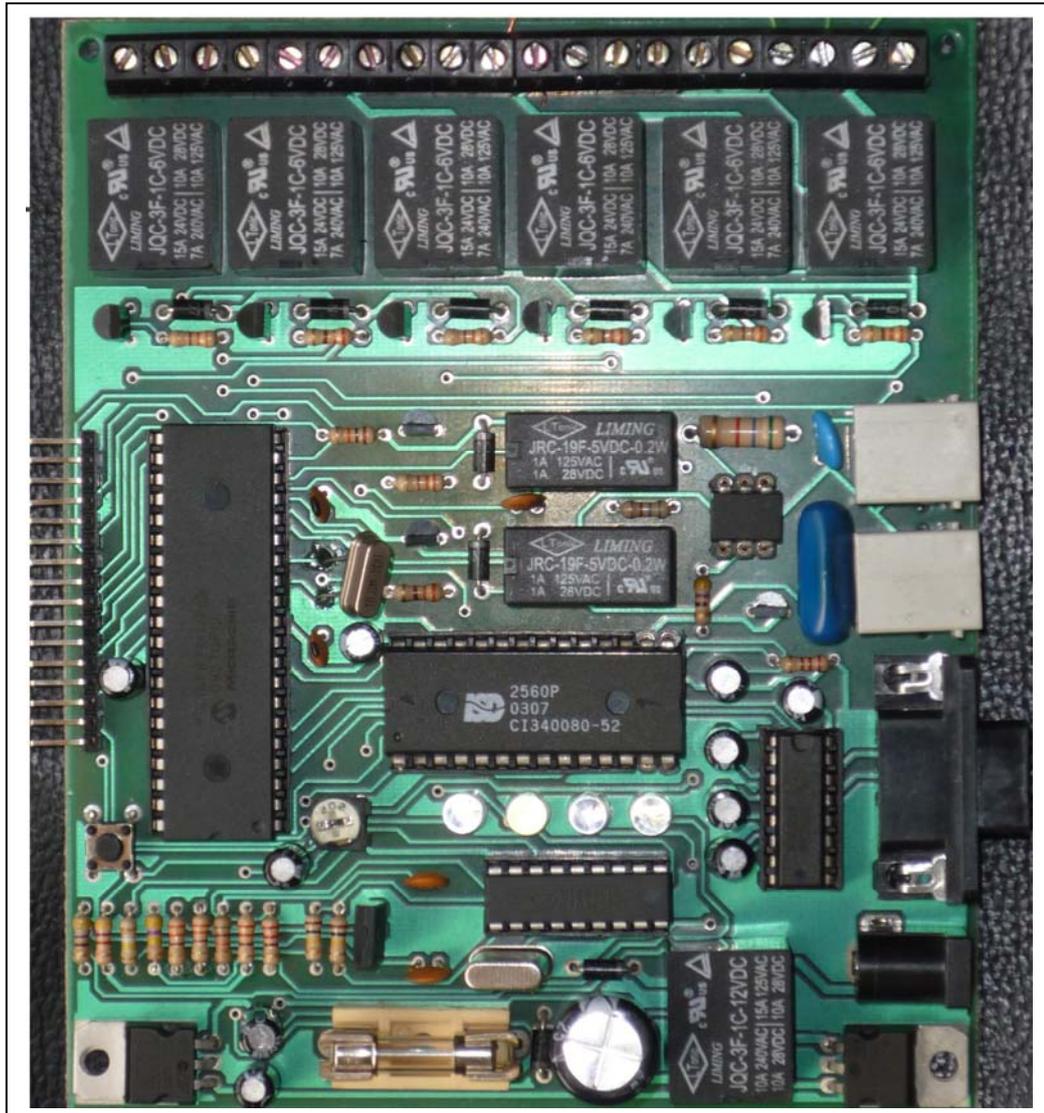


Figura 3.10. Montaje terminado del PCB.

Fuente: Autores.

La figura 3.10. Muestra la imagen real del PCB terminado.

3.5. Pruebas de Funcionamiento.

Al terminar el montaje de todos los componentes sobre la tarjeta se procede a realizar las pruebas que corroboren su correcto funcionamiento. Para evaluar la funcionalidad

del circuito se procede de la misma forma que en las pruebas realizadas sobre el tablero (*protoboard*), esto es pruebas individuales o modulares para cada parte del circuito. Para lo cual se generan programas de PIC independientes. Cada programa puede ser luego reutilizado en forma de subrutinas que formaran parte del programa que administra todo el equipo.

La primera prueba, como es lógico, debe ser de la fuente de tensión. Inicialmente al conectar el *plug* de alimentación se debe encender un LED como indicación de que la polaridad de la tensión es correcta, luego se debe medir la salida del estabilizador de tensión LM7805, pues el PIC trabaja con 5V de alimentación. Al desconectar la alimentación el relé de ingreso debe conmutar la fuente, quedando energizado el equipo desde la batería. La transición entre las dos posibles fuentes no debe causar el apagado del dispositivo. Después de comprobar la fuente se puede colocar el PIC y empezar a evaluar el equipo.

Para poner a prueba el funcionamiento del LCD se debe tener en cuenta que a mas de elaborar las subrutinas de visualización de carteles se debe también controlar el puerto de salida activo en 1, que controla el *backlight* y comprobar el correcto funcionamiento del potenciómetro que controla el contraste del indicador.

Para facilitar las pruebas del puerto serial se puede elaborar un conector tipo *Loopback*, mediante el cual se recibe el mismo dato que se envía, esto a fin de evitar el tener que lidiar con algún otro dispositivo que tenga puerto serial. Este elemento es simplemente un conector DB9 que tenga los pines de transmisión y recepción puenteados tal como se muestra en la figura 3.11.

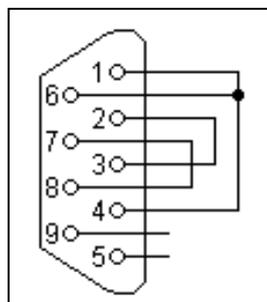


Figura 3.11. Conector DB9 en configuración *Loopback*

Fuente: Autores.

Para las pruebas sobre el control de los relés se puede conectar cualquier carga en las borneras correspondientes a cada uno de ellos, incluso sin carga alguna es fácil de percatarse cuando se activan por el ruido que provocan al cambiar de estado. Aunque es preferible evaluar su desempeño controlando corriente alterna en combinación con el uso de la transmisión serial, de esta forma se sabe si hay inducciones de ruido considerables.

CAPITULO IV SOFTWARE SOBRE LABVIEW

4.1. Introducción al Ambiente LABVIEW.

Para controlar los sistemas de adquisición de datos, casi siempre, se requiere un software de instrumentación, flexible para la edición y de fácil manejo. El programa debe, además de administrar los recursos de la computadora, presentar los datos y realizar el análisis. LABVIEW es un software especializado y desarrollado para la creación de instrumentación virtual, y por ello cuenta con numerosas herramientas para adquisición de datos, análisis y generación de señales, creación de botones, indicadores y controles, etc.

Es posible crear interfaces de usuario mediante consolas interactivas basadas en *software*. Para programar en LABVIEW se puede usar un lenguaje de programación compatible como lenguaje c o *basic*, pero también se puede simplificar la programación utilizando lenguaje G. Cuando se programa sobre LABVIEW se trabaja bajo "Instrumentos Virtuales", abreviando, VIs por su siglas en ingles, se pueden crear VIs a partir de especificaciones funcionales. Un VI se puede utilizar dentro de otra aplicación, como una sub función dentro de un programa general.

Su característica de programación gráfica hace de LABVIEW una herramienta mucho más fácil de utilizar que los lenguajes de programación tradicionales. Obviamente, es más claro y fácil de entender un grupo de iconos y símbolos que líneas de código, aunque se trate de un lenguaje sencillo.

El software diferencia dos partes del programa el diagrama de bloques, que es donde se desarrolla el programa con las especificaciones y todas las funciones, y el panel frontal que contiene los indicadores, controles y todas las respuestas o salidas del sistema. Una vez creado el programa y probado su funcionamiento, es posible eliminar

el diagrama de bloques y ejecutar la aplicación solamente con el panel frontal, pero se debe asegurar su total funcionalidad, ya que sin el diagrama de bloques lo único que es sensible a cambios es la apariencia de los controles.

Cada herramienta en el lenguaje G es un icono que contiene un símbolo que caracteriza su función, una interfaz con el usuario, un color que identifica el tipo de dato que maneja y una o varias salidas. En la figura 4.1. Se muestran algunos ejemplos de funciones del diagrama de bloques que forman un instrumento virtual. El ejemplo realiza un proceso cuyos elementos son: un *while loop*, una demora de 10ms (milisegundos), un comparador con una constante, y un indicador (*power*), además de un control general (*stop*).

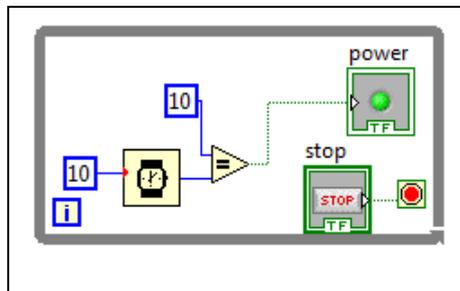


Figura 4.1. Ejemplo de un instrumento Virtual en su etapa de Bloques.

Fuente: Autores.

Al crear un Indicador o un control en el diagrama de bloques se crea también su contraparte en el Panel de Frontal, pero con la diferencia de que en el panel se puede editar la apariencia, en cuanto a forma, color, orientación, etiquetas, estilo del indicador y en algunos casos la forma en la que funcionan los controles. También es posible crear un archivo ejecutable mediante el cual se puede realizar la aplicación sin abrirlo desde LABVIEW por lo que no es necesario tenerlo instalado, si se usa las herramientas de creación de ejecutables.

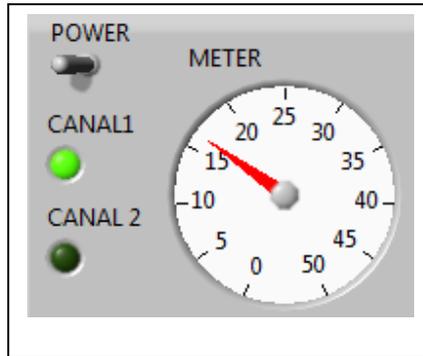


Figura 4.2. Ejemplo de Panel Frontal.

Fuente: Autores.

Depende de la aplicación para la cual se esté programando, se necesitan varios componentes, por ejemplo, para la comunicación serial se debe instalar el componente “VISA”, para manipular imágenes y visión artificial se necesita de “IMAQ VISION” y *Vision Assistant*, etc. Existen complementos para aplicaciones que van desde lo didáctico hasta lo industrial y, por supuesto, la domótica.

Como se mencionó en capítulos anteriores, para adquirir imágenes y video desde una cámara web y manipular la información desde LABVIEW, es necesario que exista cierta compatibilidad entre el dispositivo y el software, la arquitectura requerida se ilustra en la figura 4.3. En la que se muestra el componente de filtrado *Direct Show* como parte de los controladores de la *webcam* y tres componentes de visión artificial *National Instruments* que deben instalarse previamente.

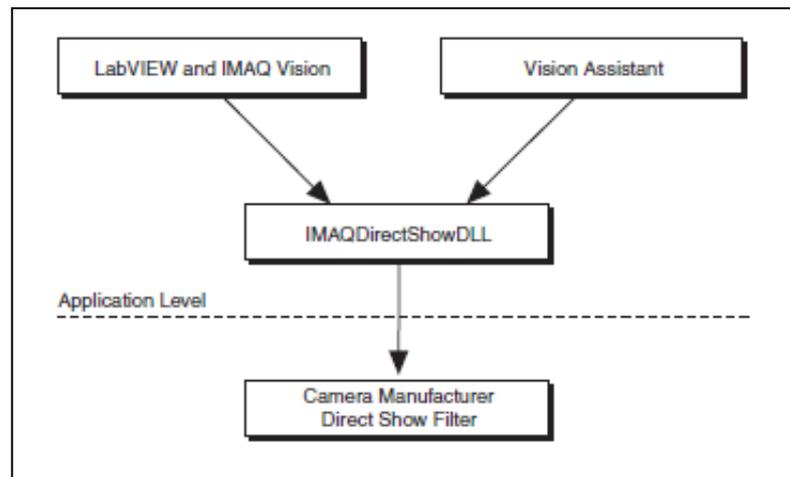


Figura 4.3. Arquitectura de Adquisición de Video.

Fuente: USER GUIDE NI-IMAQ for USB Cameras.

4.1.1. Requerimientos para la Instalación.

Para esta plataforma el fabricante especifica los siguientes requerimientos para que la computadora ejecute correctamente el software: por lo menos 256 MB de memoria RAM, pero se recomienda 1 GB o más. Se requiere una resolución de pantalla de 1024 x 768 pixeles como mínimo. Para plataforma Windows, ya sea 2000, XP o Vista se necesita: Pentium III o Celeron de 866 MHZ o un procesador equivalente, pero se recomienda Pentium 4/M o mayor. Se necesita un mínimo de 1.2 GB de espacio libre en disco para la instalación completa.

Los requerimientos de pantalla se deben a que LABVIEW maneja gráficos con colores de 16 bits con una paleta de 256 colores. También se aconseja tener *Adobe Reader* 5.0.5. ó mayor para acceder a los manuales. El *software* no es soportado por los *Windows* NT/Me/98/95 y tampoco las versiones XPx64 ni Vistax64, la versión de *Windows* 2000 que lo tolera es el *Service Pack* 3.

Es necesario aclarar que, como se pretende manipular y transmitir video en tiempo real, mientras mayor sea la memoria RAM y la velocidad del Procesador el instrumento virtual trabajara de manera más eficiente; de ser necesario se debe bajar la resolución

de la cámara a fin de evitar cortes o saltos en la adquisición de video. También se depende de la capacidad de bajada-subida de la conexión de internet para la correcta ejecución remota del VI.

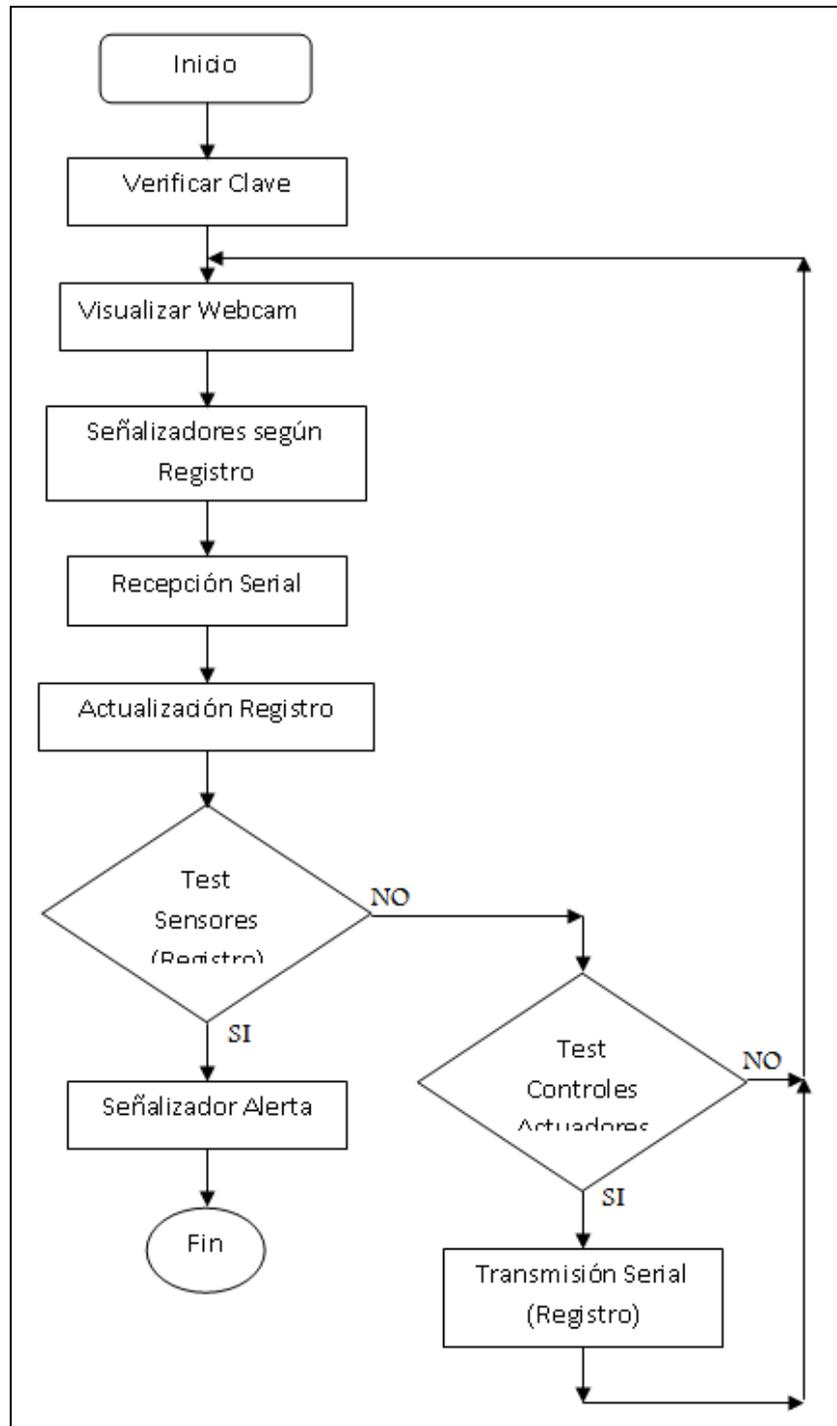
4.2. Modulo de Adquisición.

Típicamente en adquisiciones de datos usando LABVIEW se usan dispositivos (DAQs) de marca *National Instruments*, de transmisión vía puerto USB en los cuales es necesario establecer el modo de trabajo, es decir, entradas analógicas o digitales, salidas analógicas o digitales, frecuencia de muestreo, modos de referencia (modo diferencial o modo referencial a tierra), control en líneas o en puertos, etc. De la misma forma en el *hardware* que reemplaza al DAQ se deben setear algunos parámetros necesarios para el correcto funcionamiento entre la tarjeta y el ordenador.

La comunicación que se establece entre la tarjeta de adquisición y la computadora se realiza por completo a través del puerto serial, por ello lo que se debe especificar, durante la puesta a punto del sistema, es el número de puerto, la velocidad de transmisión, el número de bits que se transmiten en cada trama, la paridad, todo esto de acuerdo e idéntico a la configuración del puerto serial del dispositivo destino, en este caso el PIC. De tratarse de una computadora de escritorio, la cual cuenta con un puerto serial, el número de puerto es el COM1, pero de ser una laptop y si se está usando un conversor serial-USB suele usarse el COM3. Como se trató en capítulos anteriores la comunicación serial se realiza a 9600 baudios con 8 bits de datos un bit de stop y sin paridad tal y como se estableció para el micro controlador.

4.3. Software sobre LABVIEW.

4.3.1. Diagrama de Flujo.



Fuente: Autores.

4.3.1.1. Funcionamiento.

Al Inicio el instrumento presenta el panel frontal con todos los actuadores apagados y sin adquisición de video, esto hasta que sea ingresado el nombre de usuario y clave correctos, al verificarse las dos claves conocidas únicamente por el administrador del sistema entra en funcionamiento el panel frontal. En cada ciclo se ejecuta la adquisición y reproducción de video desde las cámaras web, así como también la recepción del puerto serial que modifica el estado de los indicadores del panel frontal, finalmente se envía el dato del nuevo estado a través del puerto serial.

En caso de recibir un cambio de estado en los bits correspondientes a los sensores se produce una alerta, traducida en el encendido del indicador del panel frontal. Del mismo modo se pueden encender o apagar los actuadores a través de los controles creados para ello, de ser el caso después de cualquier cambio de estado en cuanto a actuadores se realiza una transmisión serial que modifica el registro en la memoria de programa del PIC, con el fin de que exista concordancia entre lo que indica el LCD y el panel frontal.

De no haber cambio alguno ni en sensores, actuadores ni alertas la transmisión y recepción se realiza de forma periódica, esto garantiza que si se produjeran errores de comunicación el error se corrige en la actualización, en el siguiente ciclo. Es necesario aclarar que desde el instrumento virtual no se puede controlar las funciones de envío y recepción telefónica puesto que lo contrario implicaría la dependencia entre las dos formas de control e indicación.

4.3.2. Diagrama de Bloques.

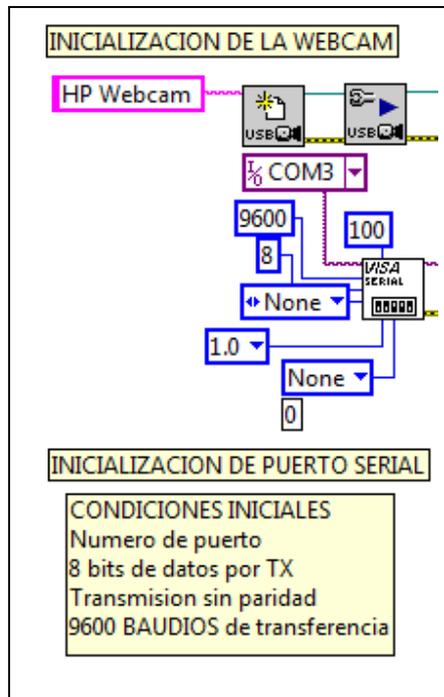


Figura 4.5. Inicialización del puerto serial.

Fuente: Autores.

En primera instancia se inicializa el puerto serial a través de VISA, visa es un estándar de comunicación que se puede utilizar para comunicación GPIB, serial, USB, Ethernet, PXI o VXI. Solo se necesita especificar acerca del protocolo que se desee utilizar. Tal y como se muestra en la figura 4.5. Se utiliza el icono de VISA serial, que se encuentra en la paleta de telecomunicaciones y dentro de protocolos, para seleccionar los valores con los que va a operar el puerto serial. Para definir los parámetros se cablean constantes numéricas hacia las entradas del icono. Cabe aclarar que el puerto se inicializa una sola vez, por lo se debe utilizar fuera de cualquier lazo de repetición presente en el programa. De la misma forma se debe encender la cámara web mediante el nombre por el cual el ordenador identifica al dispositivo, este nombre se puede ver en el recuadro de color rosa "HP Webcam" (según la cámara o cámaras conectadas).

A continuación viene el cuerpo del programa, el cual se compone de dos funciones principales las mismas que deben repetirse constantemente por lo que se ubican dentro de un *while loop*. Pero antes de realizarse cualquier adquisición o visualización se debe validar un nombre de usuario y una clave, que activa el caso verdadero de un *case* tal como se muestra en la figura 4.6. Caso contrario el programa deriva sus acciones hacia el caso falso que está vacío. La combinación de usuario y clave se establecen como constantes que se comparan con entradas de teclado y solamente las dos coincidencias dan acceso a la funcionalidad del programa.

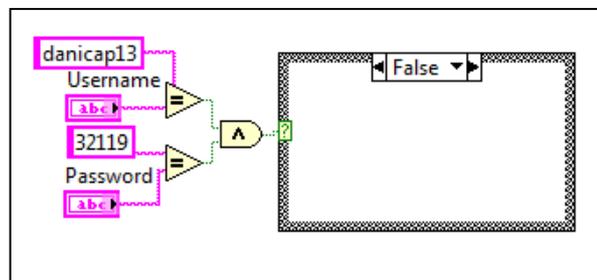


Figura 4.6. Validación de *Username* y *password*.

Fuente: Autores.

La primera función del lazo debe adquirir video de la webcam y visualizarse dentro de un indicador de imagen. En la figura 4.7 se muestra al forma de la adquisición continua de imagen de dos cámaras web. El icono etiquetado *IMAQ* es necesario debido a que mediante este se le asigna, temporalmente, un espacio de memoria al video que está siendo adquirido. El icono diferenciado USB captura la imagen de la webcam y la manda hacia el indicador. De idéntica forma se realiza en los dos casos de una estructura tipo *case* seleccionable a través de un control. Este proceso se está repitiendo constantemente, pero para optimizarlo se debe añadir una demora entre final y principio de la siguiente iteración, la cual se debe calibrar según la frecuencia de adquisición de la cámara que se esté utilizando. Cuando se finaliza el programa se debe cerrar el puerto y liberarse el espacio de memoria que se había reservado para el video, esto se logra cableando cada señal de video hacia un *close* que termina la adquisición y apaga las cámaras.

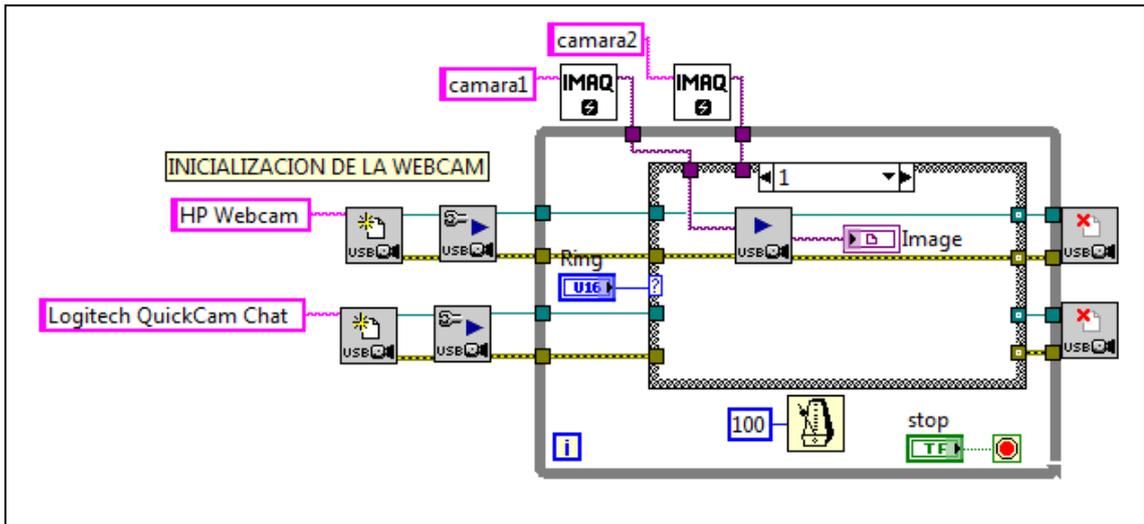


Figura 4.7. Diagrama de bloques de la adquisición, continua, de dos webcams.
Fuente: Autores.

Para lograr una mayor eficiencia del instrumento virtual se utiliza la técnica conocida como Máquina de Estados, esto es dividir dos o más procesos, y realizarlos de manera mutuamente excluyente, esto es posible solamente cuando los efectos de los procesos no se requieren de manera simultánea. La máquina de estados funcionara en tres estados posibles. En el primer caso se encuentra la adquisición de video de la cámara número uno, ubicada en la zona principal del domicilio. En el segundo caso se detiene el funcionamiento de la cámara uno y se adquiere de la cámara dos, que se supone en una área secundaria. En el tercer caso se detienen las dos cámaras. De esta forma se puede optimizar el uso del procesador y de la memoria RAM del computador, pero para mejorar aun mas las condiciones, se puede adquirir imagen de forma monocromática, es decir, en escala de grises.

La segunda función principal del programa es la comunicación vía puerto serial, esta tarea se divide en dos partes.

La transmisión serial o escritura se realiza si y solo si hay un cambio de estado en los controles que corresponden a los actuadores. Para simplificar el diagrama, por razones de espacio, el efecto de los 6 actuadores se ubica dentro de un subVI llamado "OR6" por tratarse de la suma (compuerta OR) de 6 entradas tal como se ilustra en la

figura 4.8. La salida de este sub instrumento, que solo puede ser verdadero o falso, se cablea al selector de un case que de ser falso no realiza ninguna transmisión.

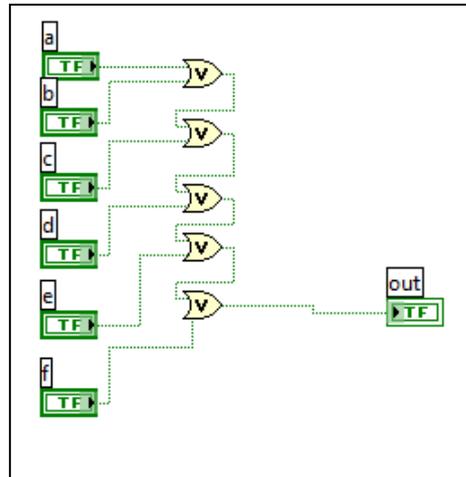


Figura 4.8. SubVI OR6.

Fuente: Autores.

De ser verdadero, es decir cuando hay un cambio de estado en cualquiera de los actuadores, se realiza la transmisión de un carácter según sea el actuador que se desea afectar. Los controles están cableados hacia las entradas de "OR6", pero también cada botón se conecta a la entrada de un selector que de ser falso deja pasar un dato de tipo *string* (cadena de caracteres), un espacio en blanco, pero de ser verdadero deja pasar el carácter correspondiente al canal (a, b, c, d, e, f). Las salidas de los 6 selectores están conectadas a un concatenador, que une los datos en un arreglo de caracteres, pero al afectarse solo un actuador a la vez el arreglo será de un carácter junto con 5 espacios en blanco. El arreglo de caracteres se cablea a la entrada del icono que escribe en el puerto serial tal como se muestra en la figura 4.9.

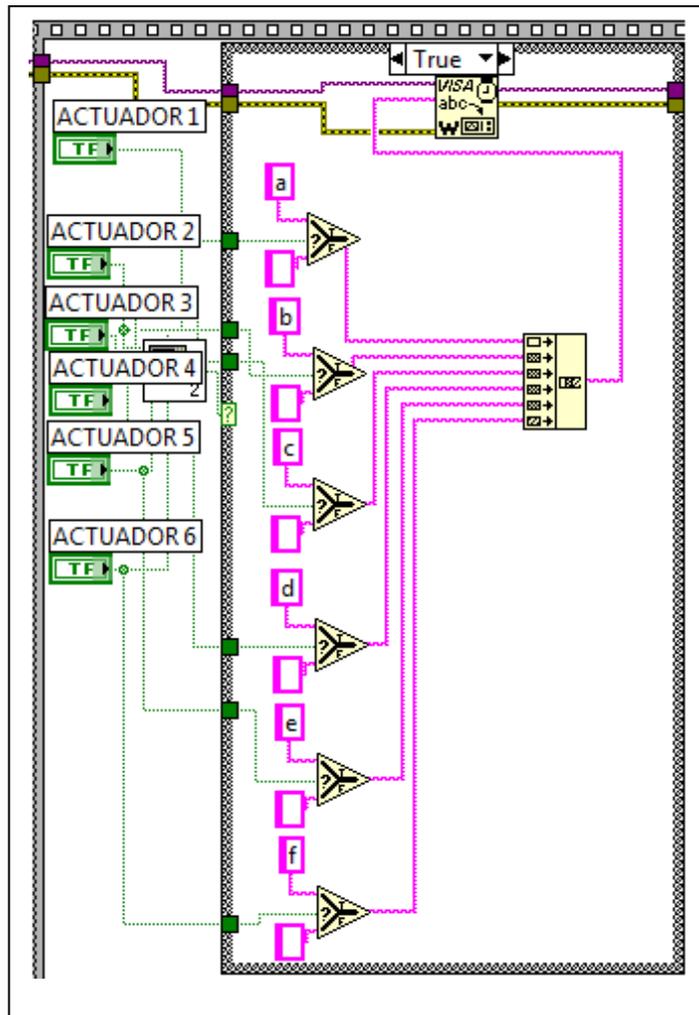


Figura 4.9. Transmisión Serial.

Fuente: Autores.

En la figura 4.10 se ilustra la etapa final del programa, la recepción serial, la cual se logra con el icono de lectura del puerto, pero con una característica especial, el número de caracteres que se reciben a la vez no puede ser dinámico debe ser estático y necesariamente 1. El dato recibido se compara con varias posibilidades desde la a hasta la p. El PIC envía un dato según el estado del actuador, sensor o alarma, por ejemplo, si se ha producido una alarma el computador recibe una letra “a” si el dispositivo no está en estado de alarma se recibe una “b”, si cualquiera de los sensores se activa se recibe una “c” y no están activados una “d”, si el actuador número 1 está encendido se recibe una “e” y si está apagado una letra “f”, así sucesivamente hasta llegar al sexto actuador. El carácter recibido se identifica con los comparadores conectados a la salida del lector del puerto.

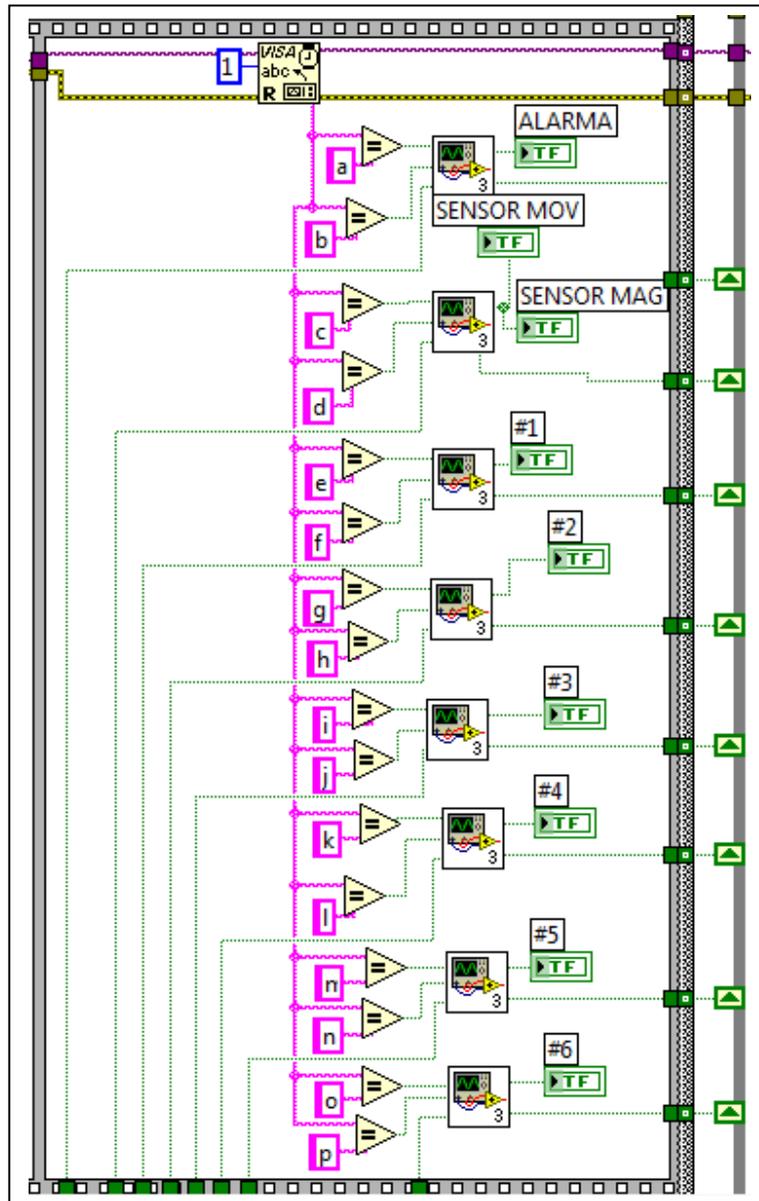


Figura 4.10. Recepción Serial.

Fuente: Autores.

Sobre la estructura *while* se pueden adicionar *shift register* que almacenan algún resultado para reutilizarlo en la siguiente iteración. En este caso son de color verde por almacenar datos de tipo Booleano (verdadero o falso) y almacenan el estado de las alertas, sensores y también de los actuadores.

Según sea el carácter recibido se activa una de las entradas del subVI “mantiene_constante”. el cual simula el funcionamiento de un *Flip-flop* tipo D, el cual al recibir un estado alto en su entrada “on” enciende la salida correspondiente, si el estado alto se encuentra en la entrada “off” entonces apaga la salida y en la entrada “shift” se cablea la realimentación. El estado de la salida “out” cambia solo si el estado anterior y el actual son diferentes.

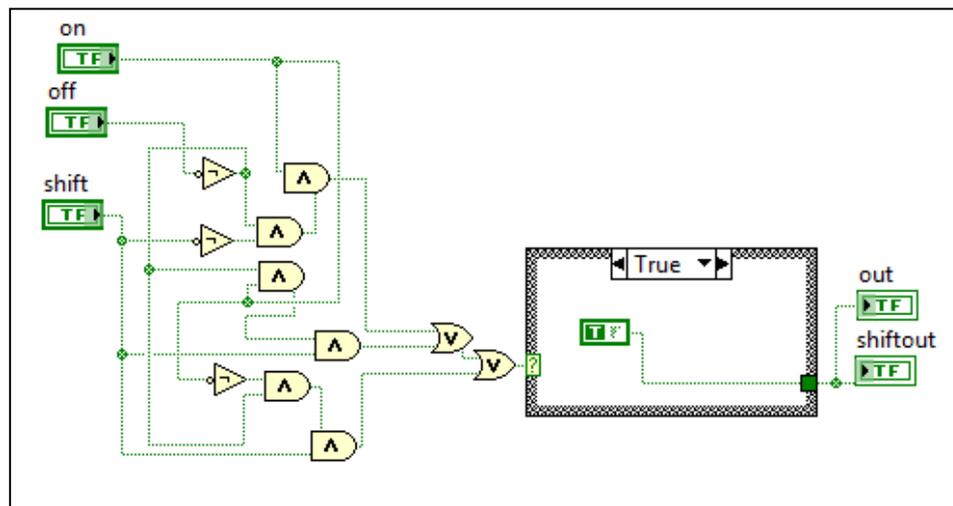
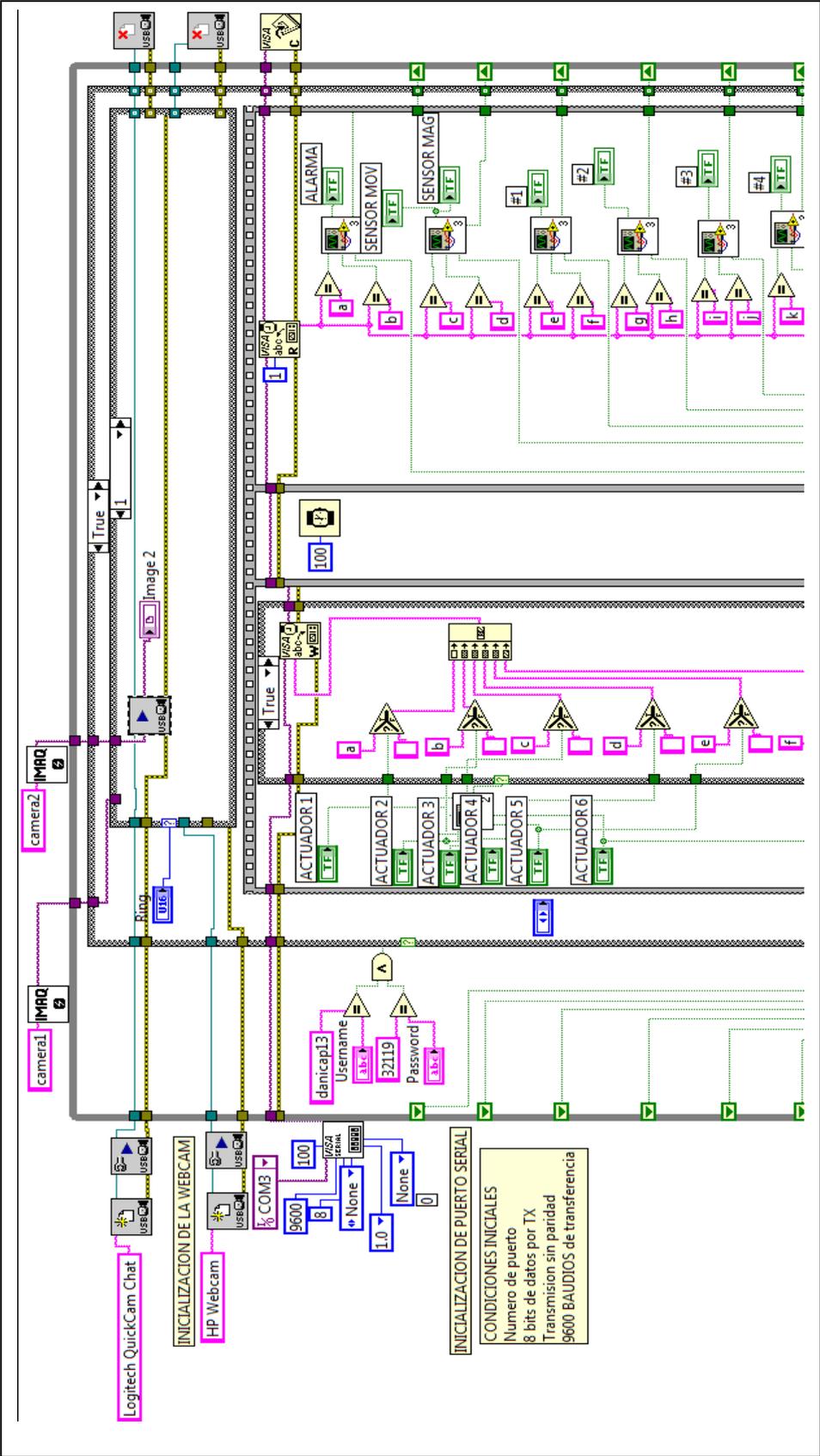


Figura 4.11. SubVI “mantiene_constante”.

Fuente: Autores.

En la figura 4.12 se ilustra la imagen reducida del diagrama de bloques funcional. Las funciones de la comunicación serial están dentro de una estructura tipo *flat* que sirve únicamente para establecer el orden en el que se ejecutara cada componente. Entre los bloques de transmisión y recepción se encuentra una demora que sirve para no sobre ocupar el procesador de la maquina.



4.3.3. Panel Frontal.

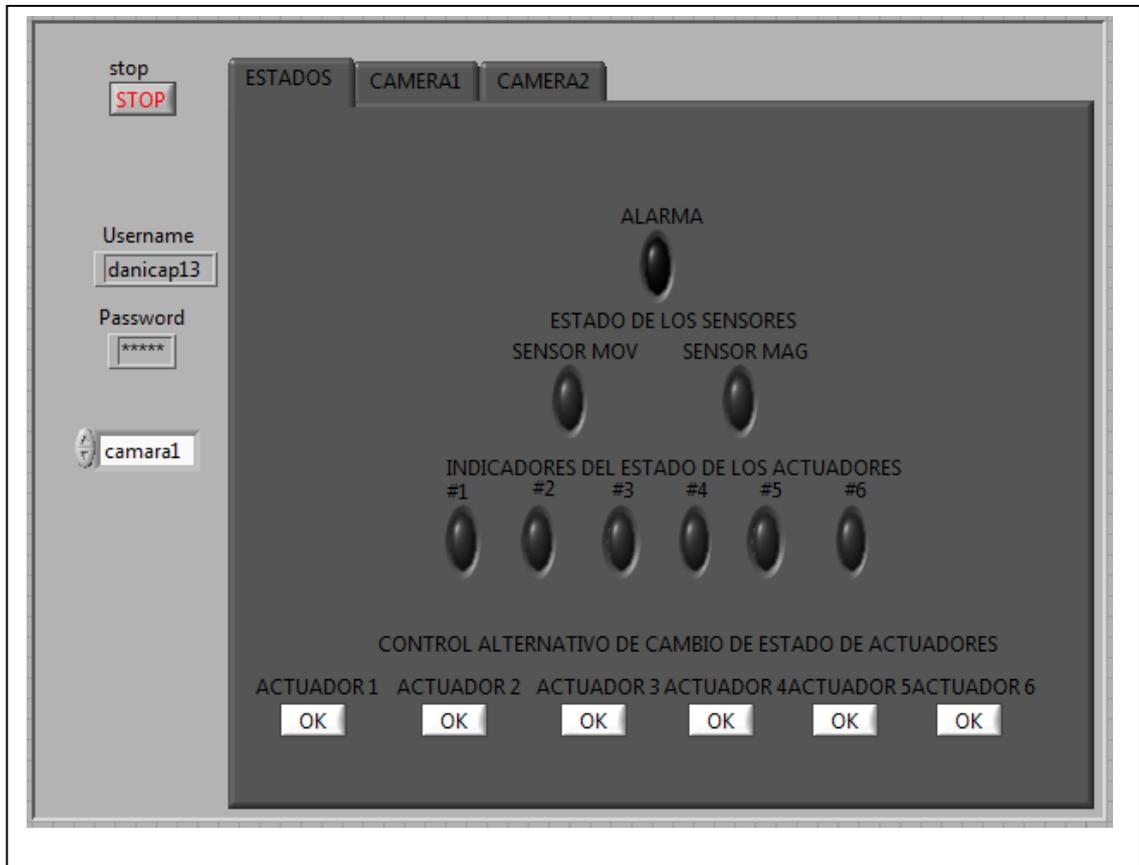


Figura 4.13. Panel Frontal del instrumento.

Fuente: Autores.

En la figura 4.13. Se muestra el panel frontal del instrumento virtual, en la parte izquierda están dos casillas destinadas al ingreso de un nombre de usuario y una clave que valida el funcionamiento de todo el sistema, debajo de los casilleros se ubica un control tipo anillo que sirve para seleccionar la fuente de video que se desea adquirir. Para los indicadores se usa un contenedor tipo “tab” que consta de diferentes frentes. En la caja etiquetada “ESTADOS” se encuentran los indicadores de la alarma, estado de los sensores y también el de los 6 indicadores, hacia abajo al final se encuentra los controles de los actuadores, plenamente identificados. En las cejas siguientes se encuentran las imágenes correspondientes a las cámaras 1 y 2.

Como seguridad se coloca un stop general en la parte superior izquierda del panel frontal, que detiene todos los procesos.

4.4. Comunicación con el PIC.

Como se indicó en el primer capítulo, el modulo de adquisición no adquiere señales analógicas por lo que no es necesario poner demasiado cuidado en parámetros como frecuencia de muestreo, amplificación instrumental, linealización o rechazo al modo común, todos estos procesos se realizan típicamente en cualquier dispositivo de instrumentación que trabaje con señales con rangos muy sensibles al ruido.

La facilidad de trabajar con señales que ya están en formato digital agiliza mucho el proceso, por ello el instrumento virtual se limita a editar e interpretar los datos que le llegan a través del puerto serial e indicar los resultados con el panel frontal del VI. El dato recibido se identifica y el sistema actualiza los indicadores según sea el cambio de estado de las entradas y salidas del dispositivo.

Dentro del PIC existe un registro que guarda el estatus del sistema, es decir, todos los estados individuales de las entradas y salidas importantes. Antes de que el dispositivo realice la transmisión serial el registro de comunicación se actualiza con los valores de sensores, de actuadores y de alertas. Por su puesto los únicos valores sensibles a cambios desde el instrumento virtual son los estados de los actuadores, mismos que se indican con actualizaciones periódicas.

Si se necesitara controlar un mayor número de actuadores se tiene que cambiar la forma de transmisión, poniendo el estado de los actuadores en un solo registro de 8 bits y transmitirlo después de enviar el que contiene los sensores y alertas. Si son dos datos diferentes los que se deben comunicar se deben anteponer de marcas para su identificación, tal como preceder el dato de los sensores por un dato en ASCII y el dato de los actuadores por uno diferente. Típicamente se utiliza como marcas de datos los caracteres ASCII como "*" o "#", fácilmente identificables tanto por un micro controlador como por la PC.

Es necesario aclarar que la comunicación no se limita si la computadora en la que se va a implementar el sistema no tiene puerto serial, como sería el caso de una computadora portátil, debido a que actualmente los modelos comerciales de laptops no cuentan con terminales de este tipo. Existe un conversor que sirve para hacer compatible la transmisión serial con un puerto de tipo USB, el protocolo de comunicación sigue siendo RS-232, pero se accede a la computadora a través de un terminal USB.

El conversor se encarga de igualar las tensiones de operación de la comunicación serial a los valores que tolera el protocolo de transmisión USB, obviamente la velocidad de la comunicación está limitada a los estándares Seriales (la menor entre ellas).

4.5. Comunicación con la WEB.

Un aspecto muy importante de LABVIEW es que usa las tecnologías de red enfocadas a sus sistemas de medición. El software provee una plataforma para diseñar un sistema de prueba que toma ventaja de las tecnologías de redes más recientes mientras el ambiente sigue dirigido a desarrollar sus aplicaciones rápidamente. Cualquier aplicación en LABVIEW puede convertirse en una aplicación remota accesada vía un Web Browser. Se puede habilitar la capacidad de compartir datos en vivo para objetos en su interfaz de usuario.

Existen muchas tecnologías disponibles que sirven para compartir información a través de la web y son utilizadas para hacerlo de diferentes formas. La publicación de datos genera un reporte de web estático de los resultados de las pruebas, que luego se puede compartir con otros usuarios; se puede ingresar al reporte desde cualquier Web Browser estándar. El compartir datos expande las funciones de la publicación de datos para lograr la transferencia entre computadoras, esto en algunos tipos de procesos en los que se pueden cambiar los parámetros de medición mientras la aplicación se está ejecutando. El control remoto expande el concepto de solo compartir datos habilitando a otra computadora para conectarse a la prueba y controlarla desde la computadora remota. La ejecución distribuida tolera que varios usuarios accedan a la información desde diferentes computadoras.

La figura 4.14. Ilustra un instrumento virtual que adquiere una señal de ruido y la filtra, este ejemplo sirve para demostrar el proceso que se sigue para usar un panel remoto de una aplicación cualquiera. Como se menciona anteriormente es necesario el panel frontal ya terminado y probado antes de comenzar, para la comunicación vía web no es preciso trabajar con el diagrama de bloques.

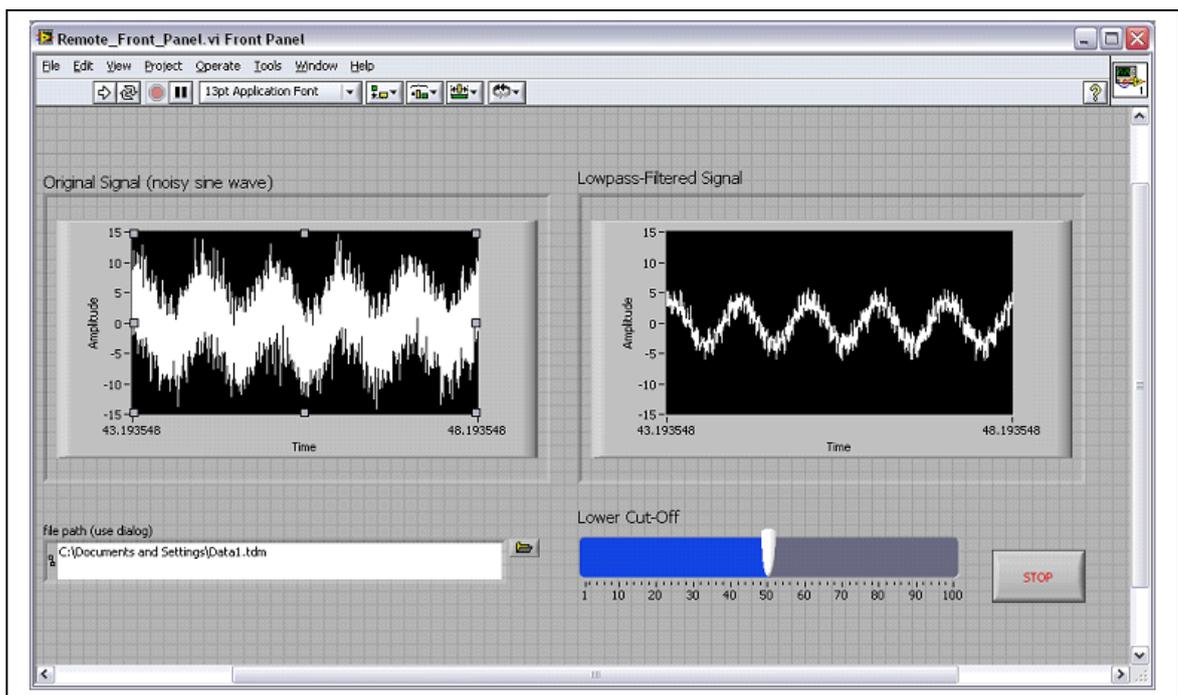


Figura 4.14. Ejemplo de Instrumento Virtual.

Fuente: ni.com

Para publicar información en el internet es necesario que se registre dicha información en una página web y para esto se necesita de un servidor Web, LABVIEW cuenta con la posibilidad de hacer las veces de servidor. Para configurar el servidor se ingresa en el menú *Tools > Options*, en la parte izquierda de la ventana que aparece se elige *Web Server: Configuration* y se habilitan las opciones que se muestran en la figura 4.15. Por defecto se encuentra designado el puerto HTTP 80. También se puede definir el tiempo que el instrumento se tarda en actualizar la información en el navegador desde la aplicación Web.

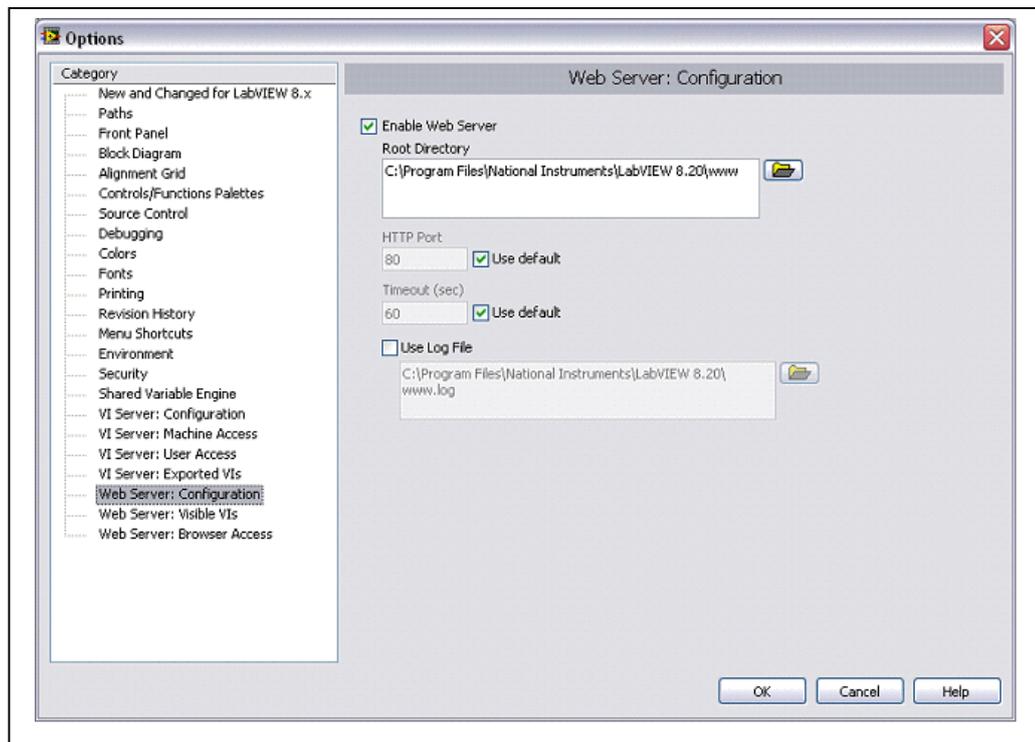


Figura 4.15. Habilitando Web Server.

Fuente: ni.com

Una vez que se ha habilitado el Servidor Web se debe establecer las condiciones iniciales con las que comenzará a trabajar el instrumento cada vez que sea abierto. Se puede hacer que la aplicación esté disponible para solo un usuario a través del internet. Ejecución reentrante significa que no hay límite para las máquinas que pueden ingresar a la página Web simultáneamente; esto se logra con solo habilitar la opción ingresando al menú *File > VI Properties*, se despliega el menú de propiedades y se elige *Execution*. Las otras opciones se setean como se muestra en la Figura 4.16.

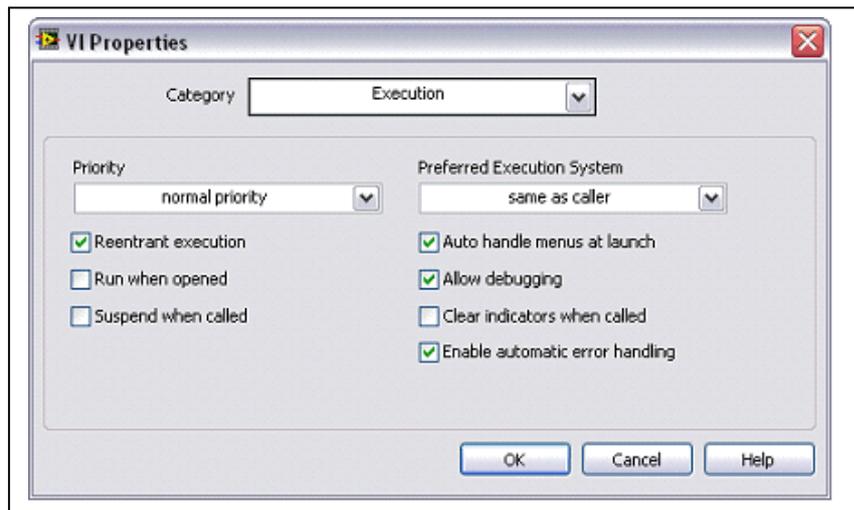


Figura 4.16. Propiedades de la Página Web.

Fuente: ni.com

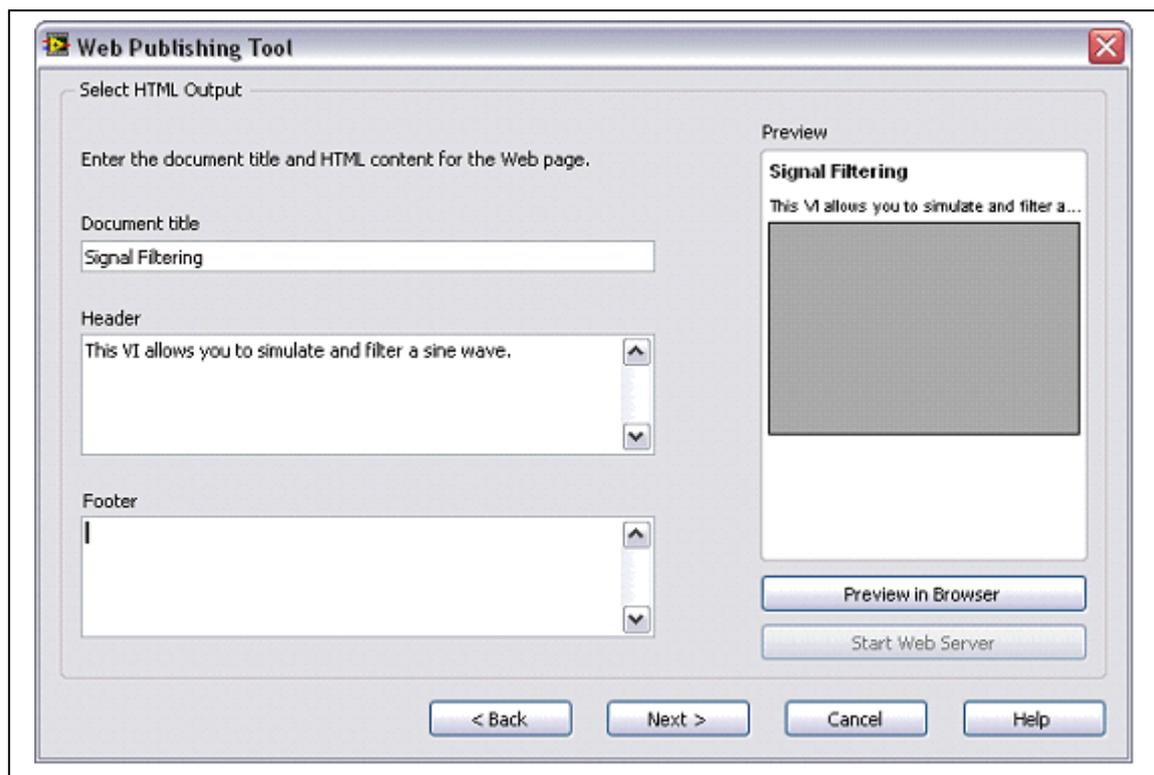


Figura 4.17. Título, Encabezado y pie del Documento Web.

Fuente: ni.com

Una vez establecidas las propiedades del VI se debe Publicar el mismo, para ello se debe ir al menú *Tools > Web Publishing Tool*. Como se ilustra en la figura 4.17. Dentro de este menú se tiene la posibilidad de editar el título que llevara la página web, de la misma manera se puede escribir el encabezado y el pie que acompañan al VI cuando se visualiza a través del navegador. Al terminar la edición del texto que se publicara en la web, para seguir con la configuración, se hace clic en el botón *next*.

La siguiente ventana sirve para definir la forma en la que se publicara la aplicación en la web (figura 4.18.), para este caso puntual, se debe seleccionar de la lista desplegable que aparece bajo el Título *VI name* el instrumento Virtual que se desea publicar. Se debe seleccionar la opción *Embedded* dentro de los modos en los que se puede visualizar el instrumento dentro de la página web. También se debe habilitar la búsqueda de control cuando la conexión sea establecida. Para continuar la configuración se hace clic en *next*.

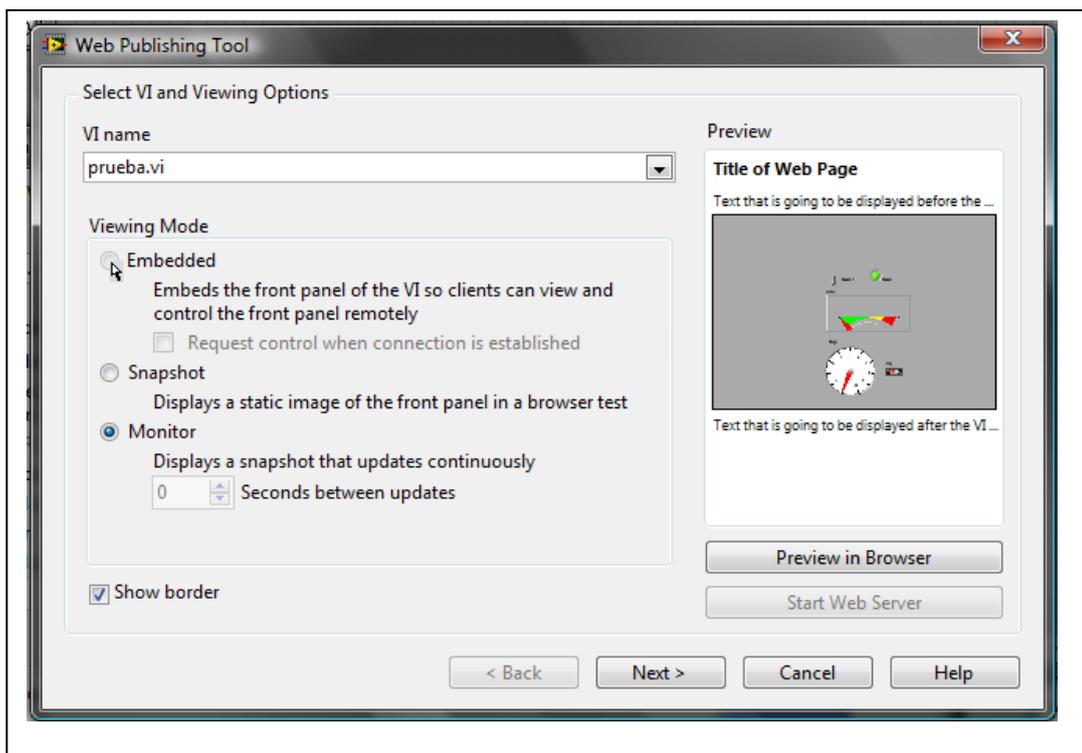


Figura 4.18. Establecimiento del modo de Visualización.

Fuente: ni.com

En la figura 4.19. Se muestra la ventana subsiguiente, en ella se muestran, en la parte izquierda, el directorio en donde se creara la página web, debajo aparece un espacio en que se debe escribir el nombre del archivo .HTML que contiene la información y por ultimo aparece la dirección URL que sirve para conectarse con el servidor WEB y acceder a la aplicación, ya sea de forma *Local Host* o externamente. Hay que resaltar que la dirección web se crea por defecto con el ID de la maquina que se está usando como servidor.

En la parte derecha de la ventana como en otras anteriores aparece una vista previa del instrumento virtual junto con su titulo, encabezado y pie, tal como se podrá visualizar mediante un navegador cualquiera. Haciendo clic en el botón *Save to Disk* se crea el archivo que contiene el VI dentro de la carpeta web a continuación aparece un cuadro de dialogo que da la posibilidad de conectar de manera *Local Host* con la aplicación a través del navegador de internet.

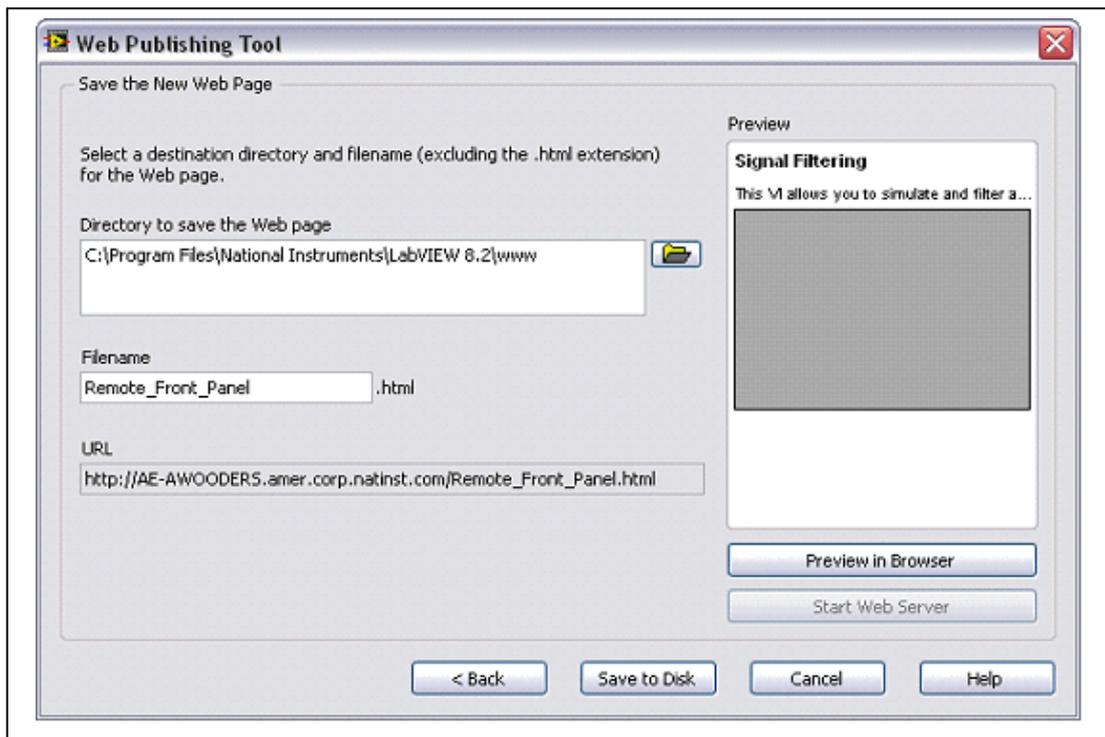


Figura 4.19. Creación de la Pagina HTML para el Panel Remoto.

Fuente: ni.com

En la figura 4.19. Se puede ver la ruta de acceso a la carpeta web en la que se graban todos los instrumentos disponibles a través de la web, así como el nombre del archivo .html (formato soportado por el navegador web).

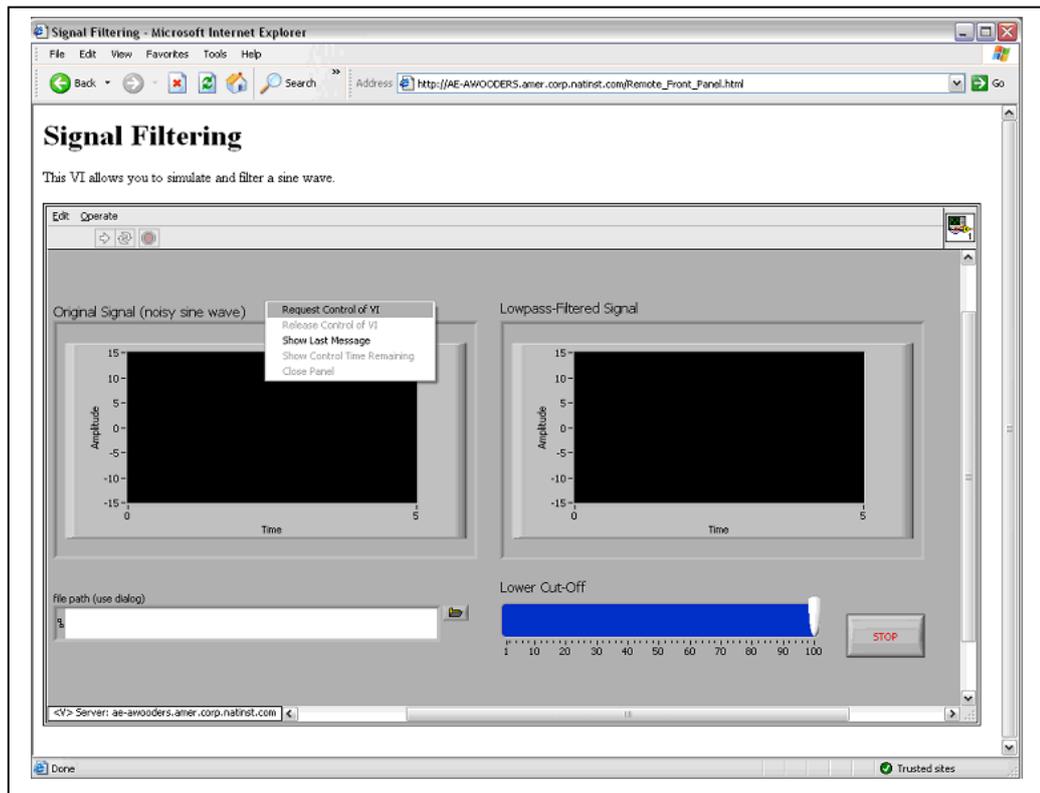


Figura 4.20. Panel Remoto abierto desde el Navegador de Internet.

Fuente: ni.com

En la figura 4.20. Se muestra el instrumento virtual abierto desde el *Internet Explorer*, para ejercer control sobre él se debe hacer clic derecho sobre VI y luego elegir la opción *Request Control of VI* a continuación aparecen tres botones en el borde superior del VI como se muestra en la figura 4.21. El primero es para la ejecución del programa, el segundo realiza una ejecución continua y el último la detiene.



Figura 4.21. Botones del Panel Remoto.

Fuente: Autores.

Finalmente en la figura 4.22. Se muestra el instrumento virtual ejecutándose de forma remota desde el *Internet Explorer* y con la capacidad de controlar las variables dinámicas del mismo. Desde la ubicación remota se puede reclamar el control del instrumento virtual, pero el servidor puede recuperar el control en cualquier momento. Obviamente esto solo se puede hacer si se conocen tanto el nombre de usuario como la clave que valida el funcionamiento del programa. Cuando se está ejecutando el VI desde el servidor es conveniente bloquear el acceso de cualquier maquina remota esta precaución brinda seguridad adicional ante cualquier intento de acceso no permitido.

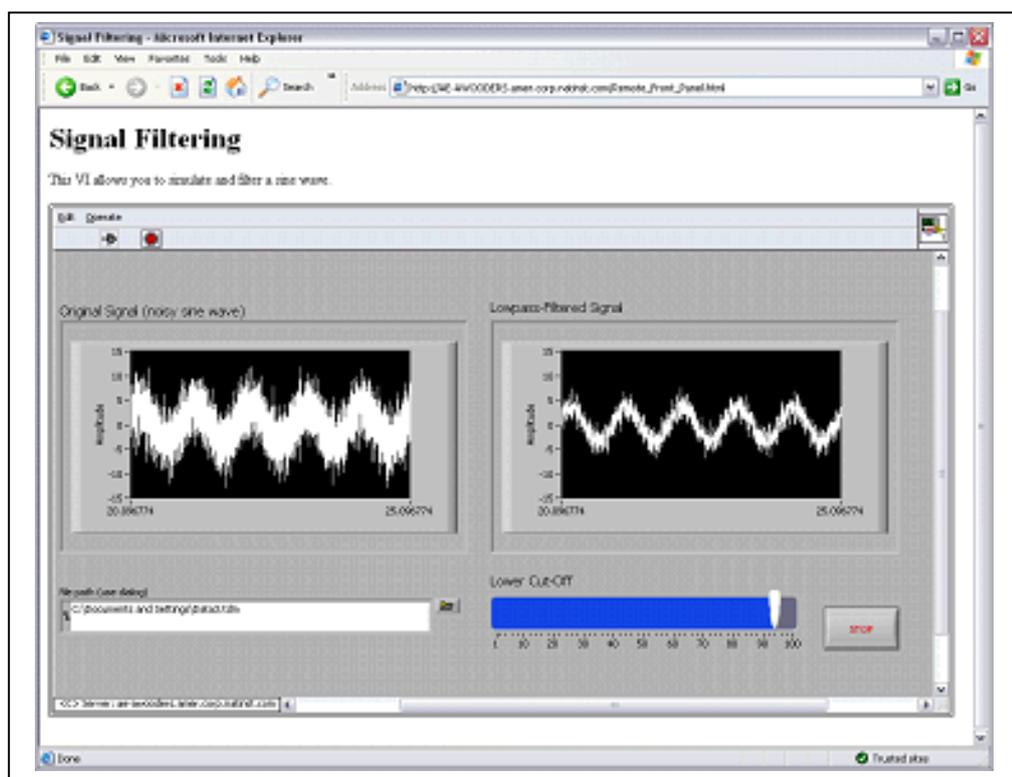


Figura 4.22. Instrumento Virtual Controlado desde el Navegador de Internet.

Fuente: ni.com

Evidentemente para editar los colores o cualquier detalle en la apariencia del panel se debe trabajar desde el archivo .vi y no desde el .html, debido a que los únicos cambios posibles desde este último son el título, el encabezado y el pie de la página web.

4.6. Seguridad en Páginas WEB.

Debido a la amplia gama de datos o información que se puede intercambiar a través del internet, actualmente y conforme avanza el uso de sistemas y redes informáticas se desarrollan más y mejores formas de proteger la información que se encuentra en el internet, esto en datos que están disponibles para usuarios seleccionados. Cada vez es más común que personas se dediquen a vulnerar la seguridad de sitios en la web, esto a fin de manipular información preciada para ellos, para terceros o simplemente para demostrar que pueden hacerlo. Con la finalidad de darle mayor seguridad a páginas web, que administren información delicada, se han desarrollado varias formas de bloquear usuarios y acciones no autorizados.

De hecho en ciertos sistemas a más de información confidencial es necesario proteger los recursos disponibles propios de la red, cuando estos son capaces de comandar o acceder a subsistemas de control. Para llevar una buena política de seguridad es imperativo implementar alguna medida que rechace accesos no autorizados, tal como se haría para la información y recursos de una entidad privada que cuente con una red conectada al internet (Figura 4.23). Obviamente el sistema de seguridad que se implemente debe conceder el acceso a los usuarios autorizados, tanto de manera local como remota.

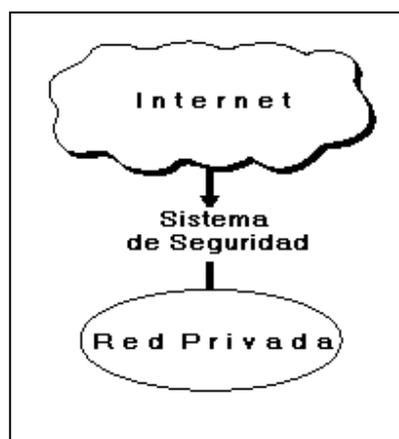


Figura 4.23. Interacción Red Privada y la Web.

Fuente: monografias.com

4.6.1. Firma Digital.

Cuando manejamos información delicada en un computador que no es de uso personal lo más conveniente es proteger el archivo o archivos a través de un bloqueo con clave, esto es más seguro cuanto mayor número de caracteres posea dicha clave. El método de la llamada firma digital se basa en este principio, pero con la combinación de dos claves. A un usuario autorizado, para acceder al sistema, se le asigna una clave pública o nombre de usuario la cual se puede conocer por cualquier visitante del sitio en internet, pero para cada clave pública existe una sola clave privada que valida o autentifica a la persona de pretende usar el instrumento virtual. De esta manera se puede conocer no solo las acciones que se realizan dentro de la página sino también el usuario que las está llevando a cabo.

Para que administrador y usuario puedan intercambiar mensajes electrónicos, de manera fiable, se necesita que existan tres condiciones. Identidad, que significa que haya la certeza de la autoría de cada mensaje electrónico. Integridad, que implica que cada mensaje enviado sea idéntico al recibido en el otro extremo. No repudiación que exige que el emisor de un mensaje no pueda, bajo ninguna circunstancia, negar que el mensaje fue generado por él.

Adicionalmente está la confidencialidad que no es un requisito, sino una ventaja del método y consiste en la fiabilidad de que el mensaje no esté disponible para nadie que no sea el emisor y el receptor, es decir que durante el proceso de transmisión no se derive hacia terceras personas.

El administrador del sitio es el único que puede otorgar una cuenta de participación dentro de la página web. Por lo que el acceso al instrumento es tan cerrado y restringido como lo decida el administrador. Adicional a la firma digital existe la posibilidad de denegar el acceso remoto a la página web cuando quien opera el servidor así lo decida, esto se logra alternado las propiedades del instrumento virtual en la parte inferior izquierda de la ventana de LABVIEW se indica la accesibilidad a través de la web.

CONCLUSIONES

Durante la realización del presente trabajo se han realizado diversas pruebas en varios ambientes y basándose en esas pruebas se pueden hacer las siguientes aseveraciones:

Se han alcanzado con éxito todos los objetivos planteados al inicio del proyecto, por lo cual se puede asegurar que el sistema, es superior a los equipos estándares actuales y los dispositivos que se asemejan en sus funciones y características usan significativamente mayores recursos tecnológicos, económicos y humanos.

A partir de esta investigación se pueden generar equipos similares con propósitos mas orientados hacia la domótica; incluso dado que se ha establecido una comunicación eficiente con el computador puede servir de base para el desarrollo de aplicaciones en instrumentación.

El funcionamiento relacionado con la línea telefónica es válido en lugares que tengan el servicio telefónico con líneas digitales, la marcación automática no es funcional para líneas analógicas. También se ha comprobado que el sistema se puede implementar cuando este funciona a través de una centralilla, claro que se necesitan mínimos cambios en hardware y software.

Cuando se utiliza una computadora de escritorio su dispositivo de comunicación serial, a través de su línea de tierra, puede introducir corrientes parasitas de rangos suficientes para interferir con otras funciones, del lado del receptor. El problema persiste aunque se utilice un adaptador de comunicaciones tipo USB. Esta interferencia se controla solamente con un opto acoplador de transmisión serial.

BIBLIOGRAFIA

ANGULO, José. - ANGULO, Ignacio. Microcontroladores PIC: Diseño Práctico de Aplicaciones. España. Editorial McGraw-Hill/Interamericana de España S.A. 2003.

ANGULO, José. - ANGULO, Ignacio. Microcontroladores PIC 16F87X: Diseño Práctico de Aplicaciones 2da parte. España. Editorial McGraw-Hill/Interamericana de España S.A. 2006.

BOYLESTAD, Robert. Electrónica: Teoría de Circuitos y Dispositivos Electrónicos. Octava edición. México. Editorial Pearson Prentice Hall. 2003.

LAZARO, Antonio. LABVIEW Gi Programación Gráfica para Control e Instrumentación. Editorial Paraninfo. 2001.

PEREZ, Leonel. Anotaciones de Microcontroladores PICS. 2004

QUINTERO, José. Sistemas de Control para Viviendas y Edificios: Domótica. España. Editorial Paraninfo. 1999.

TOCCI, Ronald. Sistemas Digitales: Principios y Aplicaciones. México. Editorial Prentice Hall. 2007

VALLEJO, Horacio. Telefonía, Principios, Funcionamientos. Argentina. Editorial Quark S.A. 1999.

REFERENCIAS ELECTRONICAS

MICROCHIP TECHNOLOGY Inc. PIC 16F870/871 Datasheet. microchip.com. U.S.A. 2003.

MICROCHIP TECHNOLOGY Inc. PICmicro Mid-Range MCU Family . microchip.com. U.S.A. 2003.

NATIONAL INSTRUMENTS. Tutorial: How to do a Serial Loopback Test. ni.com. U.S.A. 2003.

NATIONAL INSTRUMENTS. NI Vision Builder for Automated Inspection Development Toolkit Tutorial. ni.com. U.S.A. 2009.

NATIONAL INSTRUMENTS. USER GUIDE NI-IMAQ for USB Cameras. ni.com. U.S.A. 2004.

WINBOND ELECTRONICS CORP. ISD2570/75/90/120 Datasheet. winbond-usa.com. U.S.A. 2003

TEXAS INSTRUMENTS. MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS Datasheet. focus.ti.com. U.S.A. 2002.

MITEL SEMICONDUCTOR. MT 8870 Datasheet. www.mitel.com. U.S.A. 2001.

ANEXOS

Anexo A MANUAL DEL USUARIO

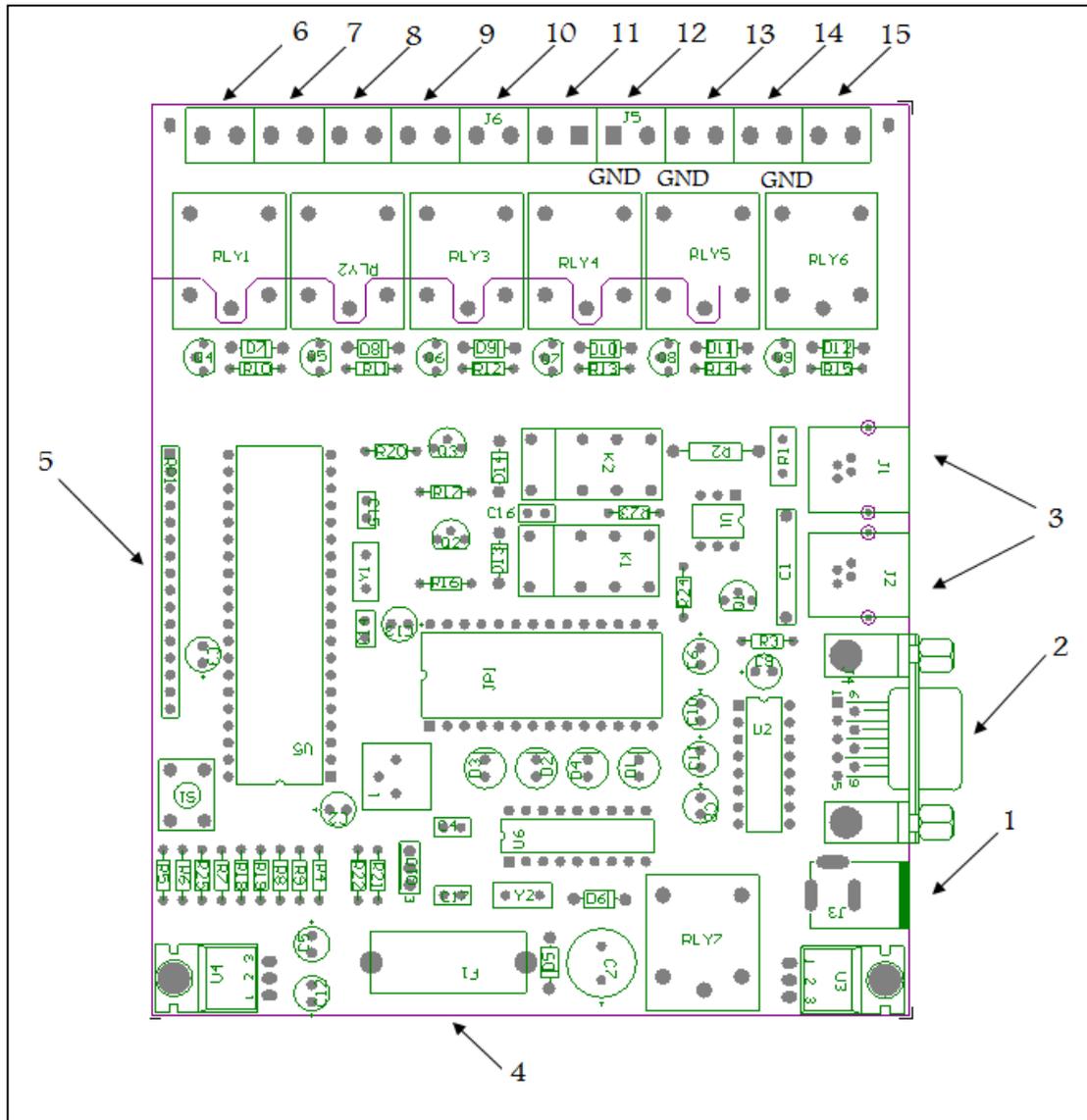


Figura A.1. Indicación de Entradas y Salidas en el PCB Terminado.

Fuente: Autores.

Instalación del Hardware.

Para un correcto funcionamiento del sistema completo es necesario conocer cada parte de las entradas y las salidas del mismo. A continuación se detallan todos los conectores, sean estos puertos de comunicación, *plugs* o borneras. En la figura A.1. Se muestran las partes numeradas del equipo. Aunque el diseño es fácil de usar, es indispensable que no haya confusión entre las entradas y salidas del dispositivo puesto que la manipulación errónea de sus recursos puede derivar en daños permanentes al equipo.

Elementos de Entrada y Salida.

1. Plug de alimentación.
2. Puerto serial tipo hembra.
3. Conectores telefónicos RJ-11.
4. Porta fusible.
5. Puerto de comunicación hacia el LCD.
6. Bornera doble para actuador 1.
7. Bornera doble para actuador 2.
8. Bornera doble para actuador 3.
9. Bornera doble para actuador 4.
10. Bornera doble para actuador 5
11. Bornera doble, salida hacia la bocina.
12. Bornera doble de entrada para la batería, negativo y positivo.
13. Bornera de salida de voltajes de izquierda a derecha GND y VDD respectivamente.
14. Bornera doble de izquierda a derecha, entrada para el sensor de movimiento y GND.
15. Bornera doble de izquierda a derecha, entrada para el sensor de magnético y VCC.

Descripción.

La fuente de alimentación que se recomienda es cualquiera no menor de 12 voltios de corriente continua con por lo menos 1 Ampere. Para alimentar al circuito se dispone de con conector tipo hembra (numeral 1 figura A.1). Simultáneamente a la fuente se conecta una batería de 9 voltios, típicamente usada en los sistemas de alarma convencionales. La batería se conecta a través de la bornera numerada 12 en A.1. Al momento de producirse una alarma la bocina se alimenta de la batería por lo que es necesaria, además, si se llegara a perder la energía en el toma corriente que alimenta a la fuente, la batería suplirá la función de la misma.

A continuación se debe conectar el terminal DB9 (#2 figura A.1.) del puerto serial hacia la computadora, al prescindir de este conector se interrumpirá la comunicación y por lo tanto se vuelve imposible el monitoreo y el control desde el internet. El puerto serial es de tipo hembra por lo que, obviamente, el cable hacia la computadora debe ser tipo macho ó utilizar un adaptador, pero si es el caso se debe comprobar que los terminales de transmisión y recepción sean correctos. Esto se puede llevar a cabo con una comunicación *loopback* tanto en el dispositivo como en el ordenador.

Los dos conectores RJ-11 (#3) se usan con cables telefónicos estándar, los dos terminales están en paralelo por lo que no es necesario guardar un orden específico para conectarlos. Con el uno se debe conectar el equipo hacia la línea telefónica, mientras que con el otro hacia el terminal telefónico.

El bus de comunicación, etiquetado "5" en el esquema A.1, se conecta al puerto tipo peineta y el otro extremo al LCD, para esto se debe tener en cuenta la orientación del mismo. El pin numero 1 está marcado en la placa.

Las borneras que corresponden a los actuadores 1 al 5 se conectan de manera análoga a la indicada en las figuras A.2 y A.3. Cada par de borneras (6, 7, 8, 9 y 10 en la figura A.1) están representadas por los contactos auxiliares de los relés.

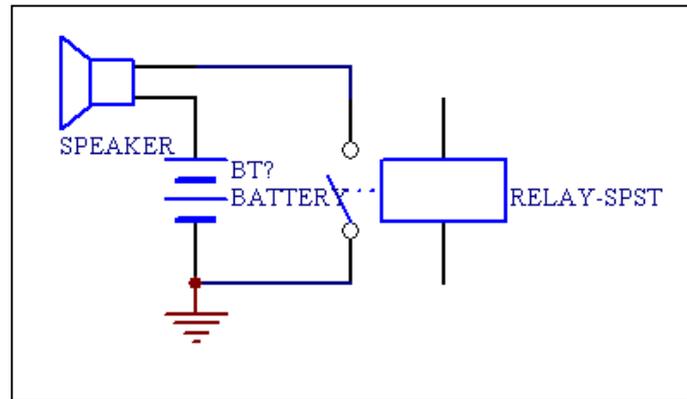


Figura A.2. Ejemplo de Conexión de un Actuador de Corriente Continua.

Fuente: Autores

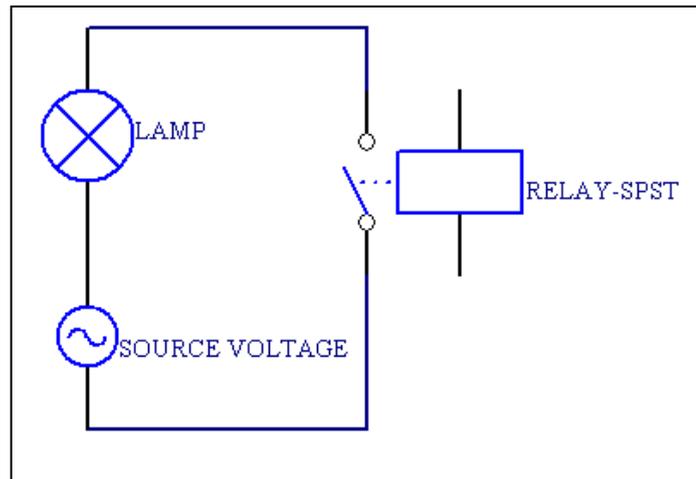


Figura A.3. Ejemplo de Conexión de un Actuador de Corriente Alterna.

Fuente: Autores.

Las borneras 13 y 15 contienen voltajes como VDD, VCC y GND (12V, 5V y tierra), estas salidas sirven como alimentaciones auxiliares para alimentar en caso de que actuadores trabajen con corriente directa o en caso de que los sensores necesiten más de un voltaje como suele suceder con algunos sensores de movimiento que requieren 12 y 5 voltios para funcionar. Las entradas de los sensores magnéticos y de movimiento son para recibir el voltaje que produzcan los mismos ya sean 0 o 5 (alto o bajo).

Instalación del Software.

Los primeros instaladores necesarios para el sistema son los *drivers* de la o las cámaras Web que se vayan a utilizar. Por lo general con cada cámara también se incluye un software para utilizarla, para la aplicación no es necesario el programa de cada cámara, basta con los controladores. En caso de no contar con un puerto serial integrado en la maquina en la que vaya a correr la aplicación es necesario el uso de un convertidor de estándar USB a serial tipo DB9 y este dispositivo también necesita de la previa instalación de *drivers*.

Una vez instalados los controladores de los elementos del sistema se abre el archivo ya sea desde LABVIEW o a su vez el archivo ejecutable creado a partir del panel Frontal. En el primer caso se debe tener instalado LABVIEW PROFESSIONAL 8.5 o superior junto con el paquete de visión artificial y el *toolkit* de adquisición USB.

Al tener abierto el instrumento virtual lo único que se configura es el número de puerto serial que se va a utilizar (típicamente 3, 4 ó 5). El control se muestra en la figura A.4. Numeral 4.

Para correr el programa solamente se debe hacer click en el botón *play* ilustrado en la figura A.4. Numeral 1. Este control está presente ya sea en la aplicación local o remota desde un navegador de internet. Cabe aclarar que el instrumento no empieza a adquirir ni controlar hasta que se haya validado la cuenta de usuario llenando los espacios numerados 2 y 3 en la figura A.4. Para moverse entre el panel que controla e indica el estado de los recursos del sistema basta con elegir entre las pestañas “*status*” o “*video*” presentes en la grafica como 5 y 6, respectivamente. Para detener el instrumento se debe hacer click en *stop* (número 7 en la grafica A.4.).

El instrumento virtual se actualiza continuamente, por lo que si se interrumpiera el funcionamiento del mismo o de la computadora, por algún evento externo, se puede reiniciar el VI sin que esto provoque ningún contratiempo en el resto del sistema.

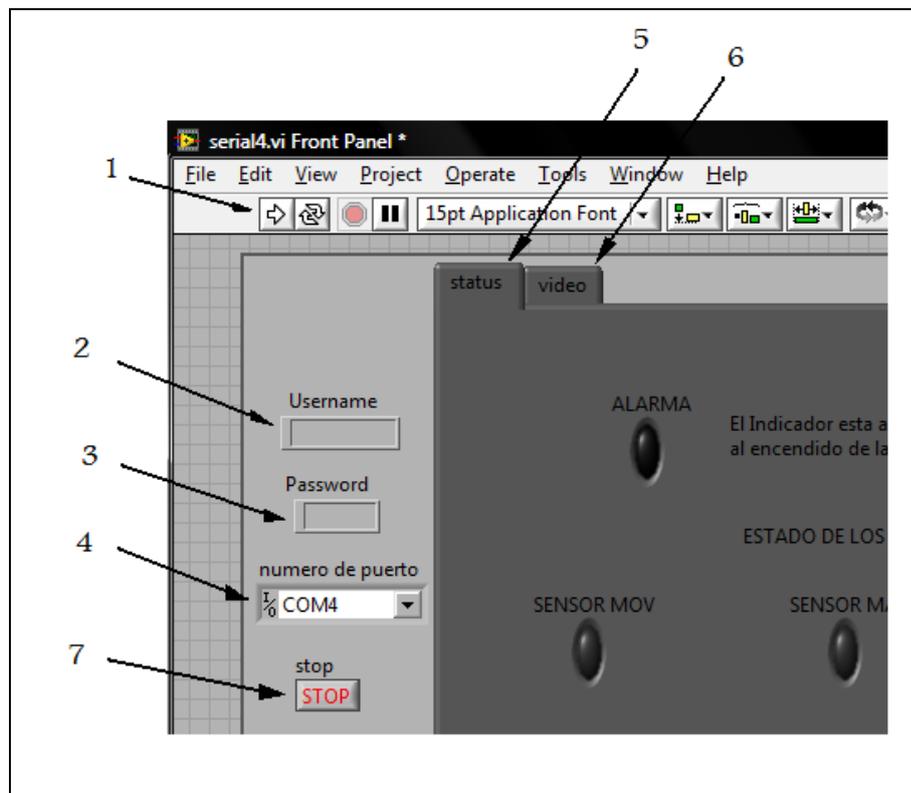


Figura A.4. Partes Relevantes del Instrumento Virtual.

Fuente: Autores.

Los botones ubicados en la parte inferior del instrumento virtual están destinados al control individual y secuencial de los actuadores. Debido al volumen de información que significa la manipulación de video existe un pequeño lapso de tiempo (menor que 1 segundo) entre el comando desde la computadora y la respuesta sobre los indicadores, cabe aclarar que este intervalo de tiempo no está presente en la ejecución solo en su indicación. Un click en los actuadores 1 al 4 provoca un cambio de estado sobre su salida. Mientras que las secuencias operan de la siguiente forma:

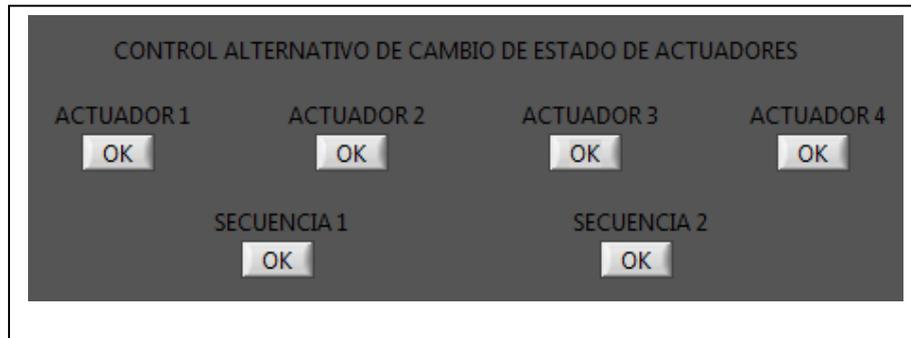


Figura A.5. Sección del Instrumento que Comanda los Actuadores.

Fuente: Autores.

- **Secuencia 1:**

Actuador 1 ON.

Demora: 5 segundos.

Actuador 2 ON.

Demora: 4 segundos.

Actuador 1 OFF.

Demora: 3 segundos

Actuador 3 ON

Demora: 5 segundos

Actuador 2 OFF

Actuador 3 OFF

- **Secuencia 2:**

Actuador 3 ON.

Demora: 3 segundos.

Actuador 3 OFF.

Demora: 5 segundos.

Actuador 2 ON.

Demora: 3 segundos.

Actuador 2 OFF.

Operaciones Básicas Desde el Teléfono.

Valores por Defecto.

Inicialmente al encender el equipo el LCD muestra un asterisco "*" y permanece con la luz de fondo (*backlight*) apagada, el dispositivo está listo para utilizarse, pero son los valores por defecto. Por ejemplo la clave grabada en la memoria por defecto es 123456. También en los valores de teléfono fijo y teléfono celular están ocupados, pero deben ser reemplazados desde la primera vez que se utilice el equipo.

Para utilizar las opciones del dispositivo se comienza por levantar el teléfono y presionar el "*" el LCD guía cada acción visualizando varios menús. Todas las opciones elegibles dentro de los menús están acompañadas por la tecla correspondiente que las activan.

Cambio de Clave.

Los siguientes pasos conducen a cambiar la clave del sistema.

1. Levantar el teléfono.
2. Pulsar *.
3. Pulsar # para activar la opción "CONFIGURACION".
4. Pulsar * para activar "CLAVE".
5. Ingresar la clave antigua.
6. Si la clave es correcta se ingresa la nueva clave, sino automáticamente se regresa al menú inicial.

Ingreso del Teléfono Fijo.

1. Levantar el teléfono.
2. Pulsar *.

3. Pulsar # para activar la opción "CONFIGURACION".
4. Pulsar # para activar la opción "TELEFONOS".
5. Ingresar la clave.
6. Pulsar # para activar la opción "FIJO".
7. Ingresar el número de teléfono fijo.
8. Si el numero es correcto presionar * para aceptar el numero, si el numero es incorrecto presionar # para cancelar.

Ingreso del Teléfono Móvil.

1. Levantar el teléfono.
2. Pulsar *.
3. Pulsar # para activar la opción "CONFIGURACION".
4. Pulsar # para activar la opción "TELEFONOS".
5. Ingresar la clave.
6. Pulsar * para activar la opción "CELULAR".
7. Ingresar el número de teléfono móvil.
8. Si el numero es correcto presionar * para aceptar el numero, si el numero es incorrecto presionar # para cancelar.

Control de los Actuadores y Secuencias.

1. Levantar el teléfono.
2. Pulsar *.
3. Pulsar 0 para activar la opción "ACTUADORES".
4. Ingresar la clave.
5. Pulsar la tecla correspondiente a cada actuador (1, 2, 3, ó 4) o 5 ó 6 para iniciar las secuencias. Después de realizar los cambios de estado el LCD muestra el cartel "HECHO" y a continuación se regresa al menú principal.

Armado.

1. Levantar el teléfono.
2. Pulsar *.
3. Pulsar * para activar la opción “ARMAR SISTEMA”.
4. Ingresar la clave. Si la clave es correcta aparece el cartel “HECHO”, se apaga el *backlight* y el sistema queda armado. Si la clave es incorrecta se regresa al menú principal y el sistema no se arma.

Desarmado.

1. Levantar el teléfono.
2. Pulsar *.
3. Pulsar * para activar la opción “DESARMAR SISTEMA”.
4. Ingresar la clave. Si la clave es correcta aparece el cartel “HECHO”, el sistema queda desarmado y de regreso al menú principal. Si la clave es incorrecta el sistema no se desarma y se regresa al menú anterior.

NOTA: si cualquiera de las opciones se realizan de manera remota, el único cambio es que después de llamar a la línea conectada al sistema, el teléfono se levanta de manera automática y se indica con un mensaje de bienvenida. A partir del mensaje se comienza con los pasos antes descritos según la función que se desee realizar.

Mal Funcionamiento.

Al utilizar adquisición de video a través de LABVIEW durante tiempos muy prolongados se pueden producir errores típicos, que provocan cortes en la adquisición pero que no interrumpen el funcionamiento general del instrumento virtual, estos errores se muestran en la tabla A.1.

Error Code	Description
-1074396032	Not enough memory.
-1074396030	Timeout error.
-1074396028	Internal error. Camera Manufacturer property page produced an error.
-1074396026	Not a valid USB session.
-1074396024	System memory error.
-1074396023	Feature not available in camera. Camera Manufacturer property page produced an error.
-1074396020	Filter name not found.
-1074396018	No acquisition in progress.
-1074396017	Invalid value on Property Page. Camera Manufacturer property page produced an error.
-1074396016	Multiple camera acquisition not supported.
-1074396015	Not able to connect to camera, may be in use.
-1074396014	Invalid image. Make sure the image is not NULL.
0	No error.

Tabla A.1. Identificación de errores de Adquisición de Video.

Fuente: *USER GUIDE NI-IMAQ for USB Cameras.*

NOTA: En caso de perder la clave o necesitar interrumpir el funcionamiento del equipo se debe resetear el equipo, para ello hay un micro pulsante junto al conector del LCD.

Anexo B
LISTA DE MATERIALES.

Part Type	Designator	Footprint	Description
0.1uf	C4	RAD0.1M	Capacitor
0.1uf	C3	RB.1/.2M	Bi-Polar Capacitor
1N4004	D14	DIODE0.2M	Diode
1N4004	D13	DIODE0.2M	Diode
1N4007	D12	DIODE0.2M	Diode
1N4007	D9	DIODE0.2M	Diode
1N4007	D11	DIODE0.2M	Diode
1N4007	D8	DIODE0.2M	Diode
1N4007	D7	DIODE0.2M	Diode
1N4007	D6	DIODE0.2M	Diode
1N4007	D5	DIODE0.2M	Diode
1N4007	D10	DIODE0.2M	Diode
1uf	C2	RB.1/.2M	Bi-Polar Capacitor
2N3904	Q6	BJT_1	NPN BJT
2N3904	Q7	BJT_1	NPN BJT
2N3904	Q5	BJT_1	NPN BJT
2N3904	Q8	BJT_1	NPN BJT
2N3904	Q2	BJT_1	NPN BJT
2N3904	Q1	BJT_1	NPN BJT
2N3904	Q4	BJT_1	NPN BJT
2N3904	Q9	BJT_1	NPN BJT
2N3904	Q3	BJT_1	NPN BJT
3.5795MHZ	Y2	XTAL1	Crystal Oscillator
4.7k	R6	AXIAL0.2M	Resistor
4.7uf	C6	RB.1/.2M	Bi-Polar Capacitor
4N25	U1	DIP6_300M	Opto Isolator
5.1k 1/4w	R8	AXIAL0.2M	Resistor
5.6k/1w	R2	AXIAL0.6M	Resistor
5VSPDT	RLY1	RELE_SPDT	SPDT Relay
5VSPDT	RLY2	RELE_SPDT	SPDT Relay
5VSPDT	RLY5	RELE_SPDT	SPDT Relay
5VSPDT	RLY6	RELE_SPDT	SPDT Relay
5VSPDT	RLY4	RELE_SPDT	SPDT Relay
5VSPDT	RLY3	RELE_SPDT	SPDT Relay
10k	R20	AXIAL0.2M	Resistor
10uf	C12	RB.1/.2M	Bi-Polar Capacitor
10uf/16V	C13	RB.1/.2M	Bi-Polar Capacitor
10uf /16v	C8	RB.1/.2M	Bi-Polar Capacitor

10uf /16v	C9	RB.1/.2M	Bi-Polar Capacitor
10uf /16v	C10	RB.1/.2M	Bi-Polar Capacitor
10uf /16v	C11	RB.1/.2M	Bi-Polar Capacitor
12VSPDT	RLY7	RELE_SPDT	SPDT Relay
16.000MHZ	Y1	XTAL1	Crystal Oscillator
16PIN	RP1	HDR1X16M	Peineta
18pf	C15	RAD0.1M	Capacitor
18pf	C14	RAD0.1M	Capacitor
22uf	C5	RB.1/.2M	Bi-Polar Capacitor
47	R24	AXIAL0.2M	Resistor
78L05	U4	REG_H2	Positive Voltage Regulator
78L12	U3	REG_H2	Positive Voltage Regulator
100k 1/4w	R21	AXIAL0.2M	Resistor
100k 1/4w	R22	AXIAL0.2M	Resistor
100nf	C16	RAD0.1M	Capacitor
100nf	C17	RAD0.1M	Capacitor
220	R4	AXIAL0.2M	Resistor
220	R3	AXIAL0.2M	Resistor
330	R18	AXIAL0.2M	Resistor
330	R19	AXIAL0.2M	Resistor
330	R9	AXIAL0.2M	Resistor
330	R17	AXIAL0.2M	Resistor
330	R13	AXIAL0.2M	Resistor
330	R12	AXIAL0.2M	Resistor
330	R16	AXIAL0.2M	Resistor
330	R10	AXIAL0.2M	Resistor
330	R15	AXIAL0.2M	Resistor
330	R14	AXIAL0.2M	Resistor
330	R11	AXIAL0.2M	Resistor
390k 1/4w	R25	AXIAL0.2M	Resistor
470	R5	AXIAL0.2M	Resistor
470k 1/4w	R7	AXIAL0.2M	Resistor
470nf/250V	C1	RAD0.7M	Capacitor
560	R23	AXIAL0.2M	Resistor
2200uf/25V	C7	RB.2/.4M	Bi-Polar Capacitor
CON2	J3	PWR_JACK	Connector
CON4	J2	RJ11	Connector
CON4	J1	RJ11	Connector
CON8	J5	BOR_8	Connector
CON12	J6	BOR_12	Connector
DB9	J4	DB9	Conector
FUSE1	F1	FUSE2	Fuse
HEADER 14X2	JP1	DIP28_600M	Zocalo

LED0	D4	LED	light Emitting Diode
LED0	D1	LED	light Emitting Diode
LED0	D2	LED	light Emitting Diode
LED0	D3	LED	light Emitting Diode
MAX232ACSE(16)	U2	DIP16_300M	RS-232 TRANSCEIVERS
MT88L70AC(18)	U6	DIP18_300M	Integrated DTMF Receiver
NPN	Q10	BD135	NPN Transistor
16F871	U5	DIP40_600M	40-Pin 16F871 Microcontroller
RELAY-DPDT	K1	RELE_DPDT	Rele Doble
RELAY-DPDT	K2	RELE_DPDT	Rele Doble
5.6K	R1	RAD0.3M	Resistor
5K	1	POT	Potentiometer
SW-PB	S1	PULS	Pulsante

Anexo C
SUBROUTINA QUE ARMA EL SISTEMA.

```

;=====
;SUBROUTINA QUE ARMA EL SISTEMA
;=====
armar      call Vclave      ;subrutina que verifica la clave
           btfsc flag,2
           goto start0     ;regreso al stand by
arma       call salir      ;apaga el timer y limpia registros
           call off0
           call C_des
           call Cact

armar1     call trans
           btfsc flag1,3
           goto ringg
           btfsc 6,0       ;testeo de RB0 sensor principal
           call prealarma  ;tiempo preventivo
           btfsc 9,0       ;sensores secundarios
           call alarma
ringg      btfss 6,6       ;testea entrada de RING
           goto armar1

           call time3      ;6s
doss       btfss flag,0    ;bandera para salir luego del time
           goto doss
           call on
           call dem_1s
           call dem_1s
           bsf 9,1         ;on ISD.
           movlw d'11'
           movwf cont
           call ISD
           bcf 9,1        ;off ISD.
           bcf 6,4
           call Vclave
           btfsc flag,2
           goto loco
           call salir ;apaga el timer y limpia registros
           call C_des
           call Cact

           call time
           call test
           movf clave,w
           sublw b'00001011' ;* Desarmar Sistema
           btfsc 3,2
           goto onn
           movf clave,w

```

```
sublw b'00001010'      ;0 Actuadores
btfsc 3,2
call actua
goto arma

loco    call dem_1s      ;en caso de clave incorrecta
        call dem_1s
        bsf 9,1          ;on ISD.
        movlw d'13'
        movwf cont
        call ISD
        bcf 9,1          ;off ISD.
        bcf 6,4
        goto arma
```

Anexo D

SUBRUTINAS DE CONFIGURACION DEL SISTEMA.

```

;=====
;SUBROUTINA CONFIGURA
;=====
configura  call Vclave
           btfsc flag,2      ;bandera verifica clave correcta
           return
           call salir      ;off tmr1 y salgo de la int
           call Cclave0
           call Ctelef
           call test
           movf clave,w
           sublw b'00001011'    ;* "clave"
           btfsc 3,2
           call Clave
           movf clave,w
           sublw b'00001100'    ;# "confuguracion"
           btfsc 3,2
           call configural
           return
;=====
;SUBROUTINA CONFIGURA 1
;=====
configural call Ccelu0
           call Cfijo0
           call test
           movf clave,w
           sublw b'00001011'    ;* "celular"
           btfsc 3,2
           call Celu
           movf clave,w
           sublw b'00001100'    ;# "fijo"
           btfsc 3,2
           call Fijo
           return
;=====
;SUBROUTINA CLAVE
;=====
Clave     call Cvieja
           call Vclave
           btfsc flag,2
           return
           call salir ;off tmr1 y salgo de la int
Clave0    call Cnueva
           call Cclave
           call Dclave
           call guion
           call Dclave
otraC     clrf dato_w

```

```

    call test
    movf clave,w
    movwf dato_w
    call w_eeeprom
    call Caster
    incf dir
    decfsz numero
    goto otraC
    call verif
    btfsc flag,0
    goto Clave0
    return
;=====
;SUBROUTINA CELULAR
;=====
Celu      call Ccelu
          call Dcelu
          call guion
          call Dcelu
otraCL    clrfsf dato_w
          call test
          call visualizar
          movf clave,w
          movwf dato_w
          call w_eeeprom
          incf dir
          decfsz numero
          goto otraCL
          call verif
          btfsc flag,0
          goto Celu
          return
;=====
;SUBROUTINA FIJO
;=====
Fijo      call Cfijo
          call Dfijo
          call guion
          call Dfijo
otraF     clrfsf dato_w
          call test
          call visualizar
          movf clave,w
          movwf dato_w
          call w_eeeprom
          incf dir
          decfsz numero
          goto otraF
          call verif
          btfsc flag,0
          goto Fijo
          return

```

Anexo E

SUBROUTINA DE CAMBIO DE ESTADO DE ACTUADORES.

```

;=====
;SUBROUTINA ACTUADORES
;=====
actua0    call Vclave
          btfsc flag,2
          return
          call salir      ;off tmr1 y salgo de la int
          call time      ;tiempo aproximado de 10s
actua     btfsc flag,1   ;bandera de retorno
          goto lban
          call C12345
PRUEBA    call test
          movf clave,w
          sublw b'00001010' ;0 salir
          btfsc 3,2
          return
          movf clave,w
          sublw d'1'
          btfsc 3,2
          call actuador1 ;actuador 1
          movf clave,w
          sublw d'2'
          btfsc 3,2
          call actuador2 ;actuador 2
          movf clave,w
          sublw d'3'
          btfsc 3,2
          call actuador3 ;actuador 3
          movf clave,w
          sublw d'4'
          btfsc 3,2
          call actuador4 ;actuador 4
          movf clave,w
          sublw d'5'
          btfsc 3,2
          call actuador5 ;actuador 5
          goto actua

actuador1 btfsc 9,2
          goto off
          bsf BANDERA,2      ;0n actuador 1
          call C_hecho
          return
off       bcf BANDERA,2      ;off actuador 1
          call C_hecho
          return
actuador2 btfsc 6,1
          goto of1

```

```

        bsf BANDERA,3          ;0n actuador 2
        call C_hecho
        return
of1     bcf BANDERA,3          ;off actuador 2
        call C_hecho
        return
actuador3 btfsc 6,2
        goto of2
        bsf BANDERA,4          ;0n actuador 3
        call C_hecho
        return
of2     bcf BANDERA,4          ;off actuador 3
        call C_hecho
        return
actuador4 btfsc 6,3
        goto of3
        bsf BANDERA,5          ;0n actuador 4
        call C_hecho
        return
of3     bcf BANDERA,5          ;off actuador 4
        call C_hecho
        return
actuador5 btfsc 6,4
        goto of4
        bsf BANDERA,6          ;0n actuador 5
        call C_hecho
        return
of4     bcf BANDERA,6          ;off actuador 5
        call C_hecho
        return
```

Anexo F
SUBROUTINA DE ALARMA.

```

;=====
;SUBROUTINA ALARMA
;=====
alarma    bsf flag1,3           ;indica intruso
          movlw 'q'
          call transmite
          btfss 6,0
          goto mag
          movlw 'c'
          call transmite
mag       btfss 9,0
          goto llama
          movlw 'a'
          call transmite
llama    call on
          call Dfijo
          call dem_1s

          call alarma1
          call on
          call Dcelu
          call dem_1s

          call alarma1
          bsf 6,5             ;ON SIRENA
          return

alarma1  bsf 9,1             ;conecta ISD.
otro_1   call r_eeeprom
          movf dato_r,w
          movwf cont
          call visualizar
          call ISD
          incf dir
          decfsz numero
          goto otro_1
          call time          ;demora de 10s espera q conteste
dos1     btfss flag,0
          goto dos1
          movlw d'4'
          movwf numero      ;numero de veces

siga     movlw d'12'
          movwf cont        ;numero de alarma
          call ISD
          call dem_1s
          decfsz numero
          goto siga

```

```
bcf 9,1          ;off ISD.  
bcf 6,4  
call off0  
  
call dem_1s  
return
```

Anexo G**SUBROUTINA DE REPRODUCCION CON EL ISD2560.**

```

;=====
;SUBROUTINA DE CONTROL DEL ISD.
;=====
ISD
    bsf 6,4           ;conecta resistencia
    bsf 5,2           ;PD reset a la dirección del isd.
    call dem_5ms
    bcf 5,2
    movf cont,w
    sublw d'4'
    btfss 3,2
    goto next5
    movlw d'14'
    movwf cont
next5
    movf cont,w
    sublw d'0'
    btfss 3,2
    goto next
    movlw d'10'
    movwf cont
next
    decfsz cont
    goto otro
    bcf 5,0           ;M0 habilita la reproducción.
play
    bcf 5,1           ;CE inicia la reproducción.
    call dem_50ms
    bsf 5,1
EOM
    btfsc 5,3         ;EOM indica fin de mensaje.
    goto EOM
    return

otro
    bsf 5,0           ;M0 deshabilita reproducción.
    call play
    goto next

```

Anexo H

SUBROUTINA DE IDENTIFICACION DE TONOS.

```
=====
;SUBROUTINA VERIFICA LLEGADA DE NUMEROS (DTMF)
;=====
test      btfsc flag,0      ;bandera para salir luego del time3
          return
          clrf clave
          btfss 7,5        ;RC5 SID MT8870
          goto test
test1     btfsc 7,5        ;RC5 SID MT8870
          goto test1
          btfsc 7,1
          bsf clave,0
          btfsc 7,2
          bsf clave,1
          btfsc 7,3
          bsf clave,2
          btfsc 7,4
          bsf clave,3
          movf clave,w
          return
```

Anexo I
SUBROUTINAS DE COMUNICACIÓN SERIAL.

```

;-----
;subrutina de transmision
;-----
transmite  movwf TXREG
           nop
           nop
           call dem_5ms
           return
;-----
;subrutina de recepcion
;-----
recibe     movf RCREG,w
           movwf buffer
           ;bcf RCSTA,4
           call actualiza
           call trama
           ;call evalua
sal        clr f RCREG
           ;bsf RCSTA,4
           Return
;-----
;subrutina que actualiza el registro despues de la recepcion
;-----
actualiza  movf buffer,w
           sublw 'a'
           btfsc 3,2
           goto act1
           movf buffer,w
           sublw 'b'
           btfsc 3,2
           goto act2
           movf buffer,w
           sublw 'c'
           btfsc 3,2
           goto act3
           movf buffer,w
           sublw 'd'
           btfsc 3,2
           goto act4
           movf buffer,w
           sublw 'e'
           btfsc 3,2
           goto act5
           movf buffer,w
           sublw 'f'
           btfsc 3,2
           goto act6

```

```

                                return
;-----
;subrutina que complementa la actualizacion
;-----
act1      btfsc BANDERA,2
          goto off1
          goto on1
act2      btfsc BANDERA,3
          goto off2
          goto on2
act3      btfsc BANDERA,4
          goto off3
          goto on3
act4      btfsc BANDERA,5
          goto off4
          goto on4

act5      bsf 9,2          ;SECUENCIA 1
          call dem_1s
          call dem_1s
          call dem_1s
          bsf 6,1
          call dem_1s
          call dem_1s
          call dem_1s
          bcf 9,2
          call dem_1s
          call dem_1s
          call dem_1s
          bsf 6,2
          call dem_1s
          call dem_1s
          bcf 6,2
          bcf 6,1
          return

act6      bsf 6,2
          call dem_1s
          call dem_1s
          call dem_1s
          bcf 6,2
          call dem_1s
          call dem_1s
          call dem_1s
          bsf 6,1
          call dem_1s
          call dem_1s
          call dem_1s
          bcf 6,1
          return

off1      bcf BANDERA,2

```

```

        return
on1      bsf BANDERA,2
        return
off2     bcf BANDERA,3
        return
on2      bsf BANDERA,3
        return
off3     bcf BANDERA,4
        return
on3      bsf BANDERA,4
        return
off4     bcf BANDERA,5
        return
on4      bsf BANDERA,5
        return
;=====
;SUBROUTINA TRAMA
;=====
trama    btfsc BANDERA,2      ;actuador 0
        goto onn2
        bcf 9,2
sig2     btfsc BANDERA,3      ;actuador 1
        goto onn3
        bcf 6,1
sig3     btfsc BANDERA,4      ;actuador 2
        goto onn4
        bcf 6,2
sig4     btfsc BANDERA,5      ;actuador 3
        goto onn5
        bcf 6,3
sig7     btfsc 6,0            ;sensor mov
        goto onn6
        bcf BANDERA,0
sig8     btfsc 9,0            ;alarma
        goto onn7
        bcf BANDERA,1
        return
;=====
onn2     bsf 9,2
        goto sig2
onn3     bsf 6,1
        goto sig3
onn4     bsf 6,2
        goto sig4
onn5     bsf 6,3
        goto sig7
onn6     bsf BANDERA,0
        goto sig8
onn7     bsf BANDERA,1
        return
;-----
;subrutina para evaluar el registro y transmitirlo

```

```

;-----
evalua    btfsc BANDERA,0
          goto pri
          goto nopri
dos       btfsc BANDERA,1
          goto seg
          goto nosej
tres      btfsc BANDERA,2
          goto ter
          goto noter
cuatro    btfsc BANDERA,3
          goto cuar
          goto nocuar
cinco     btfsc BANDERA,4
          goto quin
          goto noquin
seis      btfsc BANDERA,5
          goto sext
          goto nosext
siet      btfsc BANDERA,6
          goto sept
          goto nosept
ocho      btfsc BANDERA,7
          goto oct
          goto noct
nueve     btfsc flag1,3
          goto nov
          goto nonov
;-----
;complemento de la subrutina que envia la evaluacion
;-----
pri       movlw 'a'
          call transmite
          goto dos
nopri     movlw 'b'
          call transmite
          goto dos
seg       movlw 'c'
          call transmite
          goto tres
nosej     movlw 'd'
          call transmite
          goto tres
ter       movlw 'e'
          call transmite
          goto cuatro
noter     movlw 'f'
          call transmite
          goto cuatro
cuar      movlw 'g'
          call transmite
          goto cinco

```

```
nocuar    movlw 'h'  
          call transmite  
          goto cinco  
quin     movlw 'i'  
          call transmite  
          goto seis  
noquin   movlw 'j'  
          call transmite  
          goto seis  
sext     movlw 'k'  
          call transmite  
          goto siet  
nosext   movlw 'l'  
          call transmite  
          goto siet  
sept     movlw 'm'  
          call transmite  
          goto ocho  
nosept   movlw 'n'  
          call transmite  
          goto ocho  
oct      movlw 'o'  
          call transmite  
          goto nueve  
noct     movlw 'p'  
          call transmite  
          goto nueve  
nov      movlw 'q'  
          call transmite  
          return  
nonov    movlw 'r'  
          call transmite  
          return
```