



FACULTAD DE INGENIERIAS

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE: INGENIERO ELECTRÓNICO CON MENCIÓN EN SISTEMAS COMPUTACIONALES

TEMA

**CONTROL DE COMPONENTES ELÉCTRICOS Y
ELECTRÓNICOS POR MEDIO DE COMANDOS DE VOZ**

INTEGRANTES:

ELOY REA ESCALANTE

LUIS PINOS PUMA

TUTOR: ING. RAÚL ÁLVAREZ GUALE

GUAYAQUIL – ECUADOR

2011

DEDICATORIA

Dedico este proyecto de tesis a Dios por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mis padres por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

Eloy Rea Escalante

Dedicado a mis padres que creyeron en mí y nunca me permitieron rendirme ante nada y a todas las personas que me ayudaron a lo largo de mis años de estudios

Luis Pinos Puma

AGRADECIMIENTO

Agradecemos a Dios por darnos la fuerza y el entusiasmo de culminar con éxitos nuestra carrera universitaria.

También agradecemos a nuestro tutor Ing. Raúl Álvarez por darnos toda la asesoría necesaria para poder realizar la tesis.

A nuestros padres por enseñarnos que la perseverancia y el esfuerzo son el camino para lograr objetivos.

Y a todas aquellas personas que de una u otra forma, colaboraron en la realización de este proyecto, hacemos extensivo nuestro más sincero agradecimiento

Eloy Rea Escalante

Luis Pinos Puma

ÍNDICE GENERAL

INTRODUCCIÓN	11
ABSTRACT	11
CONTENIDO DE LOS CAPÍTULOS	12
JUSTIFICACIÓN	13
SOLUCIÓN	13
OBJETIVOS	14
DIAGRAMA DE FUNCIONAMIENTO DEL SISTEMA	15
DISEÑO DE LA MÁQUINA INDUSTRIAL	16
FUNCIONAMIENTO DEL SISTEMA.....	16
ALCANCE DEL PROYECTO	17
LIMITACIONES DEL PROYECTO	17

CAPÍTULO I

DESCRIPCIÓN DE HARDWARE

1.1 Puerto Serial	18
1.1.1 Características del puerto serial.....	18
1.1.2 Tipos de comunicación en serie	19
1.1.3 Señales del puerto serial	19
1.2 Microcontrolador PIC	20
1.2.1 Pics más usados en el mercado.....	20
1.2.2 Pics usados en el proyecto	21
1.2.2.1 PIC 16F870/871	21
1.2.2.2 PIC 16F628.....	22
1.3 XBee	24
1.3.1 Características de los xbee	24
1.3.2 Tipos de Xbee	25
1.4 Sensor Capacitivo	26
1.4.1 Ventajas y desventajas.....	26
1.4.1.1 Ventajas.....	26
1.4.1.2 Desventajas	27
1.5 Relé.....	27
1.5.1 Funcionamiento del relé	28
1.5.2 Tipos de Relé	28

1.5.3 Características de un relé.....	28
1.5.4 Características técnicas.....	29
1.5.4.1 Electromagnéticas.....	29
1.5.4.2 Contactos	29
1.6 Luces LED	29
1.6.1 Componentes del LED.....	20
1.7 Pantalla cristal líquido (LCD)	31
1.7.1 Características de los pines del LCD.....	31
1.8 Cámara Web	33
1.8.1 Componentes de la cámara Web	34

CAPÍTULO II

DESCRIPCIÓN DE SOFTWARE

2.1 Visual Basic.....	35
2.1.1 Historia	36
2.1.2 Versiones	37
2.1.3 Características.....	37
2.1.4 Entorno de desarrollo.....	38
2.1.4.1 Ejemplo de código	39
2.1.5 Ventajas	39
2.1.6 Inconvenientes	40
2.2 Reconocimiento del habla	40
2.2.1 Decodificador acústico-fonético	41
2.2.2 Modelo del lenguaje.....	41
2.2.3 Usos y aplicaciones.....	41
2.2.4 Sistemas diseñados para discapacitados.....	42
2.3 Programa de reconocimiento de voz Dragon Naturally Speaking	42
2.4 Lenguaje SQL	43
2.4.1 Sistema de Gestor de base de datos	44
2.5 MySQL.....	44
2.5.1 Especificaciones del código fuente.....	45
2.5.2 Características.....	45

2.6 Microcode Studio Plus	46
2.7 Proteus	47
2.7.1 ISIS	48
2.7.2 Ares	48
2.8 Pickit 2 Plus	49
2.8.1 Características generales	50

CAPÍTULO III

DISEÑO DEL SISTEMA

3.1 Diagrama de flujo del funcionamiento del sistema	51
3.2 Diseño del Software	52
3.2.1 Mantenimiento del sistema	53
3.2.2 Control del Sistema.....	55
3.2.3 Instrucciones del sistema.....	57
3.3 Diseño de la Base de Datos	58
3.3.1 Configuración ODBC Para MySQL.....	59
3.4 Diseño del Hardware	60
3.4.1 Módulo de Transmisión.....	60
3.4.2 Módulo de Recepción.....	62
3.5 Pruebas de funcionamiento de los módulos Xbee (Transmisión y Recepción)	71
3.6 Diseño de Circuitos Impresos	72
3.7 Diseño del Modulo de transmisión y recepción en proteus	75

CAPÍTULO IV

IMPLEMENTACIÓN DEL SISTEMA

4.1 Implementación del módulo de transmisión	77
4.2 Implementación Modulo Recepción	78
4.3 Implementación Luces Piloto.....	80
4.4 Implementación LCD.....	82
4.5 Implementación sensor	83
4.6 Armado de la estructura de la máquina Transportadora.....	84

CONCLUSIONES Y RECOMENDACIONES

Conclusiones	88
Recomendaciones	89
ANEXOS	90
BIBLIOGRAFÍA	141

ÍNDICE DE FIGURAS

Figura A: Estructura básica del sistema	15
Figura B: Diseño de la Máquina transportadora	16
1.1 Conector Serial RS-232	18
1.2 Pines RS232.....	19
1.3 PIC	20
1.4 PIC 16F871	22
1.5 PIC 16F628	23
1.6 Módulos XBee	24
1.7 Sensor de Capacitivo.....	26
1.8 Relé	27
1.9 LED / Luz Piloto	29
1.10 Pantalla LCD	31
1.11 Caracteres soportados por el LCD	32
1.12 Cámara Web.....	33
2.1 Microsoft Visual Basic.....	35
2.2 Opciones Microsoft Visual Basic	38
2.3 Software de control de Voz	42
2.4 Barra Herramientas de Dragón Natural Speaking	42
2.5 MySQL.....	44
2.6 Microcode Studio.....	46
2.7 Pantalla principal Microcode Studio	46
2.8 Proteus.....	47
2.9 Ares.....	48
2.10 Pickit 2	49

3.1 Diagrama de Flujo principal.....	51
3.2 Pantalla de usuario y contraseña.....	52
3.3 Pantalla principal	52
3.4 Pantalla de menú.....	53
3.5 Búsqueda de usuario.....	54
3.6 Actualización de usuario.....	55
3.7 Menú Inicio	55
3.8 Error de COM1	56
3.9 Puerto Com Abierto.....	56
3.10 Pantalla de control del sistema.....	57
3.11 Bloqueo del sistema	58
3.12 Base de Datos Mysql.....	58
3.13 Cable Serial RS-232	60
3.14 Módulo de transmisión PC - XBee	60
3.15 XBee transmisor	61
3.16 Reguladores de voltaje de 5v y 3.3V	62
3.17 Fuente de poder.....	62
3.18 Módulo Transmisión Recepción.....	63
3.19 Xbee Receptor.....	64
3.20 Relés controladores	64
3.21 Motor AC 110V	65
3.22 Cadena.....	66
3.23 Sensor detector capacitivo.....	67
3.24 Control de visualización al LCD	67
3.25 LCD 20x4	68
3.26 Funcionamiento LCD.....	68
3.27 Pantalla Inicio LCD.....	69
3.28 Pantalla Integrantes	69
3.29 Botoneras de control (marcha – pare).....	70
3.30 Banda transportadora	70
3.31 Máquina transportadora	71
3.32 Módulo transmisor XBEE	71
3.33 Módulo Receptor XBEE.....	72
3.34 Circuito Impreso contador LCD	72

3.35 Circuito Impreso control.....	73
3.36 Circuito Impreso Fuente de Poder.....	73
3.37 Circuito Impreso Control por medio de Relé.....	74
3.38 Circuito Impreso Transmisor - Receptor	74
3.39 Diseño del módulo de envío y recepción	75
4.1 Módulo completo de transmisión	77
4.2 Cables para interconectar los diferentes módulos para transmisión	77
4.3 Caja de protección para transmisor	78
4.4 Acoplamiento del módulo de recepción	78
4.5 Interconectando los circuitos integrados	79
4.6 Acoplamiento entre el módulo receptor con los relé	79
4.7 Luces Piloto	80
4.8 Armado Luces piloto.....	80
4.9 Montaje en la base luces piloto.....	81
4.10 Luz piloto encendida.....	81
4.11 Colocación del Led en la caja y base	82
4.12 Conexión LCD.....	82
4.13 Relé para el sensor capacitivo.....	83
4.14 Implementación del sensor capacitivo.....	83
4.15 Vista general del proyecto.....	84
4.16 Colocación de las cajas en la máquina transportadora	84
4.17 Canaletas parte superior.....	85
4.18 Canaletas parte de Abajo	85
4.19 Borneras	86
4.20 Colocación de amarras a los cables de electricidad	86
4.21 Máquina Transportadora completa.....	87
B.1 Pantalla de Login.....	121
B.2 Menú con usuario regular	121
B.3 Menú usuario Mantenimiento	122
B.4 Menú mantenimiento Datos de Usuario.....	122
B.5 Mantenimiento de Usuario.....	123
B.6 Búsqueda de empleado.....	123
B.7 Datos empleado.....	124
B.8 Mensaje Eliminación	124

B.9 Confirmación Actualización.....	125
B.10 Actualización empleado.....	125
B.11 Ingreso de Usuario.....	126
B.12 Ventana de Registro.....	126
B.13 Confirmación ingreso nuevo usuario	127
B.14 Registro guardado	127
B.15 Menú principal inicio	128
B.16 Verificación Com1.....	128
B.17 Ingresos de valores numéricos	129
B.18 Pantalla de control	129
B.19 Funcionamiento del sistema	131
B.20 Bloqueo del sistema.....	132
B.21 Estado encendido	132
B.22 Estado Error	132
B.23 Estado Apagado	133
B.24 Cerrar sistema.....	133
B.25 Menú Acerca de	134
B.26 Acerca de.....	134
B.27 Error de Clave	135
B.28 Registro no existe.....	135
B.29 Llenar campos	136
B.30 Error COM1.....	136
B.31 ingrese valor	136
B.32 Sistema Bloqueado	137

ÍNDICE DE TABLAS

1.1 Características del Pic 16F870/871	22
1.2 Características del Pic 16F628.....	23
1.3 Compuestos empleados en la construcción de leds.....	30
1.4 Descripción de los pines	32

ÍNDICE DE ANEXOS

ANEXO A: Código Fuente de sistema.....	90
ANEXO B: Manual de usuario.....	121
ANEXO C: Glosario.....	138

INTRODUCCIÓN

En un futuro cercano, el uso de la voz se convertirá en el recurso a utilizar en aplicaciones, juguetes, herramientas, computadoras, equipos médicos, industrias y robótica. Hay un gran mercado esperando para esta tecnología perfeccionada y esté disponible para las aplicaciones.

Este proyecto detalla la implementación de un módulo que utiliza la tecnología de reconocimiento de voz para poder manejar equipos electrónicos y eléctricos. Para esta tesis se atiende el área industrial en el cual un operario desde una cabina podrá transmitir órdenes a una maquinaria que él va a manipular por medio de su voz.

CONTENIDO DE LOS CAPÍTULOS

Capítulo 1: Descripción de Hardware

Menciona los componentes eléctricos y electrónicos que se van a usar para desarrollar la parte tangible del proyecto de control de voz así mismo, se describirá las características más importantes de cada uno de ellos.

Capítulo 2: Descripción de Software

Este capítulo hace referencia a todos los software usados para el diseño de las placas; describiendo al software de programación, los microcontroladores y bases de datos para que se comunique el hardware como el software.

Capítulo 3: Diseño del Sistema (Hardware – Software)

Mediante este capítulo se describe cómo es el proceso de desarrollo de la codificación de los componentes usados en el proyecto, así como las consideraciones para los diseños de las placas y el hardware restante.

Capítulo 4: Implementación del proyecto

Una vez completado el diseño del hardware y el desarrollo del sistema se unifica ambos para que se puedan interconectar y establecer comunicación hardware y software, se presentan las pruebas necesarias para que quede operativo el proyecto.

JUSTIFICACIÓN

El motivo de la realización del proyecto de control de componentes eléctricos y electrónicos por medio de comandos de voz es porque actualmente en el campo industrial hay muchos accidentes laborales, sobre todo por la exposición directa entre la persona y la máquina que crean este ambiente de peligro constante, y al estar en el mismo lugar se exponen a que en un momento desafortunado parte de su vestimenta o alguna parte de su cuerpo quede atrapado entre la maquinaria y sufra accidentes que van desde amputación del miembro hasta incluso la muerte del trabajador generando gastos a la compañía ya sea por atención de hospital o también en demandas que pueden presentar los trabajadores afectados.

SOLUCIÓN

La solución planteada trata de minimizar este tipo de accidentes ya que el operario no se encuentra directamente en el lugar donde funcionan las maquinarias sino que puede estar a muchos metros de distancia controlando las funciones de su máquina a cargo

OBJETIVOS

Objetivo General:

Establecer seguridad a los operadores de un área industrial utilizando un sistema de control de voz que se encuentre lejos del área de máquinas y así evitar accidentes laborales.

Objetivos Específicos:

- Evitar el contacto directo con una máquina eléctrica o electrónica.
- Convertir señales análogas a señales digitales de tipo audible.
- Transmitir las señales de audio vía aire por medio de dispositivos electrónicos.
- Conocer cómo se comunican estos equipos entre sí.
- Conocer el funcionamiento del módulo electrónico xbee
- Diseñar un sistema que combine el hardware con el software en un solo proyecto.

DIAGRAMA DE FUNCIONAMIENTO DEL PROYECTO

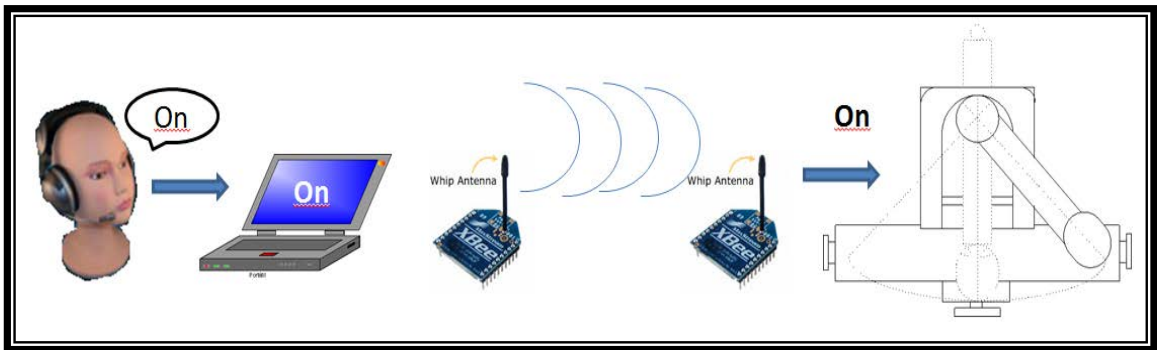


Figura A: Estructura básica del sistema

Fuente: Autores

El componente principal es el usuario el cual tendrá la posibilidad de cambiar de un estado al otro según él lo desee. El segundo componente es la computadora con la cual nosotros decodificamos la voz y la pasamos a comandos legibles para la máquina, el tercer componente es el la máquina la cual ejecutará las instrucciones que la computadora le envié por vía wireless.

Comandos de voz que se utilizaran en el sistema de control

- Ingreso de contraseña inicial
- Encendido motor
- Detener motor
- Número a contar por el sensor
- Encender y apagar video
- Manejo general del sistema (menú, ingreso, cerrar sistema)
- Bloqueo de sistema

Señales que enviará la maquinaria al sistema

- Alarma de Error
- Confirmación de Encendido de los equipos eléctricos
- Confirmación de terminación del proceso de conteo

DISEÑO DE LA MÁQUINA INDUSTRIAL

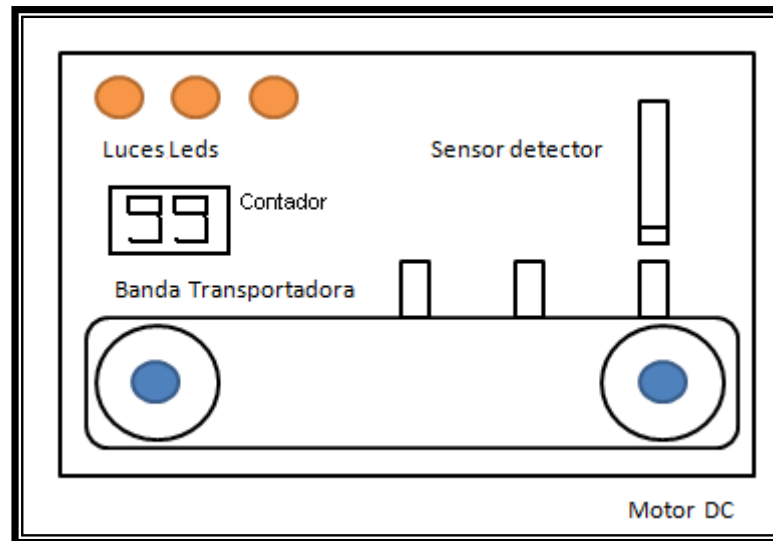


Figura B: Diseño de la Máquina transportadora

Fuente: Autores

Contará con una banda para movilizar los elementos, un sensor detector de metales, una pantalla LCD que indicará los mensajes enviados desde el computador a la máquina y viceversa; luces pilotos que indicarán los estados de las acciones enviadas: encendido del motor, apagado de los equipos y alarma de error. Un motor de corriente continua para la rotación y un contador numérico que indicará cuantos elementos pasarán por la banda transportadora.

FUNCIONAMIENTO DEL SISTEMA

Un operador desde una cabina ubicada fuera del área de máquinas puede dar instrucciones mediante un micrófono. Por medio del Dragón Natural Speaking, usaremos este software de reconocimiento de voz, que interpreta los comandos transformándolos a texto para ser grabados por el software. El texto producido es transferido al sistema que interpreta las órdenes y de ahí es enviado por medio del puerto paralelo al transmisor digital en tiempo real, transmitido vía wireless hacia el receptor que se encuentra en la habitación donde está el elemento eléctrico y pueda ejecutar la orden enviada desde la cabina del operador. Para esta propuesta se

diseñará una máquina eléctrica básica que demostrará nuestra tesis, tendrá interruptores, motores, luces indicadoras y una banda transportadora.

ALCANCE DEL PROYECTO

El sistema está diseñado para trabajar con dispositivos eléctricos y electrónicos; con modificaciones, puede ser aplicable para cualquier tipo de aparato desde el control de encendido y apagado de un foco hasta una máquina industrial de grandes proporciones.

LIMITACIONES DEL PROYECTO

El uso de dispositivos de reconocimiento de voz no está perfeccionado en su totalidad y eso implica que ninguno de los sistemas que utilicen señales auditivas para recibir órdenes decodifiquen al 100% la voz por lo que, en algunas ocasiones, se tiene que repetir la instrucción para que pueda ser interpretada por el software de control de voz, evitar usar el sistema en distancias muy extensas para evitar atenuación de la señal y el sistema solo debe de ser usados por sistemas operativos de Microsoft.

CAPÍTULO I

DESCRIPCIÓN DE HARDWARE

En este capítulo mostraremos la descripción y detalle los componentes que se utilizaron para la creación de nuestro proyecto de control de voz los cuales son microcontroladores, relé, sensor, puerto serial, cámara, luces piloto Led.

1.1 Puerto Serial



Figura 1.1 Conector serial RS-232

Fuente: Autores

Un puerto serie o puerto serial (Figura 1.1) es un puerto de comunicación de la computadora que se ha usado por más de 20 años, uno de los primeros dispositivos en usar esos puertos fueron los teclados y módems, siendo una de las primeras interfaces que permitieron compartir información externa en la computación, la transmisión es asíncrona por lo que no se necesita el control de un reloj (reloj cpu) aunque en la actualidad hay los puertos que son sincronizados, la comunicación se la realiza enviando una señal una detrás de otra (bit a bit), son bidireccionales ya que pueden enviar y recibir datos simultáneamente y tienen una atenuación de aproximadamente 15 metros. El protocolo utilizado para el envío y recepción de datos es el protocolo *Recommended Standard 232* (RS-232), también son llamados puertos COM

1.1.1 Características del puerto serial

- El puerto utilizado comúnmente es el COM1
- El estándar utilizados por el puerto serial es el RS-232

- Comunicación asíncrona

1.1.2 Tipos de comunicación en serie

Los tipos de comunicación en serie que existen son:

- Simplex: Comunicación unidireccional
- Duplex, half duplex o semi-duplex: envía y recibe datos no simultáneamente
- Full Duplex: comunicación bidireccional

1.1.3 Señales del puerto serial

- Request To Send (RTS): Indicador de transmisión
- Clear To Send (CTS): Indicador de que puede enviar datos
- Data Terminal Ready (DTR): Indicador de Recepción
- Data Set Ready (DSR): Indicador de que puede recibir datos
- Receive Signal Line Detect (RSLD) : Indicador de conexión
- Transmit Data (TD) transmite el dato
- Receive Data (RD): Recibe el dato
- Signal Ground (GND): señal a tierra.
- Carrier Detect: Detector de Transmisión

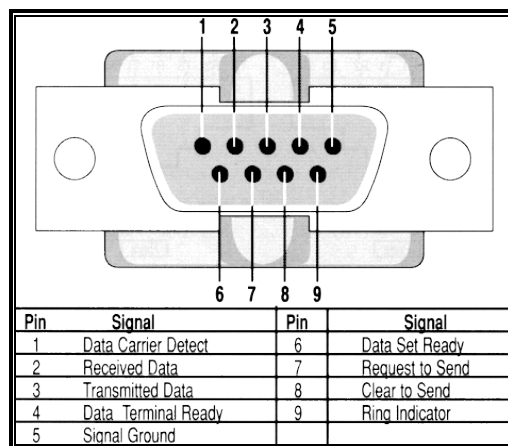


Figura 1.2 Pines RS 232

Fuente: <http://www.arcelect.com/rs232.htm>

En la figura muestra cada uno de los nombres de los pines que tiene incorporado el conector rs232

1.2 Microcontrolador PIC

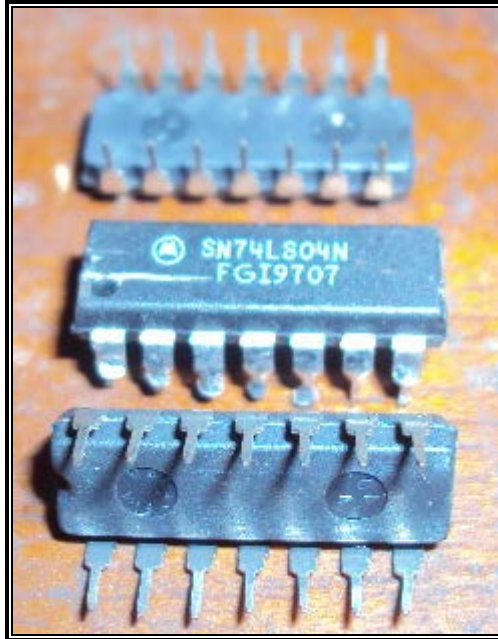


Figura 1.3 PIC

Fuente: Autores

Un pic es un microprocesador que fueron desarrollados por Microchip, el nombre completo es PICmicro pero más se lo considera como PIC (Peripheral Interface Controller).

Los microcontroladores no realizan tareas de forma automática, deben de ser programados por software de ensamblaje para que efectúen las tareas deseadas como desde encender un led hasta controlar el manejo de un robot.

Existen en la actualidad más de 180 modelos de pic

1.2.1 Pics más usados en el mercado

Se detalla la lista de los pics más usados en la actualidad

- PIC12C508/509
- PIC12F629/675
- PIC16F84
- PIC16F84A
- PIC16F628A

- PIC16F88
- PIC16F886/887
- PIC16F193x
- PIC18F2455
- PIC18F2550
- PIC18F452
- PIC18F4550
- dsPIC30F2010
- dsPIC30F3014
- dsPIC30F3011
- PIC32

1.2.2 Pics usados en el proyecto

Para el proyecto de control de dispositivos eléctricos y electrónicos por medio de comandos de voz usamos los pic 16F870 y 16F628 (Figura 1.4, 1.5)

1.2.2.1 PIC16F870/871

Características

- Memoria de programa 3584 bytes (Flash)
- Ram 128 bytes
- EEPROM 64 bytes
- 22 pinout I/O (PIC16F870) y 33 (PIC16F871)
- 22 mA por pinout
- Entradas análogas, 6 (PIC16F870), 8 (PIC16F871)

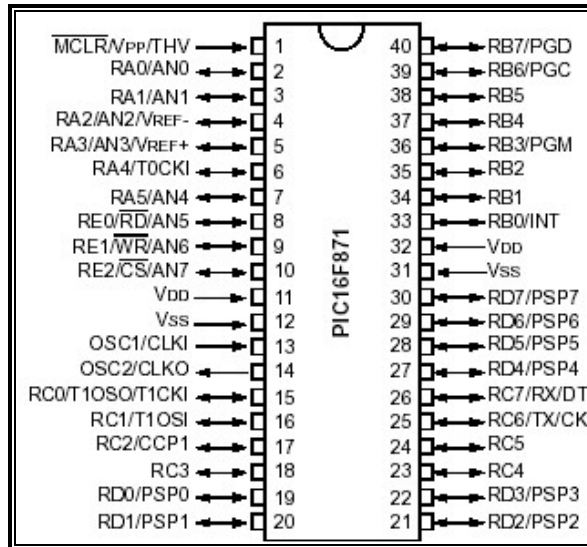


Figura 1.4 PIC 16F871

Fuente: www.microchip.com

Key Features PICmicro™ Mid-Range MCU Family Reference Manual (DS33023)	PIC16F870	PIC16F871
Operating Frequency	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	2K	2K
Data Memory (bytes)	128	128
EEPROM Data Memory	64	64
Interrupts	10	11
I/O Ports	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3
Capture/Compare/PWM modules	1	1
Serial Communications	USART	USART
Parallel Communications	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels
Instruction Set	35 Instructions	35 Instructions

Tabla 1.1 Características del PIC 16F870/871

Fuente: www.alldatasheet.com

1.2.2.2 PIC16F628

Características:

- EEPROM 128 Kbyte
- Input / Output Puertos A,B
- Rango de Voltaje 2.0 v to 5.5 v
- Four interrupt sources:

- RB0/INT pin externo
- TMR0 tiempo de sobreflujo
- PORTB<7:4> cambios sobre interrupción
- Data EEPROM de escritura completa
- Comunicación Serial (USART)

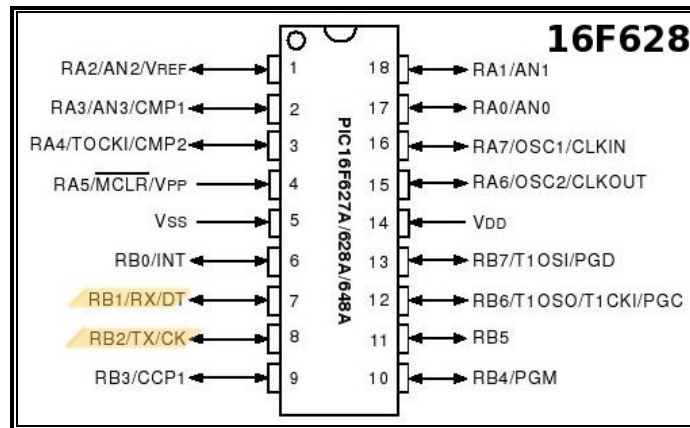


Figura 1.5 PIC 16F628

Fuente: www.microchip.com

		PIC16F627A	PIC16F628A	PIC16F648A	PIC16LF627A	PIC16LF628A	PIC16LF648A
Clock	Maximum Frequency of Operation (MHz)	20	20	20	20	20	20
	Flash Program Memory (words)	1024	2048	4096	1024	2048	4096
Memory	RAM Data Memory (bytes)	224	224	256	224	224	256
	EEPROM Data Memory (bytes)	128	128	256	128	128	256
	Timer module(s)	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
Peripherals	Comparator(s)	2	2	2	2	2	2
	Capture/Compare/PWM modules	1	1	1	1	1	1
	Serial Communications	USART	USART	USART	USART	USART	USART
	Internal Voltage Reference	Yes	Yes	Yes	Yes	Yes	Yes
Features	Interrupt Sources	10	10	10	10	10	10
	I/O Pins	16	16	16	16	16	16
	Voltage Range (Volts)	3.0-5.5	3.0-5.5	3.0-5.5	2.0-5.5	2.0-5.5	2.0-5.5
	Brown-out Reset	Yes	Yes	Yes	Yes	Yes	Yes
	Packages	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN	18-pin DIP, SOIC, 20-pin SSOP, 28-pin QFN

Tabla 1.2 Características PIC 16F628

Fuente: www.alldatasheet.com

1.3 Xbee



Figura 1.6 Módulos XBee

Fuente: Autores

Son módulos de radiofrecuencia que trabajan en una banda de 2.4 Gigahertz, utilizan 2 formas de comunicación, la del modo AT y el modo API (Application Programmers Interface), son fabricados por la empresa MAXStream. Se utilizan para las automatizaciones de hogar, alarmas, instrumental médicos, demótica, etc. Están basados en el protocolo estándar 802.15.4-2003 con tasa de transferencia de 250 kbps (Kilobits por segundo), tienen incorporadas antenas para la transmisión inalámbrica, pueden ser programados a través de una hyperterminal o una interfase serial con un max3232.

Funcionan con los tipos de arquitecturas los cuales son punto a punto, multipunto o en una red MESH y su fuente de alimentación mínima para funcionamiento es de 3.3v.

1.3.1 Características de los xbee.

- Alcance de 100 mts en módulos XBEE y 1.6 Km para los módulos Xbee Pro.
- 9 entradas/salidas analógicas y digitales.
- Son de bajo consumo eléctrico, menor a 50 mili Amperios

- Interfaz serial.
- 65,000 direcciones para cada uno de los 16 canales disponibles
- Fáciles de integrar.
- económicos, poderosos y fáciles de utilizar

1.3.2 Tipos de Xbee

Hasta el momento han aparecido en el mercado estos modelos de xbee:

- XBee 802.15.4
- XBee-PRO 802.15.4
- XBee ZB
- XBee-PRO ZB
- XBee ZB SMT- A
- XBee-PRO ZB SMT
- XBee SE- An XBee ZB
- XBee PRO SE
- XBee PRO 900
- XBee PRO 868
- XBee PRO XSC
- XBee-PRO DigiMesh 900
- XBee DigiMesh 2.4
- XBee-PRO DigiMesh 2.4

1.4 Sensor Capacitivo



Figura 1.7 Sensor Capacitivo

Fuente: Autores

Los sensores capacitivos (Figura 1.7) son un tipo de sensor eléctrico, reaccionan ante metales y no metales que al aproximarse a la superficie activa sobrepasan una determinada capacidad. La distancia de conexión respecto a un determinado material es tanto mayor cuanto más elevada sea su constante dieléctrica.

Mediante un potenciómetro se puede regular la distancia de detección.

1.4.1 Ventajas y desventajas

A continuación se detallan las ventajas y desventajas de los sensores capacitivos

1.4.1.1 Ventajas:

- Gran variedad de detección de materiales
- Responden más rápidamente que los sensores inductivos
- Confiables porque poseen mayor precisión
- No se afectan con la suciedad y polvo
- Larga vida útil

1.4.1.2 Desventajas:

- Relativamente caros comparados con los sensores inductivos
- Corta distancia de detección

Pueden ser aplicados en la industria farmacéutica, alimentaria, plásticos, alimentos de animales, detectores y control de nivel, control de flujo de líquidos, etc. Son de gran utilización tanto para aplicaciones de posicionamiento como para detectar la presencia de objetos en un determinado contexto.

1.5 Relé



Figura 1.8 Relé

Fuente: Autores

Un Relé es un dispositivo electromecánico que funciona como un interruptor, consta de dos circuitos diferentes: un electroimán y un circuito de contactos con el cual se puede controlar una potencia mucho mayor con un consumo en potencia muy reducido por lo que es considerado como un amplificador eléctrico, soporta tanto corriente alterna como corriente continua, los materiales son fabricados con plata o aleaciones de plata dependiendo de la vida útil que necesite el relé. Fue inventado por Joseph Henry en 1835.

1.5.1 Funcionamiento del relé

Para que funcione un relé debe de pasar corriente en su bobina lo que genera un campo magnético y eso hace que atraiga el contacto que se encontraba en estado apagado o reposo y lo ponga en estado activo y viceversa de forma independiente el uno del otro, cuando hace el cambio de estado produce el sonido característico de los relé (clic).

1.5.2 Tipos de relé

Existen varios tipos de relé, los más conocidos son:

- Relés electromecánicos
- Relé de estado sólido
- Relé de corriente alterna
- Relé de láminas
- Relés híbridos

Entre los Relé electromecánicos tenemos los de tipo armadura, de Núcleo Móvil, Reed o de Lengüeta, Polarizados.

1.5.3 Características de un Relé

Entre las principales características tenemos:

- Aísla los componentes de entrada y salida
- Soporta sobrecarga
- Larga vida útil
- Resistencia a golpes y caídas

1.5.4 Características técnicas

1.5.4.1 Electromagnética

- Potencia de consumo de la bobina.

- Tensión nominal para activar el Relé
- Márgen de tensión mínima y máxima
- Intensidad de necesaria para activar el relé.

1.5.4.2 Contactos

- Tensión de conexión entre contactos al abrir y cerrar
- Intensidad máxima de conexión
- Intensidad máxima de trabajo

1.6 LUCES LED



Figura 1.9 Luces LED

Fuente: Autores

Un led (Diodo Emisor de Luz) es un tipo de diodo especial que al momento de pasar corriente por el dispositivo emite una luz radiante y dependiendo de la cantidad de corriente que es enviada cambia la cantidad de intensidad de iluminación, su comportamiento es igual al de un diodo de silicio o germanio. Pueden emitir varios tipos de colores como el azul, amarillo, verde, rojo e inclusive luz ultravioleta e infrarroja dependiendo de la variación de la carga eléctrica que oscila entre 1.3 y 4 voltios.

1.6.1 Componentes del LED

Los componentes de un diodo led son:

- Ánodo
- Cátodo
- Lente
- Contacto metálico
- Cavidad reflectora
- Terminación del semiconductor
- Plaquetas
- Borde plano

Compuesto	Color	Long. de onda
Arseniuro de galio (GaAs)	Infrarrojo	940 nm
Arseniuro de galio y aluminio (AlGaAs)	Rojo e infrarrojo	890 nm
Arseniuro fosfuro de galio (GaAsP)	Rojo, anaranjado y amarillo	630 nm
Fosfuro de galio (GaP)	Verde	555 nm
Nitruro de galio (GaN)	Verde	525 nm
Seleniuro de zinc (ZnSe)	Azul	
Nitruro de galio e indio (InGaN)	Azul	450 nm
Carburo de silicio (SiC)	Azul	480 nm
Diamante (C)	Ultravioleta	
Silicio (Si)	En desarrollo	

Tabla 1.3 Compuestos empleados en la construcción de leds

Fuente: www.wikipedia.org

1.7 Pantalla de Cristal Líquido (LCD)



Figura 1.10 Pantalla LCD

Fuente: Autores

Una pantalla LCD (Display de cristal líquido), se utilizan para desplegar mensajes de los caracteres alfanuméricos ASCII, consumen menos corriente que un display de 7 segmentos, existen de varios tipos como los de 2x16, 2x20, 4x20, 4x40, etc, operan entre 4 y 8 bits.

La pantalla usada en el proyecto es de 20 x 4, es decir, 4 filas con 20 líneas de caracteres por fila y cuenta con el chip HD44780 de interfaz paralela de 8bits. Las letras son negras sobre fondo verde. Incluye retroiluminación, solo puede mostrar texto, por lo que son usados para mostrar información sencilla en impresoras, fax, copiadoras, etc,

Para usar la pantalla lcd se envían los datos de la siguiente manera:

- 1.- Poner RS en 0 para escribir un comando o 1 si se va a escribir datos.
- 2.- poner el bus de datos el comando o dato que va a enviarse.
- 3.- Enviar el pulso por el pin 6 (E).
- 4.- Esperar a que llegue la instrucción al lcd.

1.7.1 Características de los pines del lcd

- Masa
- VCC (3.3 o 5V)
- Ajuste de contraste (VO)

- RS (RS=0 comando, RS=1 datos)
- RW (RW=0 escribir, RW=1 leer)
- Disponible
- Bit[0..7] (bus de datos)
- Ánodo del backlight (+)
- Cátodo del backlight (-)

Interface pin description			
Pin no.	Symbol	External connection	Function
1	V _{SS}	Power supply	Signal ground for LCM (GND)
2	V _{DD}		Power supply for logic (+5V) for LCM
3	V ₀		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power Supply	Power supply for BKL (Anode)
16	LED-		Power supply for BKL (GND)

Tabla 1.4 Descripción de los pines

Fuente: www.alldatasheet.com

Los caracteres ASCII soportados por el LCD son los siguientes:

	0x	2x	3x	4x	5x	6x	7x	Ax	Bx	Cx	Dx	Ex	Fx
x0		0	1	2	3	4	5	6	7	8	9	A	B
x1		!	@	Q	a	q	.	7	4	ä	q		
x2		"	2	R	b	r	「	イ	ツ	ズ	β	θ	
x3		#	3	C	s	c	」	ウ	テ	ε	ε	ω	
x4		\$	4	O	T	d	t	、	エ	ト	ト	μ	Ω
x5		%	5	E	U	e	u	・	オ	ナ	1	ε	Ü
x6		&	6	F	U	f	U	ヲ	カ	ニ	ヨ	ρ	Σ
x7		'	7	G	W	g	w	ヲ	キ	ヲ	ラ	Q	π
x8		(8	H	X	h	x	イ	ク	ネ	リ	J	又
x9)	9	I	Y	i	y	オ	ケ	ル	レ	U	
xA		*	:	J	Z	j	z	エ	コ	ハ	レ	i	〒
xB		+	;	K	L	k	l	オ	サ	ヒ	ロ	*	〒
xC		,	<	L	¥	l	l	ヤ	シ	フ	ワ	Φ	〒
xD		-	=	M	J	m	j	ユ	ズ	ヘ	ン	モ	÷
xE		.	>	N	^	n	^	ヨ	セ	ホ	°	ん	
xF		/	?	0	_	o	←	ッ	ソ	マ	°	○	■

Figura 1.11 Caracteres soportados por el LCD

Fuente: <http://gzaloprgm.com.ar/lcdalfanumerico>

1.8 Cámara Web



Figura 1.12 Cámara Web

Fuente: Autores

Una cámara web (Webcam) es una cámara digital que se conecta al computador por medio del cable serial universal (USB), permite capturar imágenes y fotos en tiempo real y transmitir las al computador, son utilizadas para el hogar aunque en la actualidad hay las que son utilizadas para servicios de vigilancia como son las cámaras web de vigilancia inalámbricas y las de última tecnología poseen resoluciones de alta definición (high definition), pueden ser fijas o con movimiento de 360 grados y de diferentes tamaños.

Son muy utilizadas para las videoconferencias en hogares, empresas y en chats de Internet como el msn, yahoo, skype, todo esto gracias al software que permite que las cámaras web capturen fotos y videos para su transmisión a Internet, también pueden transmitir audio.

Se clasifica como de un periférico de entrada, ya que por medio de la cámara se envía imágenes hacia la computadora

Fue creada en 1991 por Quentin Stafford-Fraser y Paul Jardetzky, el motivo fue porque en el departamento donde trabajan había el lema del que “terminaba el café en la cafetera tenía que llenarlo” y ellos para evitar llenarla crearon un protocolo

cliente servidor y lo colocaron en una cámara que tenía una resolución de 128 x 128 la cual fue colocada en su computadora y comenzaron a transmitir video desde la cafetería hasta su computador, este protocolo lo denominaron XCoffee, luego salió al mercado con el nombre de XCam.

1.8.1 Componentes de la cámara web

- Lente
- Sensor de imagen
- Circuitería necesaria

La cámara Web más básica trabaja a resolución VGA (Video Graphics Array) con resoluciones de 640 x 480 pixeles y las cámaras más profesionales trabajan con resoluciones HD (1024 x 960)

También la cámara web ha sido utilizado para fines maliciosos como es el de transmisión de contenido erótico en las páginas de internet y por su diminuto tamaño (pueden ser del tamaño de una moneda de 1 centavo) son utilizadas como medios de espionajes.

Aunque las cámaras web necesitan de un computador para transmitir videos, hay otras que no necesitan el computador, solamente hacen uso de una conexión a la red informática por medio de tarjetas de red, y se las conoce como cámaras web inalámbricas y pueden ser vistas por varios computadores a la vez por medio del Internet.

CAPÍTULO II

DESCRIPCIÓN SOFTWARE

En este capítulo se indicará los programas usados en el desarrollo del proyecto de control de voz. En lo que se refiere al software usamos el lenguaje de programación Visual Basic 6.0 para el diseño del programa y para la Base de datos se emplea MYSQL. Para el módulo de voz se usó el programa Dragon Naturall Speaking y por último la parte electrónica se usaron los programas pikit y ares que sirven para programación del pic y diseño de placas, a continuación detallamos las descripciones de cada uno de los software.

2.1 Visual Basic

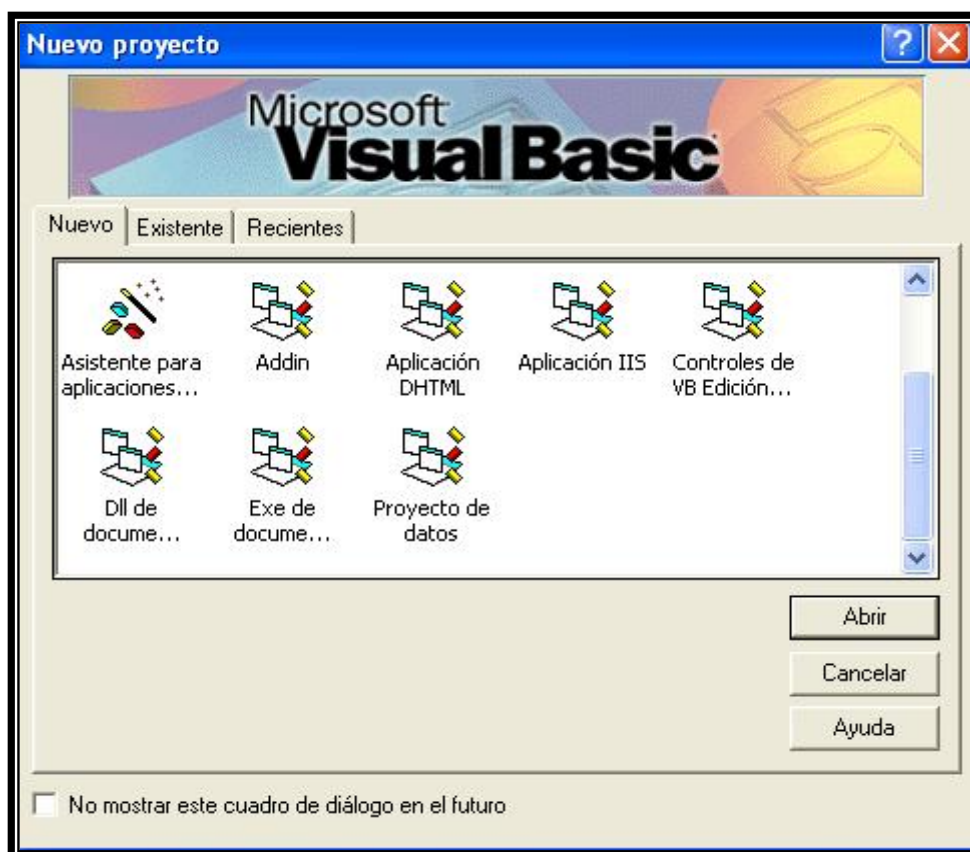


Figura 2.1 Microsoft Visual Basic

Fuente: Programa Microsoft Visual Basic 6.0 tomado por los Autores

Visual Basic (Figura 2.1) Visual Basic 6.0 es un lenguaje de programación visual, llamado lenguaje de 4ª generación. Esto indica que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el mouse sobre la pantalla.

Visual Basic 6.0 es un programa basado en objetos, aunque no orientado a objetos como C++ o Java. La diferencia está en que Visual Basic 6.0 utiliza objetos con propiedades y métodos, pero no tiene los mecanismos de herencia y polimorfismo propios de los verdaderos lenguajes orientados a objetos como Java y C++.

2.1.1 Historia

Hace 10 años y un poco más, el construir una simple aplicación basada en Microsoft Windows se habría podido describir como complicado y difícil. Construir estas aplicaciones ricas en gráficos una tarea que hoy parece sencilla no era un proceso trivial antes de la introducción de Visual Basic1.0 en mayo de 1991.

Con Visual Basic, los programadores podían, por primera vez, implementar aplicaciones en Windows en un ambiente intuitivo y gráfico, simplemente arrastrando controles sobre un formulario. Haciendo posible a los programadores profesionales y a los ocasionales ampliar su productividad, Visual Basic oriento a un renacimiento del desarrollo de aplicaciones basadas en Windows.

En estos últimos 10 años, la comunidad de Visual Basic ha crecido hasta ser la mayor comunidad de desarrolladores de software del mundo. Durante este tiempo, una industria entera de vendedores de componentes creció alrededor de este producto. Combinado con la sencilla forma de desarrollar aplicaciones para Windows.

La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte de este lenguaje hasta marzo de 2008.

Tanto esta como la posterior 6.0 soportaban características propias de los lenguajes orientados a objetos, aunque careciendo de algunos ítems importantes como la herencia y la sobrecarga. La versión 6.0 continúa utilizándose masivamente y es compatible prácticamente al 100% con las últimas versiones de Windows como Vista y Windows 7.

Las versiones actuales de Visual Basic se basan en la plataforma .NET, que se desligan de las anteriores versiones.

2.1.2 Versiones:

- Visual Basic 1.0 para Windows en Mayo de 1991.
- Visual Basic 1.0 para DOS (Sistema operativo de disco) en Septiembre de 1992.
- Visual Basic 2.0 en Noviembre de 1992.
- Visual Basic 3.0 se liberó en verano de 1993, en versiones Standard y Profesional.
- Visual Basic 4.0, de Agosto de 1995 fue la primera versión que generaba aplicaciones de 16 y 32 bits para Windows.
- Con la versión 5.0, de Febrero de 1997, Microsoft lanzó Visual Basic exclusivamente para generar programas de 32 bits.
- Visual Basic 6, salido a mediados de 1998, muy mejorado
- Visual Basic 2010 o 10, salido en 2010.

2.1.3 Características

Visual Basic 6.0 soporta la abstracción, la encapsulación, el poliformismo y la reutilización del código. Los objetos de Visual Basic tienen propiedades, métodos y eventos. Las propiedades son los datos que describen un objeto. Los eventos son hechos que pueden ocurrir sobre un objeto. Un método agrupa el código que se ejecuta en respuesta a un evento.

Al conjunto de propiedades y métodos se le llama interfaz. Además de su interfaz predeterminada, los objetos pueden implementar interfaces adicionales para

proporcionar polimorfismo. El polimorfismo le permite manipular muchos tipos diferentes de objetos sin preocuparse de su tipo.

Las interfaces múltiples son una característica del modelo de objetos componente (COM) y permiten que los programas evolucionen con el tiempo, agregando nueva funcionalidad sin afectar al código existente.

La reutilización del código es la capacidad de trasladar características de un objeto a otro, existen principalmente dos formas de reutilización del código: binario y fuente. En una jerarquía de clases, la herencia muestra cómo los objetos que se derivan de otros objetos más simples heredan su comportamiento.

2.1.4 Entorno de desarrollo

Una vez que haya instalado Visual Basic, para iniciarlo proceda igual como cualquier otra aplicación Windows: haciendo doble clic sobre el icono correspondiente o haciendo clic en el botón Inicio de la barra de tareas, eligiendo Programas y después Visual Basic.

Cuando usted arranca Visual Basic usted ve una interfaz similar a la de la figura siguiente:

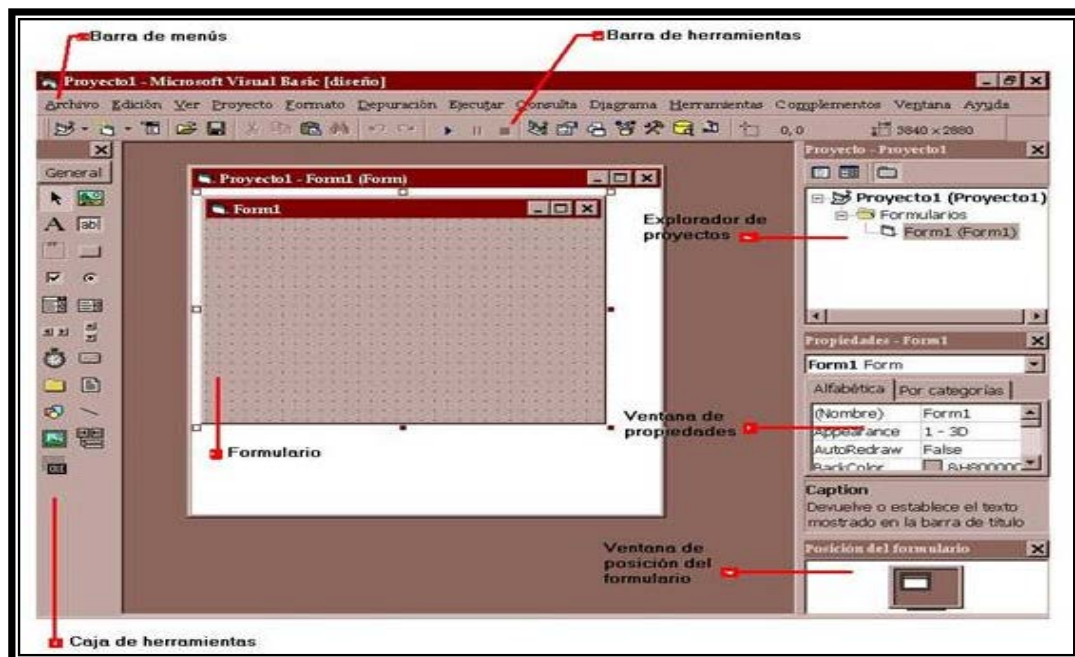


Figura 2.2 Opciones Microsoft Visual Basic

Fuente: <http://andresorellana.tripod.com/tema2.htm>

Se compone principalmente de barra de herramientas y menús, explorador de proyectos, formularios, ventana de propiedades, caja de herramientas, ventanas de posición del formulario, que se pueden personalizar con prácticamente la totalidad de los comandos del IDE a necesidad.

Usted puede interactuar con sus herramientas, para así tener facilidad en realizar el sistema de acuerdo a su necesidad.

2.1.4.1 Ejemplo de código

El siguiente fragmento de código muestra un cuadro de mensaje, en una ventana, que dice "Hola mundo!":

```
Private Sub Form_Load()  
    ' Ejecuta un simple box de mensaje que dice "Hola mundo!"  
    MsgBox "Hola mundo!"  
End Sub
```

2.1.5 Ventajas

- Posee una curva de aprendizaje muy rápida.
- Integra el diseño e implementación de formularios de Windows.
- Permite usar con facilidad la plataforma de los sistemas Windows.
- Es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos.
- Fácilmente extensible mediante librerías DLL y componentes ActiveX de otros lenguajes de programación.
- Posibilita añadir soporte para ejecución de scripts, VBScript o JScript, en las aplicaciones mediante Microsoft Script Control.
- Tiene acceso a la interfaz de programación de aplicaciones multimedia de DirectX (versiones 7 y 8). También está disponible, de forma no oficial, un componente para trabajar con OpenGL 1.1.[7](#)
- Existe una versión integrada en las aplicaciones de Microsoft Office, tanto Windows como Mac, que permite programar macros para extender y

automatizar funcionalidades en documentos, hojas de cálculo, bases de datos (Access).

- Si bien permite desarrollar grandes y complejas aplicaciones, también provee un entorno adecuado para realizar pequeños prototipos rápidos.

2.1.6 Inconvenientes

Las críticas hechas en las ediciones de Visual Basic anteriores a VB.NET son variadas, se citan entre ellas:

- Problema de versión asociado con varias librerías runtime DLL, conocido como DLL Hell.
- Pobre soporte para programación orientada a objetos
- Incapacidad para crear aplicaciones multihilo, sin tener que recurrir a llamadas de la API de Windows.
- Dependencia de complejas y frágiles entradas de registro COM10.

2.2 Reconocimiento del habla

El Reconocimiento Automático del Habla (RAH) o Reconocimiento Automático de voz es una parte de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas.

Un sistema de reconocimiento de voz es una herramienta computacional capaz de procesar la señal de voz emitida por el ser humano y reconocer la información contenida en la misma convirtiéndola en texto y emitiendo órdenes que actúan sobre un proceso.

En su desarrollo intervienen diversas disciplinas, tales como: la fisiología, la acústica, el procesamiento de señales, la inteligencia artificial y la ciencia de la computación.

2.2.1 Decodificador Acústico Fonético

Con los correspondientes procedimientos interpretativos, dan lugar a un módulo conocido como Decodificador acústico-fonético, a la entrada del decodificador acústico-fonético es la señal vocal convenientemente representada, para la misma, es necesario que ésta sufra un pre proceso de parametrización.

En esta etapa previa es necesario asumir algún modelo físico, contándose con modelos auditivos y modelos articulatorios, pudiendo ser considerados su utilización en equipos electrónicos.

2.2.2 Modelo del Lenguaje

Las fuentes de conocimiento dan lugar al modelo del lenguaje. Cuando la representación de la Semántica y sintaxis tiende a integrarse desarrollan sistemas de gramática restringida para tareas concretas.

2.2.3 Usos y aplicaciones

Aunque en teoría cualquier tarea en la que se interactúe con un ordenador puede utilizar el reconocimiento de voz, actualmente las siguientes aplicaciones son las más comunes:

- **Dictado automático:** El dictado automático es el uso más común de las tecnologías de reconocimiento de voz. En algunos casos, como en el dictado de recetas médicas y diagnósticos o el dictado de textos legales, se usan corpus especiales para incrementar la precisión del sistema.
- **Telefonía:** Algunos sistemas PBX permiten a los usuarios ejecutar comandos mediante el habla, en lugar de pulsar tonos. En muchos casos se pide al usuario que diga un número para navegar en un menú.
- **Sistemas portátiles:** Los sistemas portátiles de pequeño tamaño, como los relojes o los teléfonos móviles, tienen unas restricciones muy concretas de tamaño y forma, así que el habla es una solución natural para introducir datos en estos dispositivos.

2.2.4 Sistemas diseñados para discapacitados: Los sistemas de reconocimiento de voz pueden ser útiles para personas con discapacidades que les impidan teclear con fluidez, así como para personas con problemas auditivos, que pueden usarlos para obtener texto escrito a partir del habla. Esto permitiría, por ejemplo, que los aquejados de sordera pudieran recibir llamadas telefónicas.

2.3 Programa de Reconocimiento de Voz Dragon Naturally Speaking



Figura 2.3 Software de control de Voz

Fuente: Programa inicio Dragon Natural Speaking 10.0 tomado por los Autores



Figura 2.4 Barra Herramientas de Dragon Natural Speaking

Fuente: Programa Dragon natural Speaking 10.0. tomado por los Autores

Es el resultado de un proceso de investigación llevado a cabo por diferentes organismos relacionados con las soluciones a los problemas de los discapacitados, con la colaboración de todos han llevado a cabo esta iniciativa con el objetivo de facilitar el acceso a la informática a las personas con algún tipo de discapacidad, sobre todo con problemas de movilidad.

Muchas son las ventajas principales de este software. Se adapta y personaliza a cualquier usuario cuya voz cumpla unos requisitos mínimos de claridad, de forma que se crea una configuración especial con la que logra una tasa de reconocimiento de voz de hasta el 98 %.

Para el funcionamiento se necesita un micrófono de buena calidad. Una vez conectado el micrófono, el usuario tiene que pasar un entrenamiento en el que se configura su tipo de voz, su volumen, y calidad de sonido ambiente, y se registra al usuario para utilizar el programa, al que pueden acceder varias personas a la vez. Entonces, hay que dar instrucciones al ordenador y éste escribe lo que le dictamos.

2.4 Lenguaje SQL

Es un lenguaje especializado para la actualización, eliminación y solicitar información de bases de datos, SQL es un estándar ISO y ANSI y es el lenguaje estándar de consulta de bases de datos. Una variedad de productos de base de datos establecida soporte para SQL, incluidos los productos de Oracle y Microsoft SQL Server, es ampliamente utilizada en la industria y la academia.

En un sistema de base de datos distribuida, se ejecuta constantemente en un servidor, la interpretación de los archivos de datos en el servidor como una base de datos relacional estándar. Programas en los equipos cliente permiten a los usuarios manipular los datos, el uso de tablas, columnas, filas y los campos.

Para ello, los programas cliente envía comandos SQL al servidor. El servidor procesa estas declaraciones y respuestas, devolviendo al programa cliente.

2.4.1 Sistemas de gestión de base de datos

Los sistemas de gestión de base de datos con soporte SQL más utilizados son:

- DB2
- Firebird
- Informix
- Interbase
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- PervasiveSQL
- SQLite
- SQL Server
- Sybase ASE

2.5 MySQL



Figura 2.5 MySQL

Fuente: Programa instalación MySQL server 5.0 tomada por los Autores

MySQL es un sistema de base de datos utilizada en la web. Básicamente una base de datos. MySQL le permite crear una estructura de base de datos relacional en un servidor web en alguna parte con el fin de almacenar los datos o automatizar los procedimientos.

MySQL también es de código abierto en el que es gratuito y está bajo la licencia General Public License (GPL), por lo general son asociados con Unix / Linux en servidores basados. Interactuar con una base de datos MySQL que es algo tan sencillo como Microsoft Access ofrece. Cuando se crean tablas, ya sea que usted tiene que crear mediante el uso de sentencias SQL, o mediante el uso de otra herramienta de código abierto disponibles en línea llamado PHP u otros lenguaje de programación etc.

MYSQL le permite realizar consultas sobre la información contenida en su base de datos la selección de datos, inserción, actualización, localización y eliminación.

2.5.1 Especificaciones del código fuente

MySQL está escrito en una mezcla de C y C++.

2.5.2 Características

Las más conocidas:

- Compatibilidad entre plataformas
- Los procedimientos almacenados
- Desencadenantes
- Actualizable Vistas
- Consulta de almacenamiento en caché
- Sub- SELECT
- Texto completo de indexación y búsqueda usando MyISAM motor
- Base de datos integrada de la biblioteca
- Parcial Unicode apoyo (UTF-8 y UCS-2 cadenas codificadas se limitan a la BMP)

- Las transacciones con la InnoDB, BDB y los motores de almacenamiento en clúster, los puntos de salvaguarda con InnoDB
- SSL apoyo

2.6 MicroCode Studio Plus

MicroCode Studio, es una herramienta de generación de código libre que permite a los usuarios implementar rápidamente una comunicación bidireccional entre un PIC™ integrado a un Microcontrolador y a un PC.

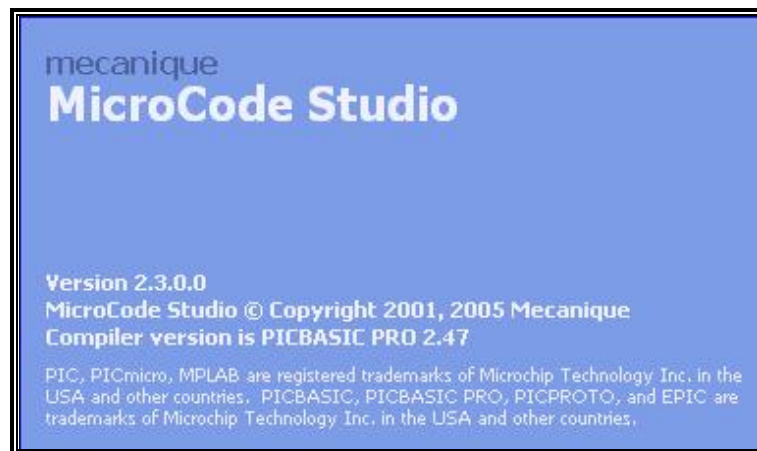


Figura 2.6 Microcode Studio

Fuente: Programa Microcode estudio 2.3 tomado por los Autores

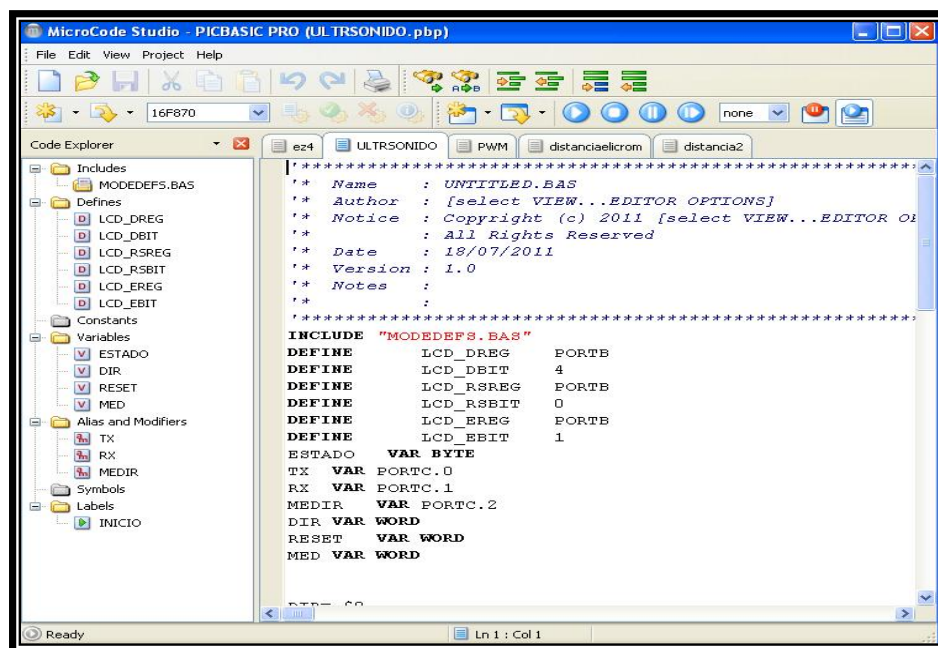


Figura 2.7 Pantalla Principal Microcode Studio

Fuente: Programa Microcode Studio 2.3 tomado por los Autores

- Destacando la sintaxis completa de su código fuente en su totalidad.
- Saltar rápidamente para incluir los archivos, los símbolos, define las variables y las etiquetas usando la ventana del explorador de código.
- Identificar y corregir errores de compilación y el ensamblador.
- Ver la salida de serie de su Microcontrolador.
- Palabra clave ayuda sensible al contexto basada.

El MicroCode queda Enlazado con el PICBASIC y el IC-PROG de manera que una vez que termina el programa compila y genera el archivo *.HEX, los programas los guarda en formato Picbase *.BASE.

2.7 Proteus

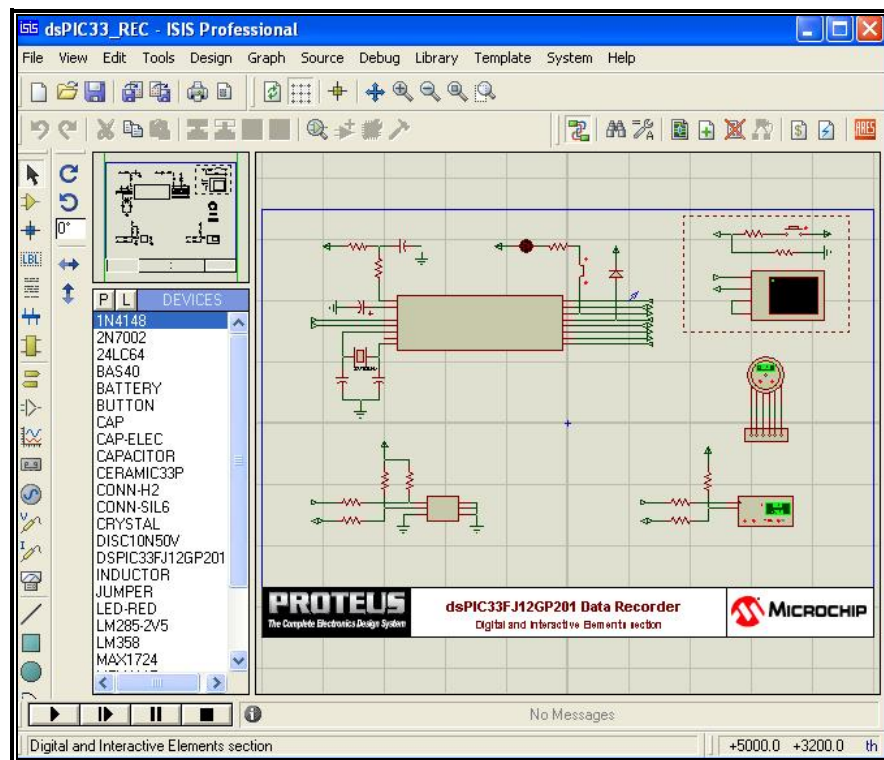


Figura 2.8 Proteus

Fuente: Programa Proteus 7 Profesional tomado por los Autores

Es una herramienta software que permite la simulación de circuitos electrónicos con microcontroladores PIC, Proteus es un simulador completo, y además es el único que permite simular microcontroladores.

El software de diseño y simulación Proteus VSM es una herramienta útil para estudiantes y profesionales que deseen mejorar habilidades para el desarrollo de aplicaciones analógicas y digitales.

El programa Proteus es una aplicación CAD (aplicación de diseño) que se compone de los módulos:

2.7.1 ISIS (Intelligent Schematic Input System) Es el módulo de captura de esquemas, permite el diseño de circuitos empleando un entorno gráfico, en el cual es posible colocar los símbolos de los componentes y realizar simulaciones de su funcionamiento, sin el riesgo de ocasionar daños a los circuitos.

2.7.2 ARES (Advanced Routing Modelling) VSM tiene la capacidad de pasar el diseño a un programa integrado llamado ARES en el cual se puede llevar a cabo el desarrollo de placas de circuitos impresos.

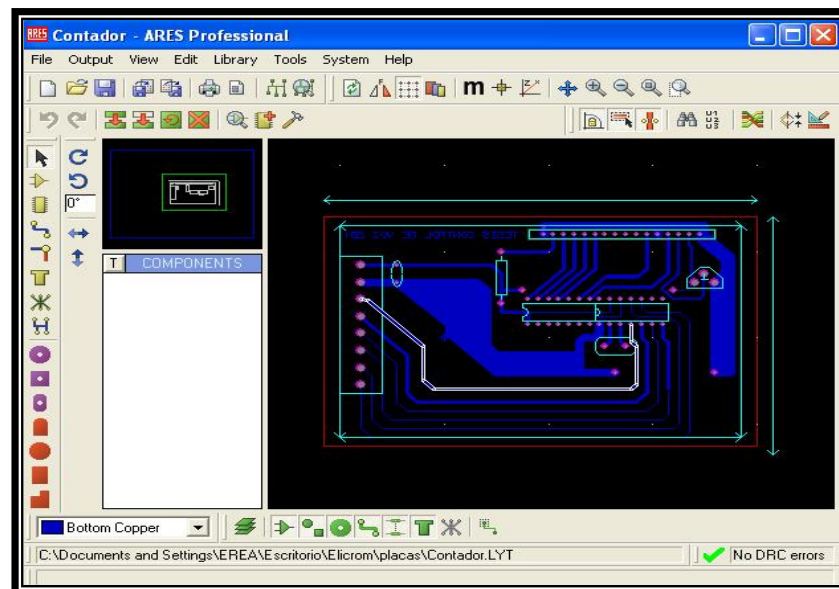


Figura 2.9 Ares

Fuente: Programa Proteus – Ares tomado por los Autores

2.8 Pickit 2 Plus

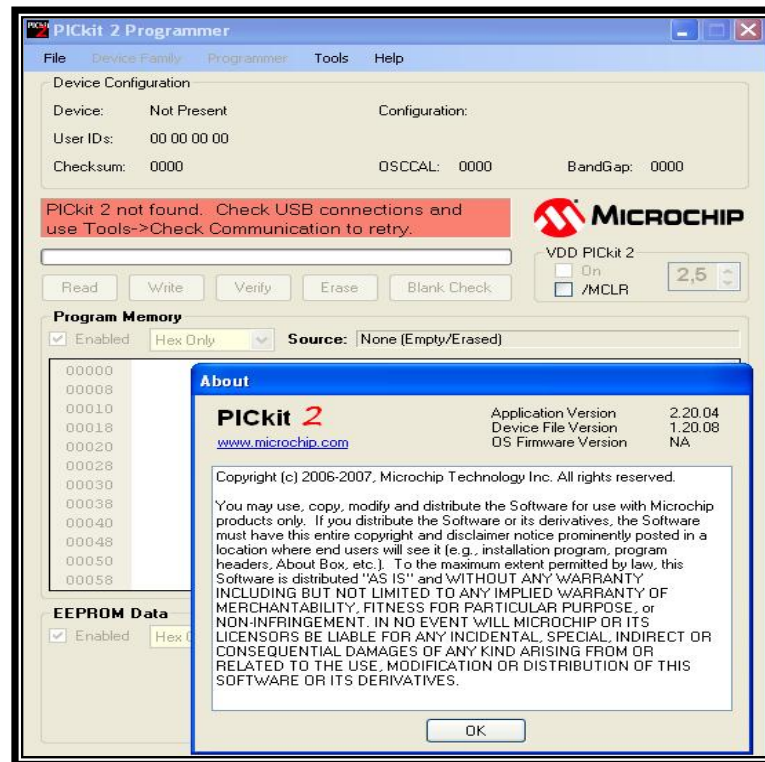


Figura 2.10 Pickit 2

Fuente: Programa Pickit2 tomado por los Autores

El PICKIT 2 es un programador de PIC. PIC es una línea de microcontroladores hecha por Microchip. El PICKIT2 es el programador de segundo más básico que se vende, es muy poco lo que no puede hacer se adapta a casi todos los esquemas electrónicos.

Es una potente herramienta para la programación de microcontroladores PIC de MICROCHIP, además soporta dispositivos vía ICSP que lo convierte en una herramienta de gran utilidad a un bajo costo (Figura 2.8).

Este programador es totalmente compatible con Mplab de Microchip el cual permite programación y debuggear programas en vivo.

2.8.1 Características generales:

- Debugger Express compatible con MPLAB.
- Analizador lógico de 2 canales.
- Puerto serial virtual UART TX - RX.
- Interfaz USB 2.0 compatible con 1.0 y 1.1.
- No requiere alimentación externa.
- Zócalo para programación tipo encapsulado DIP de dispositivos.
- Conector para programación en circuito vía ICSP.
- Interfaz UART – analizador lógico con puntas de prueba y conector independiente.

CAPÍTULO III

DISEÑO DEL SISTEMA

3.1 Diagrama de flujo del funcionamiento del sistema

Flujograma del sistema en el cual cuando arranca el proceso me indica que ingrese el usuario y contraseña, luego pasa a la verificación de que el sistema este encendido, al pasar ese proceso el sistema recibe las ordenes de los comandos de voz para que realice los procesos deseados pero antes de eso se verifica que no haya ninguna clase de error en el sistema para que continúen procesando los comandos de voz (Figura 3.1).

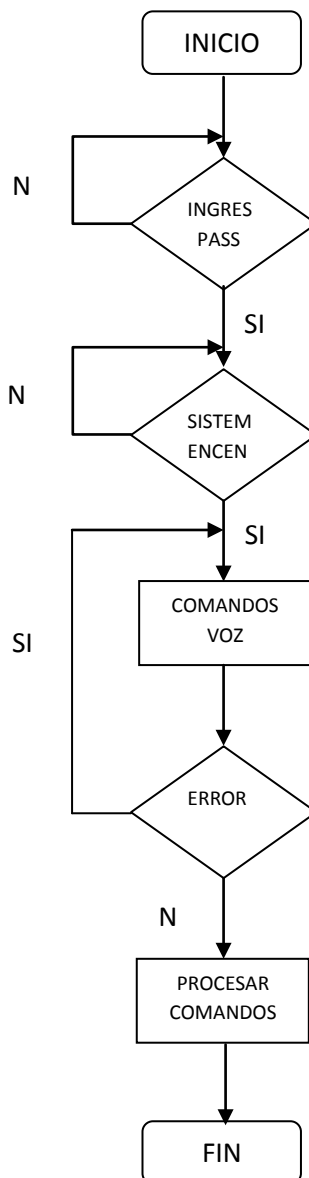


Figura 3.1 Diagrama de Flujo principal

Fuente: Autores

3.2 Diseño del Software

A continuación presentamos el diseño de la interfaz del sistema propuesto para interactuar con el usuario. Se define la forma de realizar las consultas y la información mostrada, resultados del sistema.

El sistema de reconocimiento de voz interpreta las órdenes comparándolas con los nombres de cada uno de los botones de comandos que fueron programados en visual basic, y cuando detecta que se ha dictado el nombre del botón deseado presiona el botón para que realice la selección deseada.

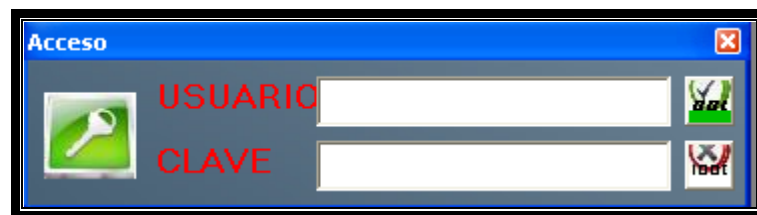


Figura 3.2 Pantalla de usuario y contraseña

Fuente: Autores

Pantalla inicial donde el operador por medio de la voz dice su usuario y contraseña (Figura 3.2), posee un usuario administrador en el cual habilita las funciones de **Mantenimiento** por medio de un menú para poder realizar modificaciones de los usuarios registrados, debe de tener funcionando el micrófono para que pueda ser leído y reconocido por el sistema.



Figura 3.3 Pantalla principal

Fuente: Autores

Pantalla del sistema el cual nos muestra el menú principal (Figura 3.3)donde se encuentran las opciones de inicio la cual contiene las funciones del sistema, el menú mantenimiento solo puede acceder el usuario que ha ingresado como el administrador del sistema, **acerca de** nos muestra la versión del sistema y los desarrolladores del mismo.

3.2.1 Mantenimiento del sistema



Figura 3.4 Pantalla de menú

Fuente: Autores

Ingresando con el usuario Administrador (Figura 3.4) tendremos acceso a la opción Mantenimiento, a la cual accedemos leyendo las opciones escritas en el sistema las cuales son Datos de usuario y la opción de Ingreso de Usuario.

En la opción datos de usuario podremos buscar, editar, y eliminar un usuario existente. La búsqueda se realiza por medio del apellido de la persona y de ahí se puede eliminarla o actualizar su información personal o cambiar la clave del usuario, ahí se puede usar la voz como el teclado, dependiendo de la comodidad del usuario

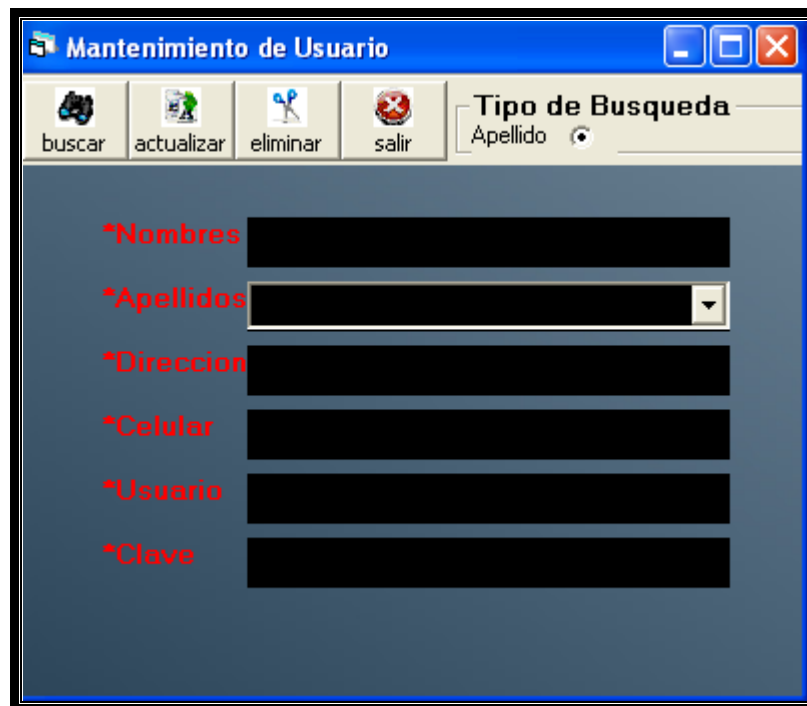


Figura 3.5 Búsqueda de usuario

Fuente: Autores

La figura 3.5 nos muestra la parte de búsqueda de un usuario existente en el sistema, la búsqueda se hace por medio del apellido del usuario, se lo realiza en caso de que el empleado no recuerde su usuario o contraseña asignado.

Una vez que se dicta el apellido del empleado se presiona el botón buscar y ahí aparecerá todos los datos del empleado seleccionado, una vez que aparecen los datos uno puede eliminar ese usuario presionando el botón que dice eliminar, también con el ratón podemos presionar en el combo box y nos desplegará los apellidos de los empleados en caso de no querer usar la voz para controlar este menú.

Cuando se presiona el botón salir, cerramos esa ventana y volvemos a la pantalla principal.



Figura 3.6 Actualización de usuario

Fuente: Autores

Cuando aparezcan los datos de los empleados si una persona quisiera modificar el contenido por motivos de que estén mal ingresados o simplemente para cambiar contraseña se creó el botón actualizar para poder modificar cualquier información (Figura 3.6).

3.2.2 Control del Sistema

En el menú inicio tenemos las opciones de Escaneo y Salir (Figura 3.7).

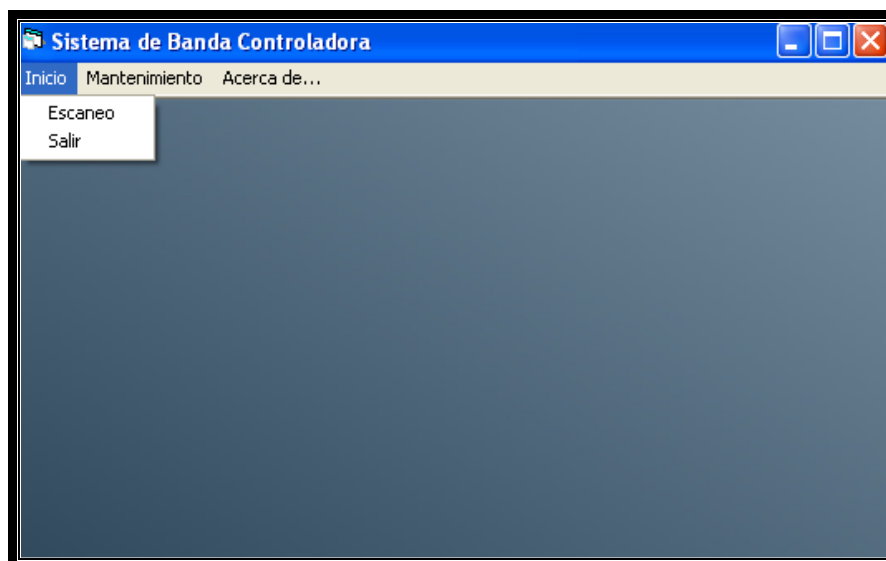


Figura 3.7 Menú Inicio

Fuente: Autores

Diciendo Inicio->escaneo accedemos al menú donde se encuentra la parte de control de las máquina transportadora.

Si decimos inicio->salir el sistema se cierra.

Se crearon las validaciones del sistema que consiste en detectar si el puerto COM1 está habilitado para ser usado en el sistema, caso contrario debe mostrar un mensaje de que el puerto no está disponible, lo cual debería de ser liberado del computador para que no vuelva a aparecer ese mensaje (Figura 3.8).

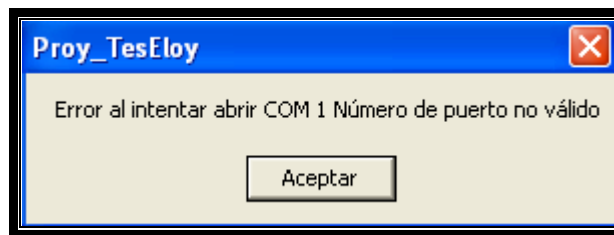


Figura 3.8 Error de COM1

Fuente: Autores

Si el puerto fue encontrado se codificó un mensaje indicándonos que fue abierto el puerto COM1 y diciendo aceptar nos permite ir a la siguiente pantalla que es la de control del sistema (Figura 3.9).



Figura 3.9 Puerto Com Abierto

Fuente: Autores

3.2.3 Instrucciones del sistema

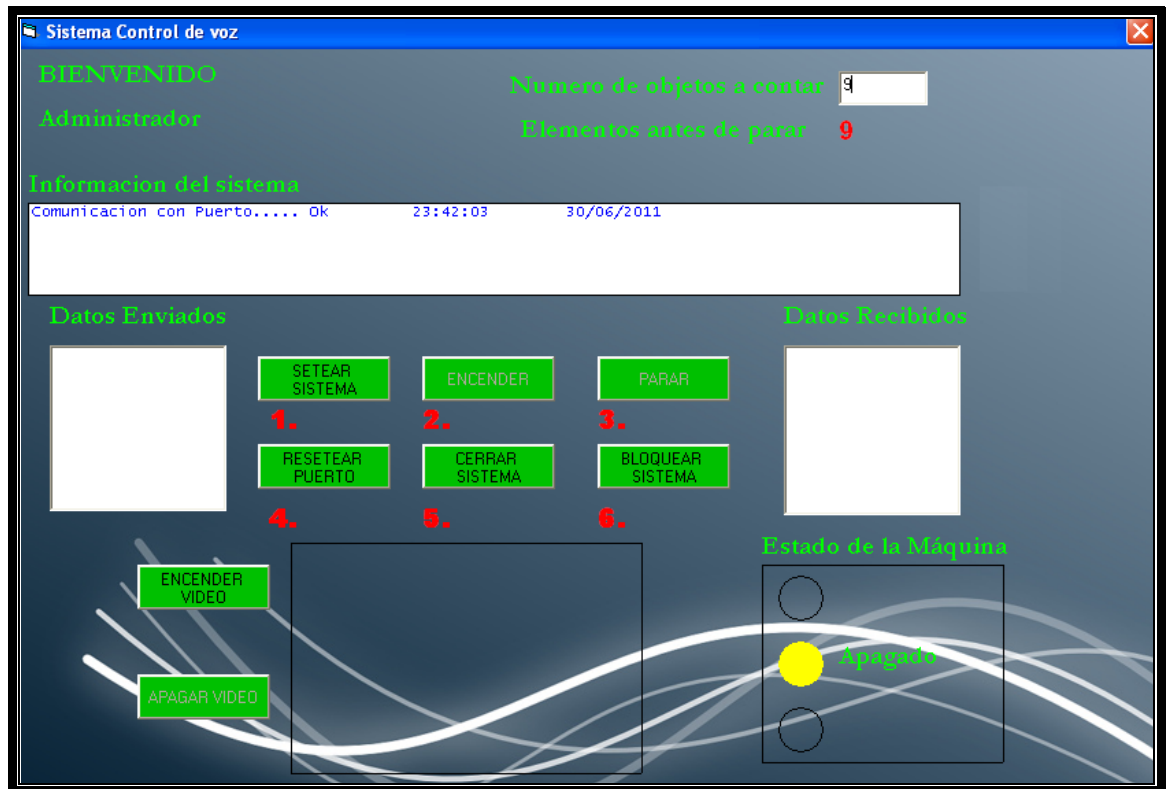


Figura 3.10 Pantalla de control del sistema

Fuente: Autores

(Figura 3.10) Pantalla de instrucciones, donde se ve el contador que puede registrar máximo 99 números, los botones de encendido, de seteo, de parar (en caso de emergencias) resetear puerto, eso es cuando el puerto no ha podido ser seteado correctamente y cerrar que nos permite salir del sistema, todo con las validaciones respectivas, en los cuadros blancos a los extremos sirven para verificar que se esté transmitiendo los datos desde el cpu a la máquina y el central me indica el estado de la conexión con el puerto.

También apreciamos los botones de encendido y apagado de la cámara de vigilancia que está apuntando a la máquina transportadora y así poder visualizar el funcionamiento correcto de nuestro proyecto.

Las luces indicadores indican si la máquina está encendida, apagada o si tiene algún error, están sincronizadas con las luces físicas que se encuentran en la máquina transportadora.

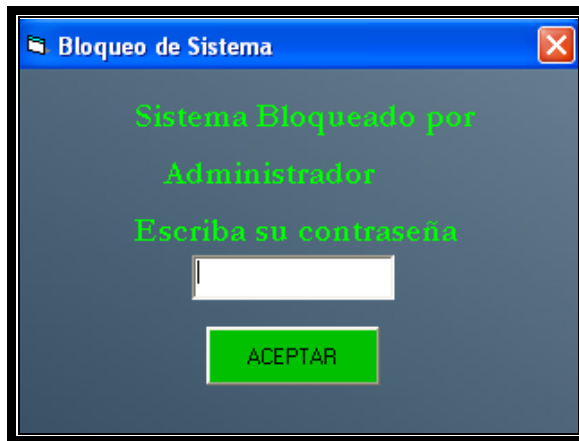


Figura 3.11 Bloqueo del sistema

Fuente: Autores

Como medida de seguridad se diseño un botón de bloqueo en caso de que el operador tenga que salir del lugar y así evitar que alguna otra persona pueda manipular el sistema y esta validado para que acepte la contraseña del usuario que está utilizando el sistema (Figura 3.11).

3.3 Diseño de la Base de Datos

La base de datos está hecha en MySQL que es un motor sql el cual no es muy pesado y puede ser usado por computadores que no poseen mucha capacidad de procesamiento y así no haya que adquirir nuevos equipos.

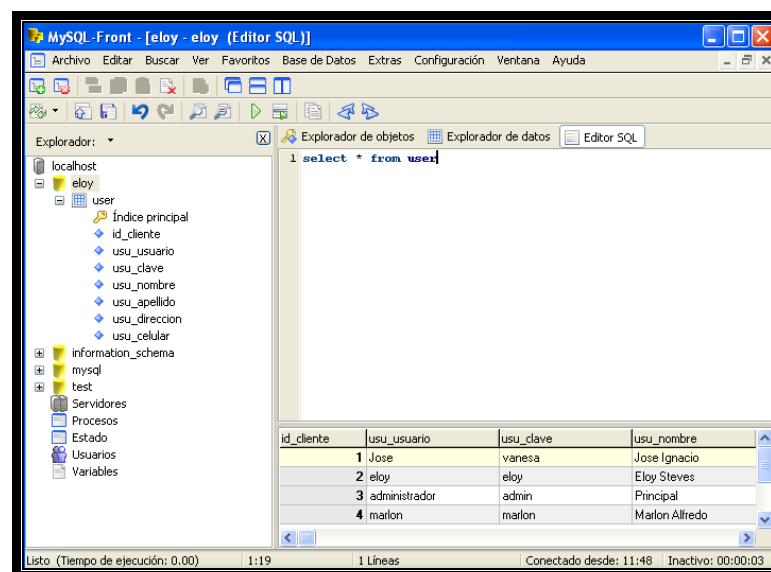


Figura 3.12 Base de Datos Mysql

Fuente: Autores

Se creó la tabla principal Eloy que es donde contienen los campos necesarios para el ingreso del usuario (Figura 3.12).

Los campos creados son:

- Id_cliente: código numérico del cliente, es autonumérico
- Id_usuario: usuario del sistema
- Usu_clave: clave de usuario del sistema
- Usu_nombre: nombre de la persona
- Usu_apellido: apellido de la persona
- Usu_direccion: dirección de la persona
- Usu_celular: número de teléfono de la persona

3.3.1 Configuración ODBC Para MySQL

MySQL es una de las base de datos más rápidas y potentes de la actualidad si a esto le agregamos lo rápido que se torna desarrollar bajo Visual Basic sin dudas corremos con grandes ventajas a la hora de crear potentes aplicaciones. No solo para acceder de forma local a nuestra base de datos sino también puede que necesitemos acceder a un servidor MySQL localizado en otra parte del mundo y hasta que este corra bajo Linux. Primero debemos de proceder a la instalación del ODBC para integrar Mysql5.1 con VB6.0 dicho componente es mysql-connector-odbc-5.1.0-alpha-win32 para proceder a la configuración del ODBC seguir los siguientes pasos: Inicio >Panel de Control>Herramientas Administrativas>Orígenes de datos ODBC.

Debemos crear la referencia a ADO (Ado Data Control 6.0) A Visual Basic, para esto vamos al menú Proyecto > Componentes... y ahí seleccionamos Microsoft Ado Data Control 6.0.

3.4 Diseño del Hardware

A continuación detallamos como se fue desarrollando el hardware para que sea usado en el proyecto de control de voz.

3.4.1 Módulo de Transmisión

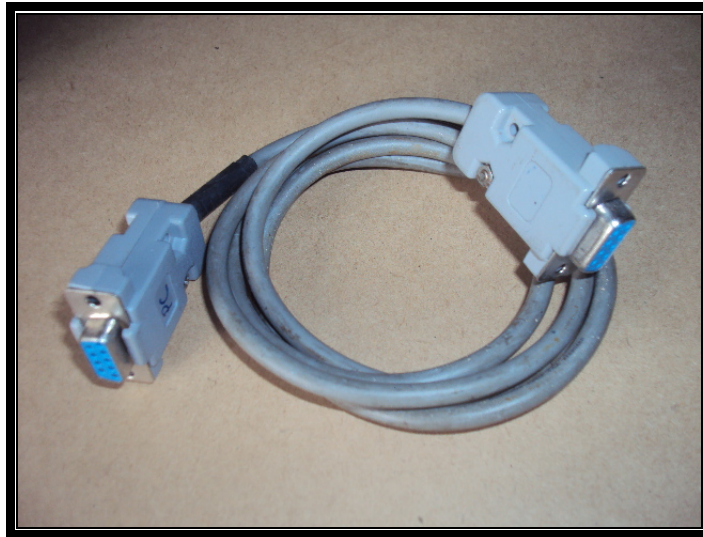


Figura 3.13 Cable Serial RS-232

Fuente: Autores

Este cable Figura 3.13 será el encargado de unir la computadora (PC) con el módulo de transmisión (Hardware) mediante los conectores DB-9 del puerto serial usando el protocolo de comunicación RS-232

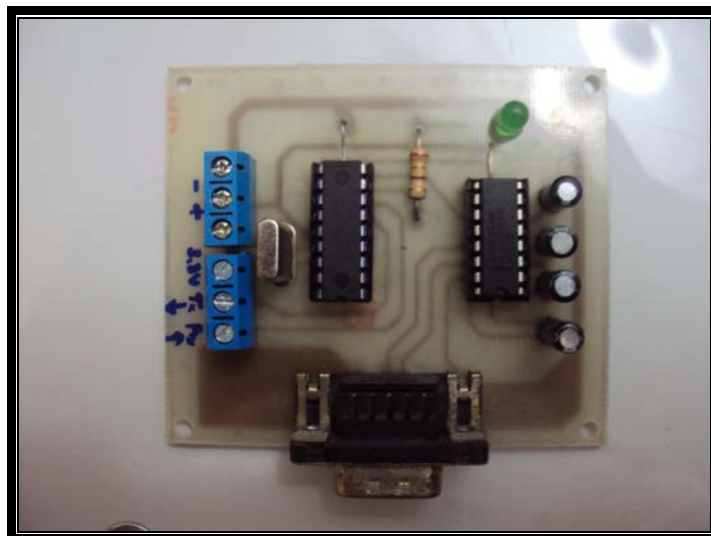


Figura 3.14 Módulo de transmisión PC - XBee

Fuente: Autores

La siguiente figura 3.14 muestra nuestro módulo de transmisión y recepción, este será el encargado de comunicar el programa con la máquina (Software con Hardware), el mismo que está conformado por los circuitos integrados MAX232 y el microcontrolador PIC 16F628.

El funcionamiento de esta tarjeta es de la siguiente manera. El MAX232 es el encargado de acoplar los niveles de voltaje del PC en el protocolo RS232 (+12v,-12v) con los niveles de voltaje TTL del microcontrolador (+5v,0v) de forma serial. Una vez acoplado los niveles de voltaje para la comunicación serial, esta información es receptada por el microcontrolador PIC, ejecutándola a la vez, y según sea la información recibida el decidirá que hacer, por ejemplo enviar un dato serialmente a los módulos de comunicación inalámbrica XBEE para ordenar el encendido o el apagado de la máquina, así como también puede recibir una información recibida por el XBEE (enviada por el módulo Receptor-Transmisor) para procesarla y definir que hacer, ya sea la de mostrar un dígito como también mostrar una alarma y esas órdenes enviarlas serialmente a la computadora usando el MAX232 de intermediario. Al momento de receptar una información el diodo LED verde se enciende indicando la llegada de algún dato.

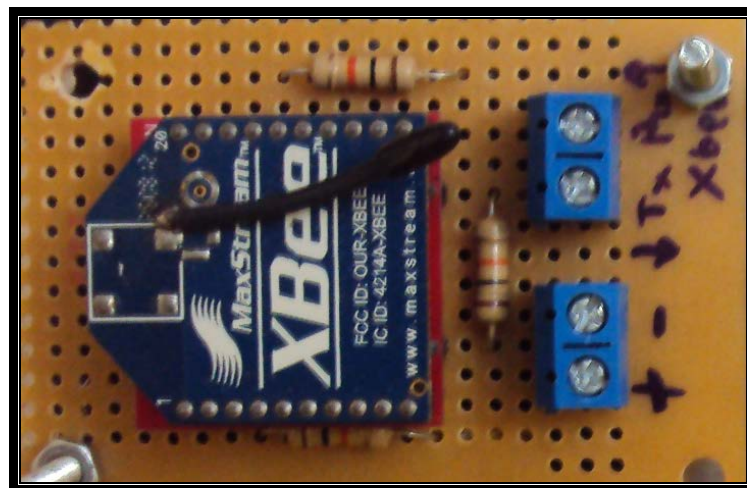


Figura 3.15 XBee transmisor

Fuente: Autores

Los módulos de comunicación XBEE (Figura 3.15) son dispositivos electrónicos utilizados para transmitir información de forma inalámbrica, la información que transmiten la reciben de forma serial y de la misma manera al momento de recibir una información la transmiten serialmente al microcontrolador.

Estos dispositivos son de bajo consumo, por eso utilizan una fuente de alimentación de +3.3v pero de gran potencial al transmitir, según sea su modelo pueden transmitir hasta un poco más del kilómetro de distancia, y puede usarse un módulo similar para utilizarlo como antena repetidora y expandir más la distancia de su alcance.

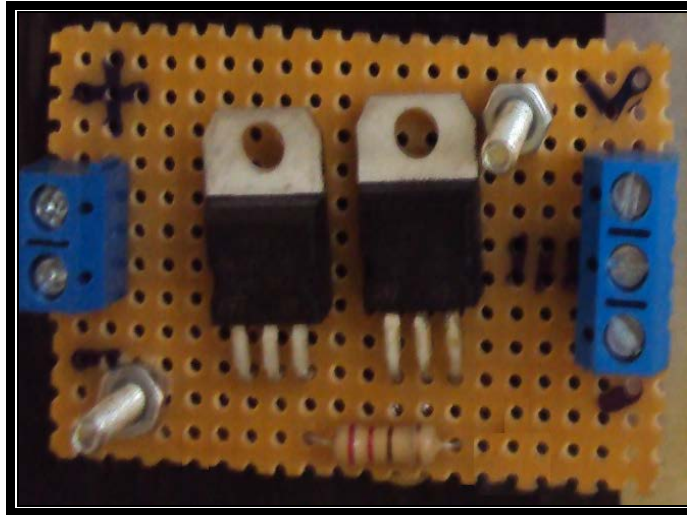


Figura 3.16 Reguladores de voltaje de 5v y 3.3V

Con los reguladores (Figura 3.16) evitamos que pasen más que el voltaje deseado y así evitar alguna sobrecarga o calentamiento

3.4. 2 Módulo de Recepción



Figura 3.17 Fuente de Poder

Fuente: Autores

La siguiente figura (3.17) muestra la fuente de poder de nuestro sistema. Dicha fuente consta de 3 reguladores de voltaje positivo los cuales son, el LM7805, LM7812 y el LM33 los cuales regulan un voltaje de entrada (15v) en +5v, +12, y +3.3v respectivamente.

Esta fuente de poder será la encargada de alimentar al módulo de recepción-transmisión, al módulo de transmisión inalámbrica (XBEE) y a los Relés para la etapa de potencia.

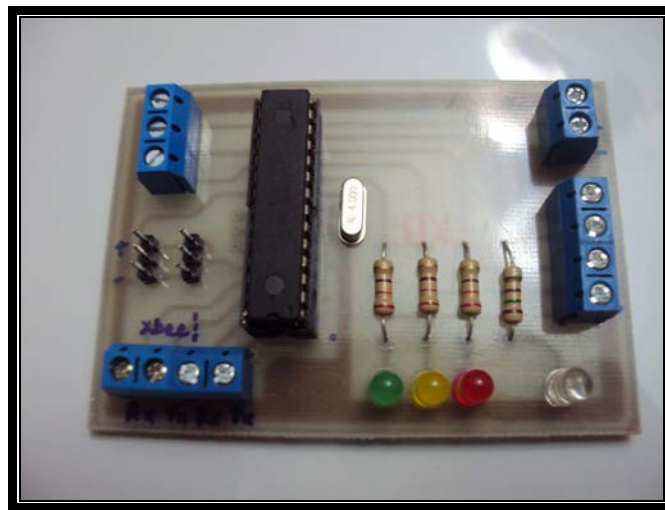


Figura 3.18 Módulo Transmisión Recepción

Fuente: Autores

En la siguiente imagen (Figura 3.18) mostramos al módulo de recepción-transmisión el mismo que está conformado por un microcontrolador PIC 16F871 como cerebro de esta operación.

Su funcionamiento es similar al del módulo de Transmisión - Recepción con la diferencia que ya no vamos a acoplar los niveles de voltaje para la comunicación serial con un computador, es decir no vamos a utilizar al MAX 232, sino que recibimos o transmitimos la información directamente por los XBEE de manera serial. Según sea la orden que reciba, podemos mandar la señal para encender el motor (blanco), y según sea la falla que presente podemos enviar a encender la luz indicadora de falla o alarma (roja), cuando entra en funcionamiento la máquina, este automáticamente enciende la luz correspondiente a marcha (verde), y si se manda a detener el sistema, automáticamente enciende la luz correspondiente a par (amarillo), además de eso, este módulo se comunica con el módulo de visualización de forma

serial, es decir el módulo de visualización envía los datos del conteo y el módulo de recepción – transmisión transmite de forma simultánea el valor del numero que se va contando.

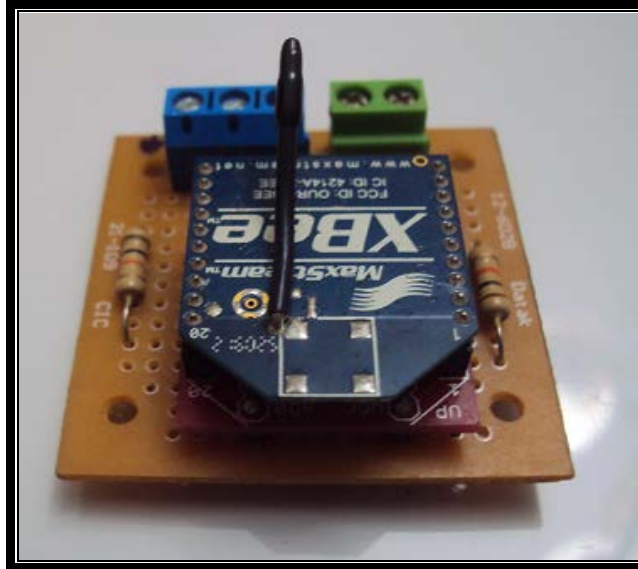


Figura 3.19 Xbee Receptor

Fuente: Autores

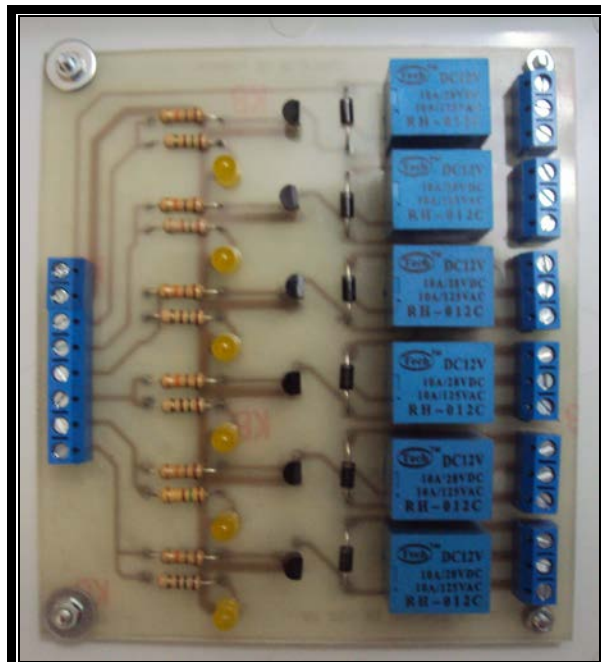


Figura 3.20 Relés controladores

Fuente: Autores

La figura 3.20 muestra al módulo de fuerza, este módulo es el encargado de accionar los relés o en encender las luces piloto que trabajan con niveles de voltaje superior a los niveles de voltaje con los que trabajan los módulos, es decir, maneja la etapa de potencia (voltajes de 110v).

Su funcionamiento es el siguiente, cuando el módulo de recepción – transmisión envía la señal para encender el motor o alguna luz (que funcionen con 110v), esta señal llega hasta los transistores, los mismos que se encuentran configurados como un switch electrónico, haciendo “excitar” la bobina del relé para que este se accione y logrando así mediante sus contactos internos activar o desactivar alguna carga (motor, luz, etc.) ya que estos contactos soportan grandes corrientes en el orden de Amperios, valores que directamente un microcontrolador no puede manejar, ya que solamente manejan corrientes bajas en el orden de miliamperios y en algunos casos microamperios.

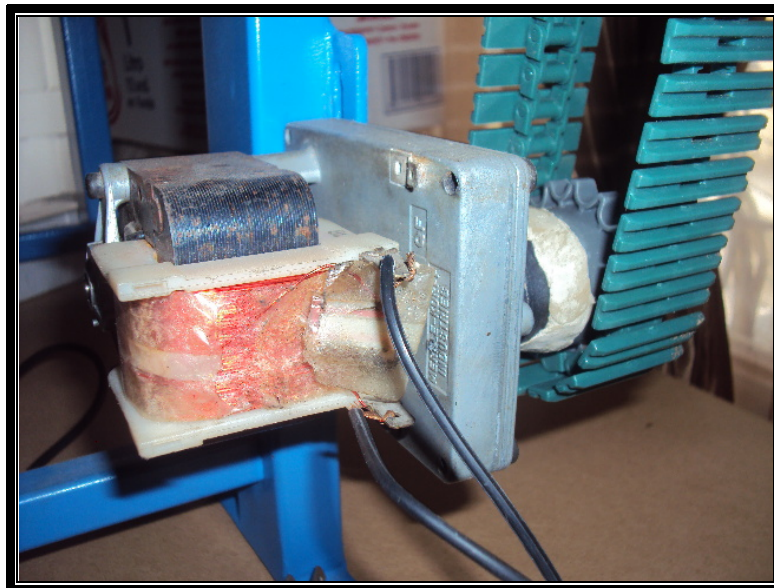


Figura 3.21 Motor AC 110V

Fuente: Autores

La Figura 3.21 muestra un motor AC 110V con caja reductora la cual reduce la velocidad y aumenta la fuerza del torque en la cual va conectada la cadena para la banda de la máquina transportadora.



Figura 3.22 Cadena

Fuente: Autores

Figura 3.22 que muestra la cadena que sirve de unión entre la banda con el motor y permite el desplazamiento de la banda.



Figura 3.23 Sensor detector capacitivo

Fuente: Autores

La figura 3.23 muestra un sensor del tipo capacitivo, lo que significa que cuando algún objeto se atraviesa dentro de su rango de alcance (6cm) este internamente cierra un contacto para enviar la señal de presencia de algún objeto. Según su tecnología puede tener salida transistorizada (gran velocidad de respuesta) o salida vía Relé (más lentos), en este caso tenemos uno con salida transistorizada.

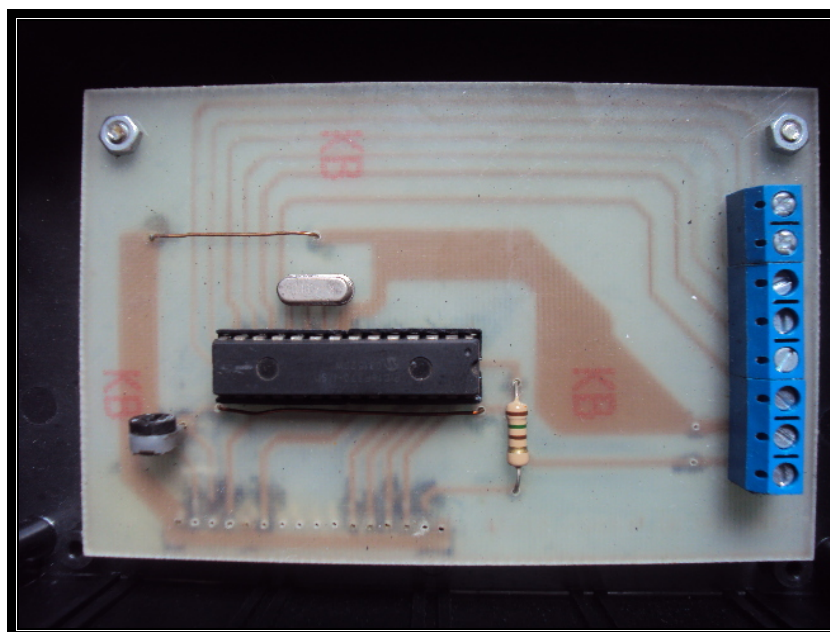


Figura 3.24 Control de visualización al LCD

Fuente: Autores

La Figura 3.24 muestra el módulo de visualización (FIG6), el mismo que consta de un microcontrolador PIC 16F871. Este es el encargado de recibir la señal del sensor (pulsos) y codificarla para luego enviarla vía serial al módulo de recepción – transmisión para luego transmitirla hasta llegar a la PC, además de eso controla una pantalla LCD de 20X4, es decir, 4 filas con 20 líneas de caracteres por fila, donde se muestra el valor real del conteo y algunas leyendas como nombres, etc.



Figura 3.25 LCD 20x4

Fuente: Autores



Figura 3.26 Funcionamiento LCD

Fuente: Autores

Las figura 3.25 se muestra un Módulo LCD 20X4 que es controlado por el PIC 16F871 para poder visualizar la información.



Figura 3.27 Pantalla Inicio LCD

Fuente: Autores



Figura 3.28 Pantalla Integrantes

Fuente: Autores

Las figuras 3.26, 3.27 y 3.28 muestran los diferentes mensajes que muestra la pantalla LCD como son la presentación del proyecto y los nombres de los integrantes.



Figura 3.29 Botoneras de control (marcha – pare)

Fuente: Autores

En la figura 3.29 se muestra un juego de botoneras Marcha (verde), Paro (roja) las mismas que serán utilizadas en caso que falle la comunicación inalámbrica entre la PC y la máquina para poder dar “marcha” o “parar” la máquina de forma manual.

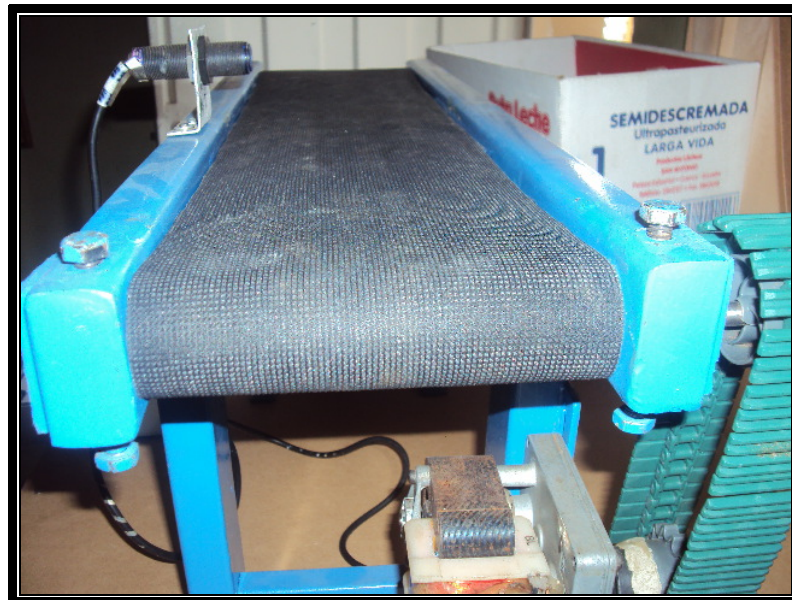


Figura 3.30 Banda transportadora

Fuente: Autores



Figura 3.31 Máquina transportadora

Fuente: Autores

3.5 Pruebas de funcionamiento de los módulos Xbee (Transmisión y Recepción)

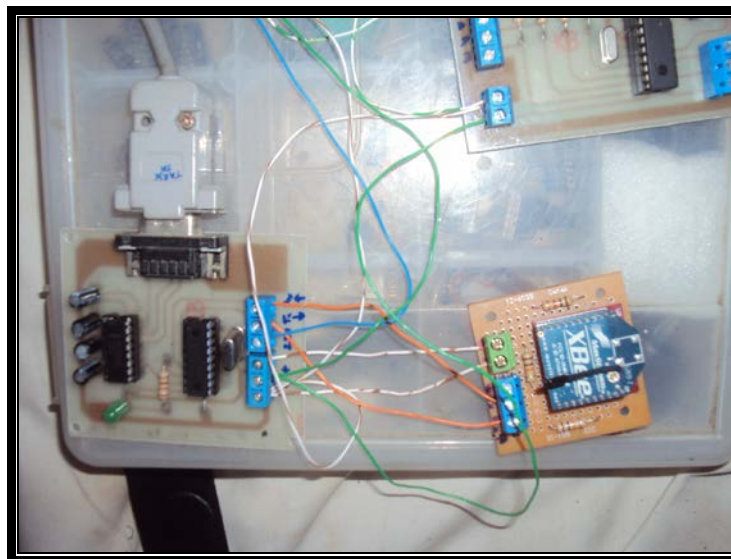


Figura 3.32 Módulo transmisor XBEE

Fuente: Autores

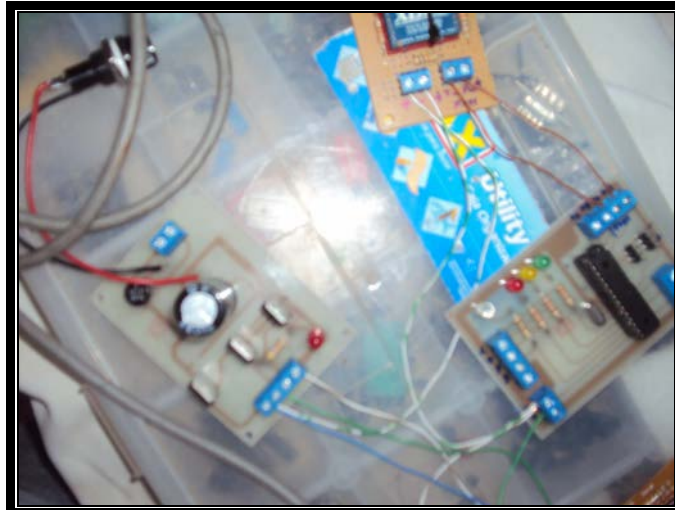


Figura 3.33 Módulo Receptor XBEE

Fuente: Autores

Las figuras 3.32 y 3.33 muestran el módulo de transmisión y recepción interconectado entre ellos para hacer la prueba de envío y recibimiento de datos de los xbee, esta prueba se realiza para verificar que los pic estén correctamente programados para poder hacer que los xbee tengan la función de enviar y transmitir los datos

3.6 Diseño de Circuitos Impresos

A continuación se detallan los diseños de las placas utilizadas, las cuales fueron realizadas con el programa ARES

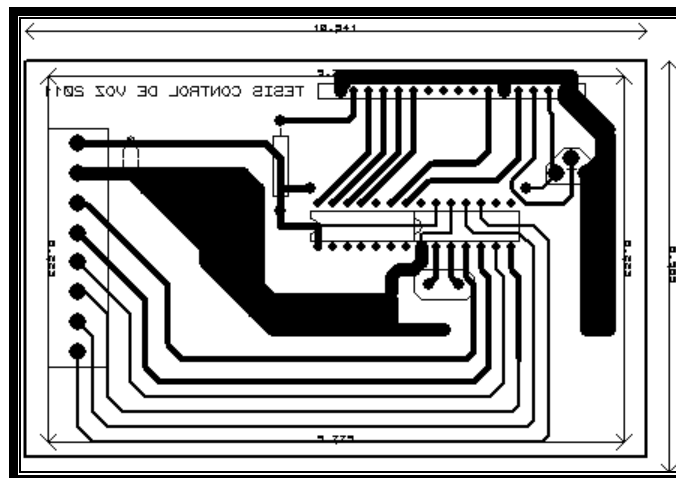


Figura 3.34 Circuito Impreso contador LCD

Fuente: Autores

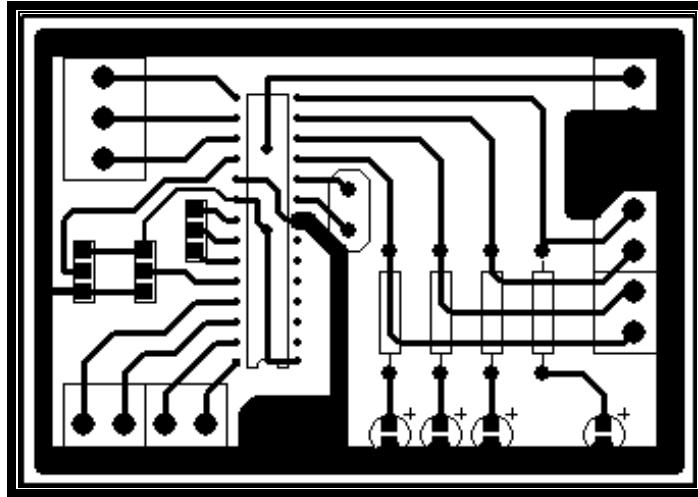


Figura 3.35 Circuito Impreso control

Fuente: Autores

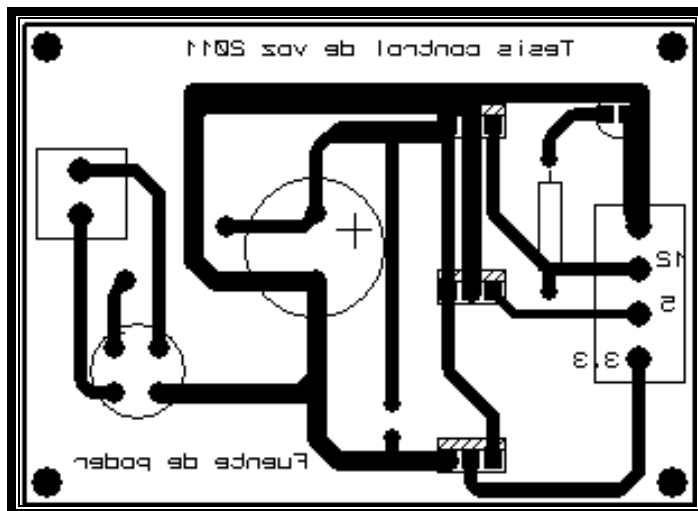


Figura 3.36 Circuito Impreso Fuente de Poder

Fuente: Autores

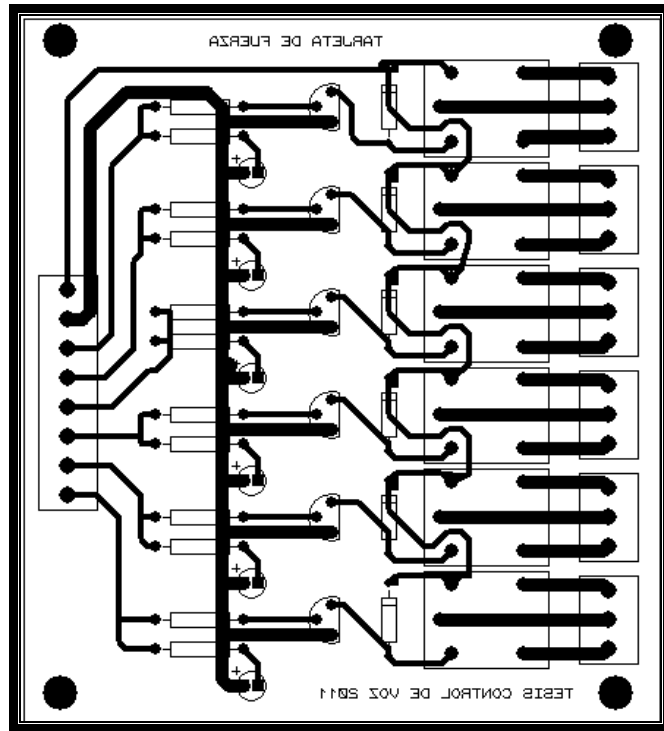


Figura 3.37 Circuito Impreso Control por medio de Relé

Fuente: Autores

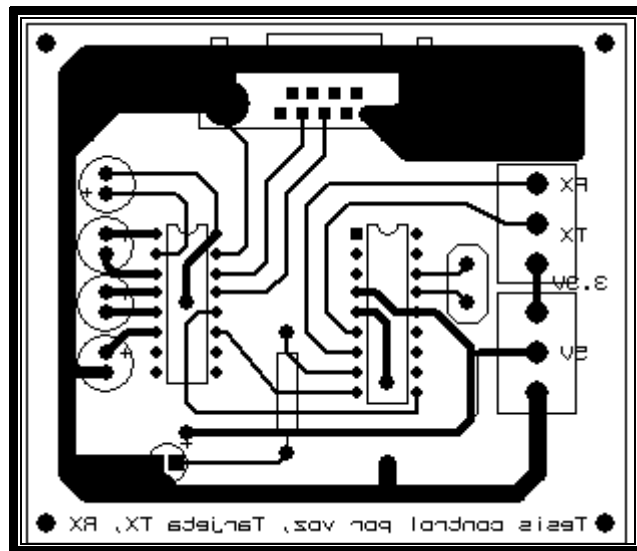


Figura 3.38 Circuito Impreso Transmisor - Receptor

Fuente: Autores

3.7 Diseño del Módulo de transmisión y recepción en proteus

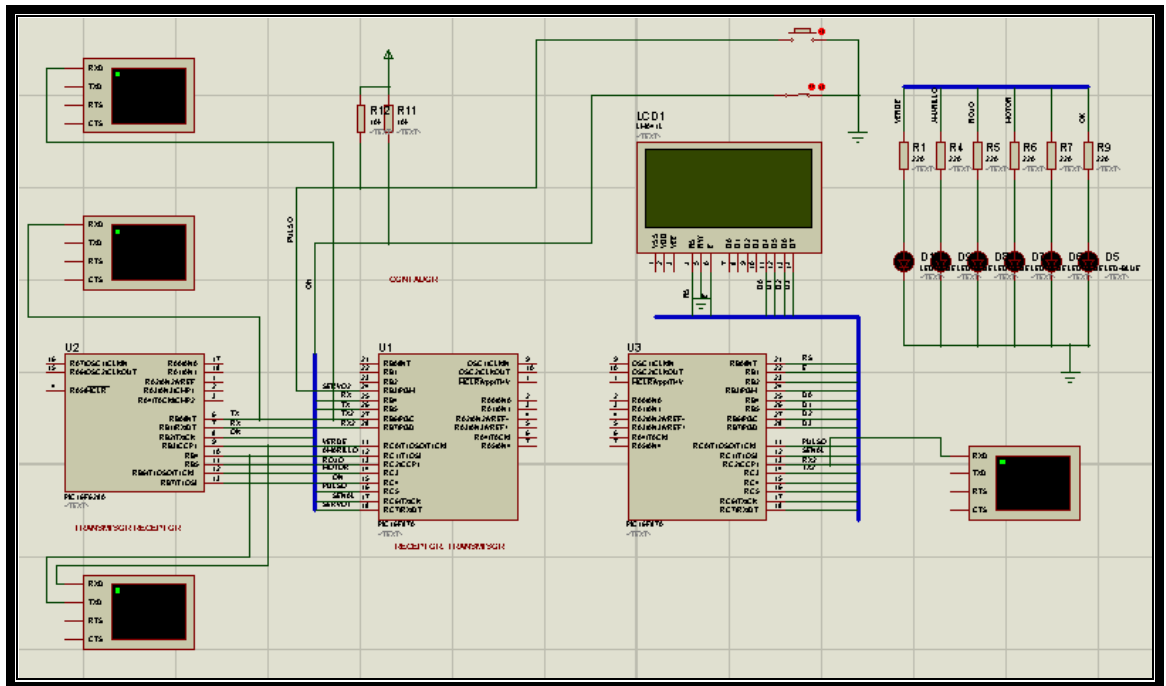


Figura 3.39 Diseño del módulo de envío y recepción

Fuente: Autores

En esta Figura 3.39 se muestra el diagrama esquemático de nuestro proyecto, el mismo que consta de tres microcontroladores PIC los cuales son, el PIC16F628 utilizado en la etapa de Transmisión - Recepción de datos, el segundo PIC es el PIC16F870, el cual conforma la etapa de Recepción - Transmisión, y el ultimo PIC es el PIC16F870 encargado de la etapa de visualización.

En cada etapa además de los PIC utilizamos otros elementos no detallados en esta imagen, como por el ejemplo en la etapa de Transmisión - Recepción utilizamos el circuito integrado MAX232, que se encarga de permitirnos una buena comunicación entre la PC y el PIC, además utilizamos también el XBEE que sería el encargado de realizar la comunicación serial de forma inalámbrica con la etapa Receptora-Transmisora, así mismo utilizamos un diodo LED el cual se enciende cuando transmitimos o recibimos un dato, identificado en el proteus como “ok”.

En la etapa de Recepción - Transmisión, que es la etapa que se encarga de controlar físicamente el sistema, tenemos conectado a la salida la etapa de potencia que es conformada por los transistores y relés, esta etapa (la de potencia) recibe las señales enviadas por el PIC para poder encender las luces de indicación como son la luz

Verde (pin 11), la luz amarilla (pin12), la luz roja (pin13) y el motor (pin14). En nuestro esquema solamente representamos estas salidas con diodos LED, ya que con la señal que obtenemos para encender al diodo LED, es suficiente para encender las luces de 110V ya que esa señal viaja a la etapa de potencia que es la encargada de encenderlos. Además de eso esta etapa de Recepción - Transmisión se comunica vía serial con la etapa de Visualización utilizando los pines 27 y 28 configurados como TX (TX2 para diferenciar del TX del XBEE) y RX (RX2 también para diferenciar) respectivamente ya que constantemente estamos enviando información y recibiendo información de la etapa de visualización, ya que esta nos envía el valor del número actual que se está contando y en caso de no contar, nos envía una alarma.

Así mismo para el módulo de los XBEE, están conectados a los pines 25 y 26 los cuales están configurados como RX y TX respectivamente.

En la etapa de visualización tenemos conectado un LCD de 20 caracteres por 4 filas. Además de eso tenemos un pin configurado para recibir los pulsos que envía en sensor, el cual es el pin 11, para la simulación utilizamos un pulsador, identificado en el proteus como “pulso”, además utilizamos unos pines configurados como TX y RX los mismo que son los pines 13 y 14 respectivamente (en el gráfico se encuentran al revés debido a que estamos usando identificadores de conexión en el proteus, es decir que el pin de transmisión del PIC encargado de la etapa Recepción - Transmisión (pin 27) va conectado al pin de recepción de la etapa de Visualización (pin 14), pero como primero conectamos la etapa de Recepción - Transmisión lleva el nombre de esta (TX2) en la etapa de visualización para indicar la conexión.

Para la simulación de la PC se utiliza la herramienta HYPER TERMINAL que posee el proteus, con esta herramienta podemos simular los comandos que envía la PC a la etapa de Transmisión - Recepción, así como toda comunicación serial en nuestro sistema, ya que así podemos controlar que la información este viajando correctamente y que no presente errores al momento de ensamblar todo el circuito. En esta imagen utilizamos 4, las mismas que están encargadas para supervisar y simular las señales que se envían entre PC-Transmisor/Receptor, para supervisar señales entre Transmisor/Receptor-Receptor/transmisor, por ultimo para señales entre Receptor/Transmisor-Visualizador y la ultima sólo para supervisar una recepción de dato entre los pines.

CAPÍTULO IV

IMPLEMENTACIÓN DEL PROYECTO

En este capítulo detallamos el proceso de acoplamiento de todo el hardware que se ha diseñado para que funcionando en conjunto realicen las tareas para los cuales fueron elaborados.

4.1 Implementación Módulo de transmisión

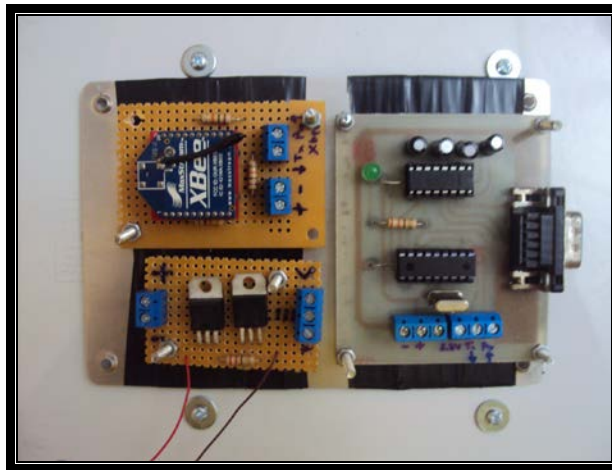


Figura 4.1 Módulo completo de transmisión

Fuente: Autores

En la figura 4.1 comenzamos armando el módulo de transmisión en el cual unimos la tarjetas que contienen el xbee, los reguladores de voltaje y el conector DB9, una vez juntado los tres componentes los guardamos en una caja protectora para evitar daños.

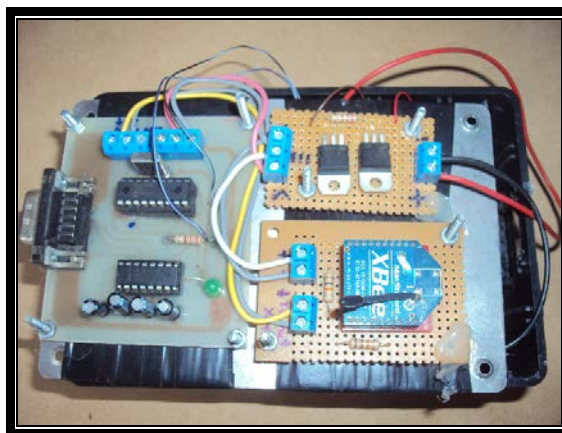


Figura 4.2 Cables para interconectar los diferentes módulos para transmisión

Fuente: Autores

En la figura 4.2 Comenzamos interconectando por medio de los respectivos cables de conexión en las tarjetas y con eso completamos el proceso de transmisión de la señal desde el computador hasta el xbee y la señal será enviada al extremo del receptor.



Figura 4.3 Caja de protección para transmisor

Fuente: Autores

La figura 4.3 nos muestra la caja con la parte de transmisión en su interior, presenta diodos led que comienzan a parpadear cuando va a transmitir datos desde el xbee.

4.2 Implementación Módulo Recepción

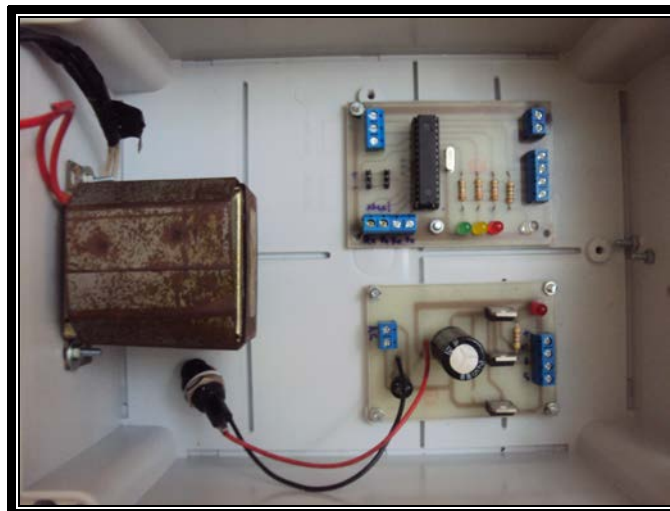


Figura 4.4 Acoplamiento del módulo de recepción

Fuente: Autores

En esta figura 4.4 hacemos el acoplamiento de la parte de recepción del proyecto, donde incorporamos el transformador de 110 a 12v, la fuente de poder y el circuito impreso del módulo de recepción.

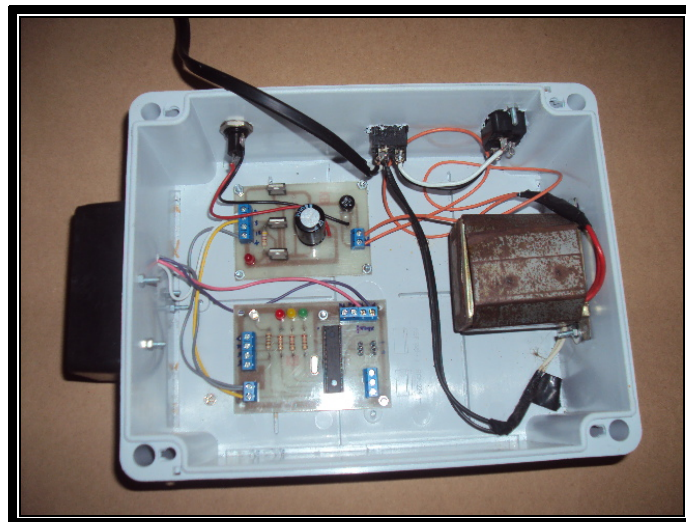


Figura 4.5 Interconectando los circuitos integrados

Fuente: Autores

La figura 4.5 nos muestra la parte de de conexión de los componentes de recepción y envío de datos hacia los relé controladores.

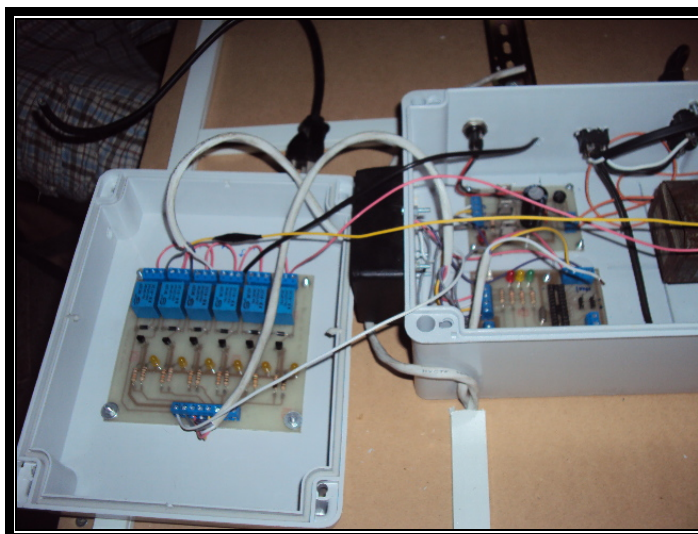


Figura 4.6 Acoplamiento entre el módulo receptor con los relé

Fuente: Autores

En la figura 4.6 se muestra que dentro del mismo cajón protector colocamos los relés controladores y hacemos las respectivas uniones con los demás hardware para completar el módulo de recepción.

4.3 Implementación Luces Piloto



Figura 4.7 Luces Piloto

Fuente: Autores

La figura 4.7 nos muestra las luces indicadoras que se usan en el sistema, las cuales son verde para apagado, amarillo para encendido y rojo que nos indica cuando se ha producido un error en la recepción de datos del sensor.

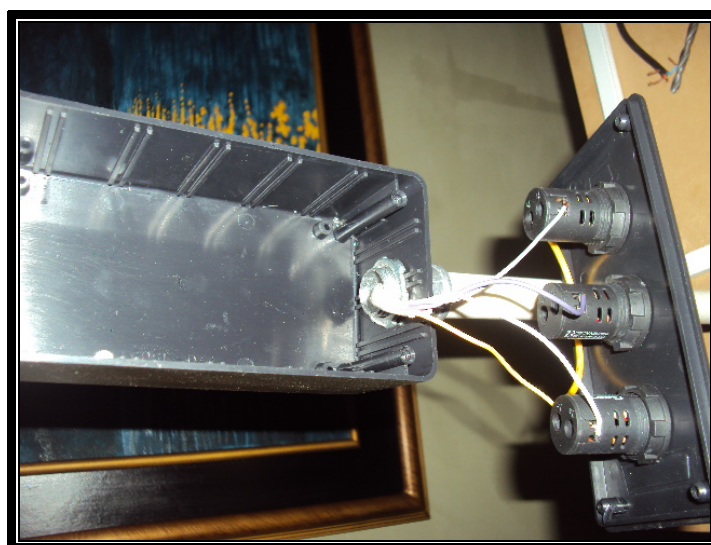


Figura 4.8 Armado Luces piloto

Fuente: Autores



Figura 4.9 Montaje en la base luces piloto

Fuente: Autores

Las luces indicadoras de la figura 4.9 se colocan en una caja protectora para evitar daños, una vez armado se conectan a un lado de la máquina transportadora y están alzadas con un tubo de plástico para que puedan ser visualizadas desde lejos por parte del operador y verificar el estado en que se encuentra la máquina ya sea marcha, detenido o en error.



Figura 4.10 Luz piloto encendida

Fuente: Autores

4.4 Implementación LCD

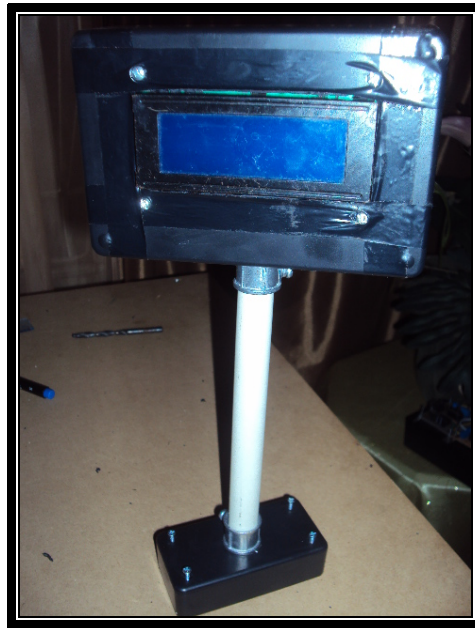


Figura 4.11 Colocación del Led en la caja y base

Fuente: Autores

En figura 4.11 apreciamos el LCD que se coloca en el otro extremo de la máquina transportadora, nos indica que proceso se está realizando en ese momento (marcha, para o error), también nos va mostrando la cantidad de elementos que está contando el sensor capacitivo.

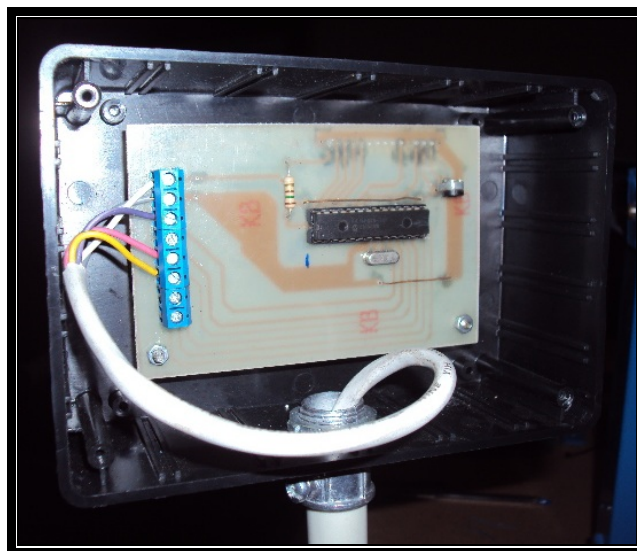


Figura 4.12 Conexión LCD

Fuente: Autores

4.5 Implementación sensor

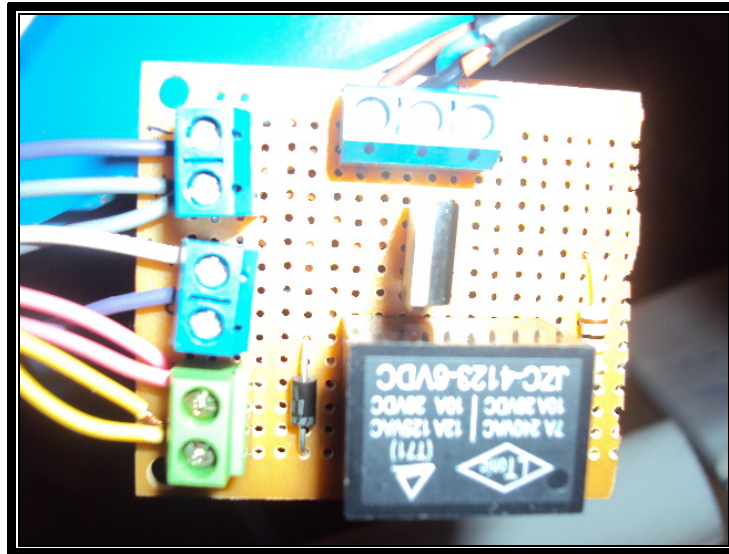


Figura 4.13 Relé para el sensor capacitivo

Fuente: Autores

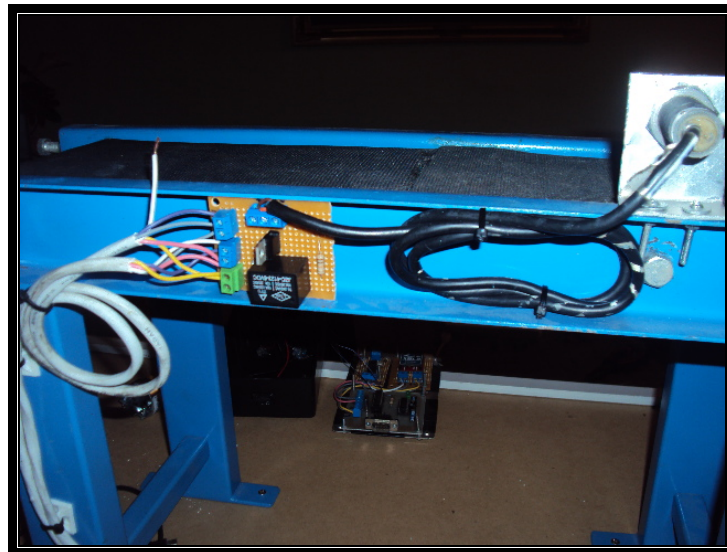


Figura 4.14 Implementación del sensor capacitivo

Fuente: autores

4.6 Armado de la estructura de la máquina Transportadora



Figura 4.15 Vista general del proyecto

Fuente: Autores

La figura 4.15 nos da una vista del montaje de los componentes de la máquina transportadora donde en la parte superior se colocarán la máquina transportadora, el sensor capacitivo, el LCD, las luces indicadoras y los botones de marcha y paro para manipular la máquina transportadora de forma manual también.

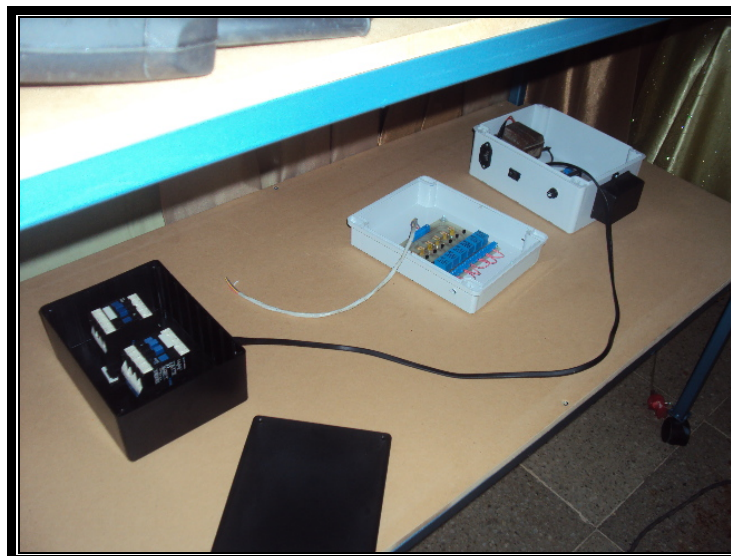


Figura 4.16 Colocación de las cajas en la máquina transportadora

Fuente: Autores

En la figura 4.16 nos muestra la parte inferior de la plataforma de montaje donde van colocados el módulo de recepción y los relé controladores.

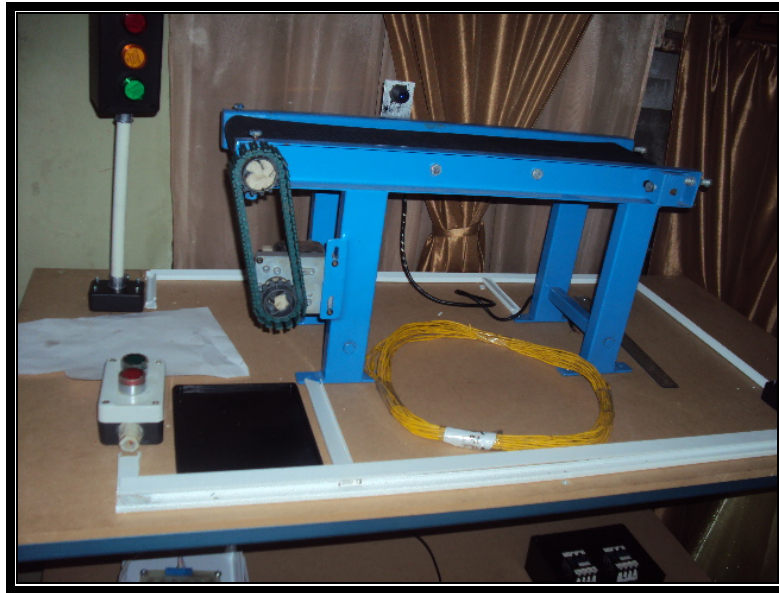


Figura 4.17 Canaletas parte superior

Fuente: Autores

En la figura 4.17 se aprecian las canaletas que van en la parte superior y en la parte inferior del proyecto donde va a pasar el cable eléctrico que interconectará todos los dispositivos a la corriente eléctrica.

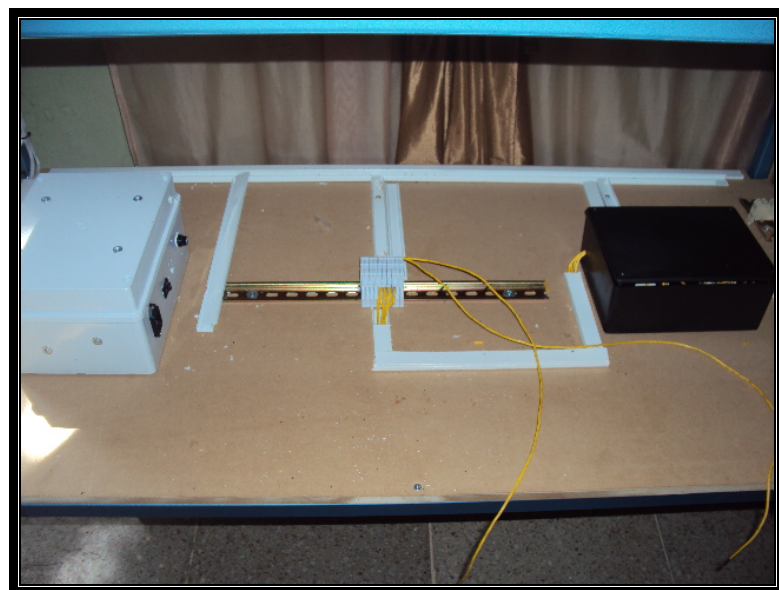


Figura 4.18 Canaletas parte de Abajo

Fuente: Autores



Figura 4.19 Borneras

Fuente: Autores

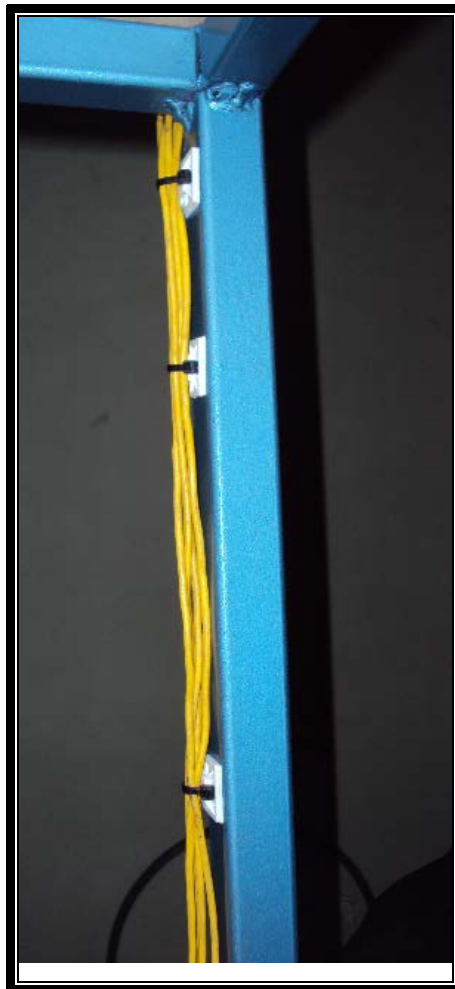


Figura 4.20 Colocación de amarras a los cables de electricidad

Fuente: Autores

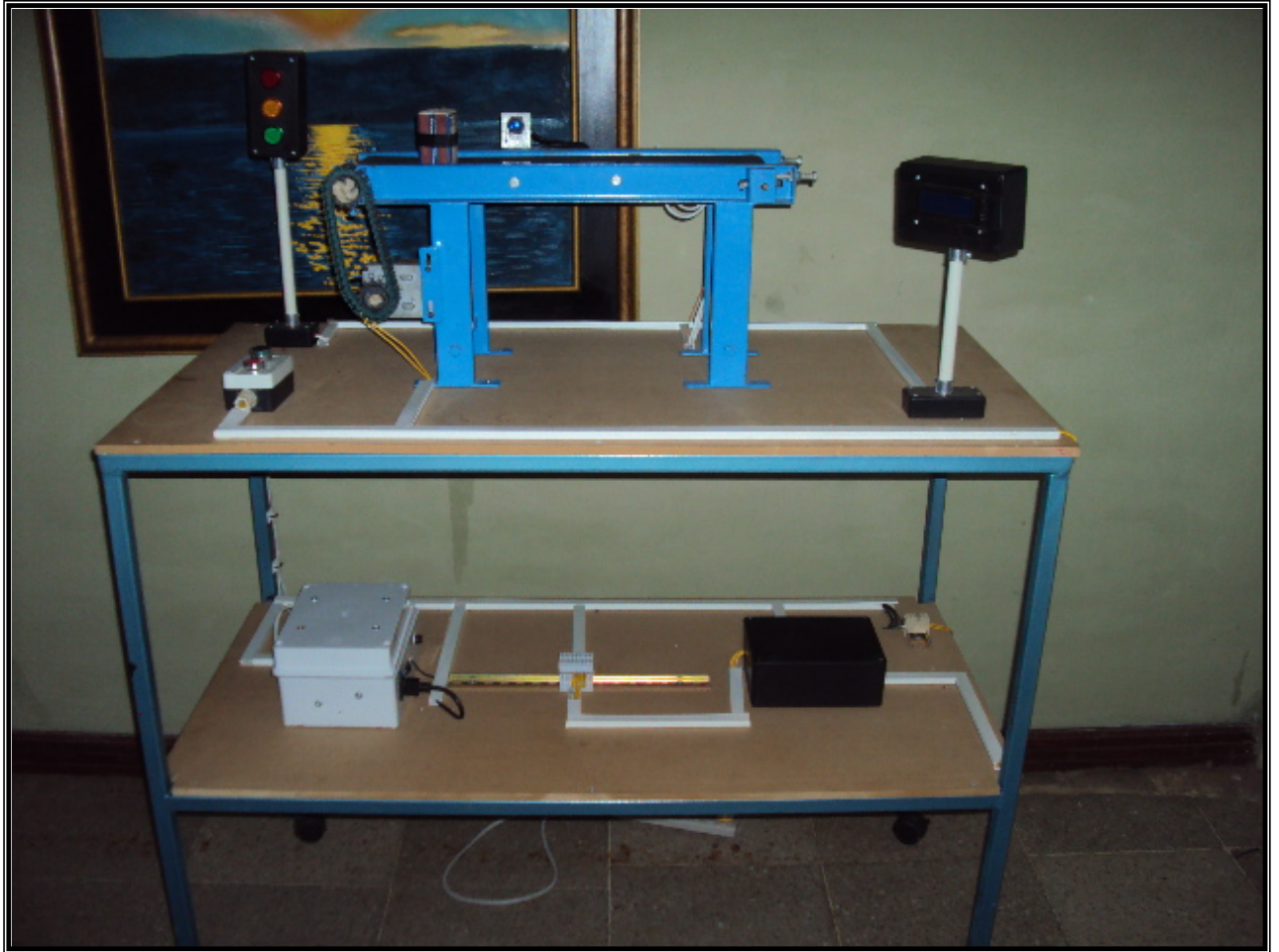


Figura 4.21 Máquina Transportadora completa

Fuente: Autores

La figura 4.21 muestra el diseño completamente armado de la máquina transportadora.

CONCLUSIONES

- La implementación de un sistema de control por medio de comandos de voz como el mostrado en el proyecto de tesis, representa una nueva forma de manejo de elementos eléctricos y electrónicos, ya que al no usar las manos como herramienta de control permite el uso a un mayor número de personas como por ejemplo discapacitados.
- Las seguridades configuradas en el sistema presentado en este proyecto, garantizan que sólo personal autorizado tenga acceso a la información.
- Microcode Studio fue una herramienta muy importante en el desarrollo del proyecto, ha sido el software que ha ayudado a interconectar el hardware con el software.
- Con el avance acelerado de los programas de computación, cada vez aparecen mejores software que con las librerías que aportan para el diseño en programación (ocx, dll) ayudan a que la captura de la voz sea mas precisa.
- Se ha podido cumplir con el proyecto propuesto, lograr desarrollar un sistema de control de voz que combine el uso del hardware con el software, teniendo sus limitaciones como las que fueron mencionadas con anterioridad.

RECOMENDACIONES

- Mantener la cabina de control fuera de cualquier clase de ruido externo para evitar que se mezclen con los comandos que el operador envía al sistema
- Usar un dialecto claro y preciso en el momento de dar los comandos al sistema de control de voz.
- Cerrar el sistema adecuadamente para evitar errores en el puerto de comunicación COM1 y evitar que en un próximo censo arroje datos erróneos.
- Evitar que el hardware entre en contacto con factores que lo puedan deteriorar (Sol extremo, Líquidos, variaciones de voltaje).
- Usar el sensor solamente en distancias no mayor a 10 cm para evitar que se salga del rango de alcance.
- No interponer objetos que obstruyan al xbee para evitar atenuación en el rango de envío de señal.

ANEXOS

Anexo A: CÓDIGO FUENTE DEL SISTEMA

Código Para realizar la conexión con el ODBC.

```
Public rs As ADODB.Recordset Public con As Connection
Public Sub Conectar() Set con = New ADODB.Connection con.ConnectionString =
"DRIVER = {MySQL ODBC 5.51 Driver}; Server = localhost;
Uid=root; pwd=pasw; database=prueba; connection "
con.Open
If con.State = 1
Then MsgBox "Conectada"
Else MsgBox "Error"
End If
End Sub
Sub registro() Set rs = New ADODB.Recordset
With rs
.ActiveConnection = con
.CursorLocation = adUseClient
.CursorType = adOpenDynamic
.LockType = adLockOptimistic
.Open "Select * from usuario"
End With
End Sub
Private Sub Form_Load()
Conectar
End Sub
```

Código Visual Basic 6 del sistema de control de voz

```
Private Sub Form_Load()
Comsj.InputLen = 0
Comsj.RThreshold = 1
Comsj.SThreshold = 1
'muxi.Controls.stop
'globalzone = ""
```

```

escaner = False
paro = False
esactivado = False
abro_puerto
MsgBox "Ingrese valor a contar"
Text4.SetFocus
End Sub

Public Sub cierre_puerto()
On Error GoTo manejar_errores 'Protejo frente al error.
Comsj.PortOpen = False
MsgBox ("Puerto COM cerrado OK")
GoTo Salir
manejar_errores:
MsgBox ("Error al intentar cerrar COM" + Err.Description)
Resume Salir
Salir:
End Sub

Public Sub abro_puerto()
On Error GoTo manejar_errores
Comsj.CommPort = 1
Comsj.Settings = "9600,N,8,1"
GoTo Salir
manejar_errores:
MsgBox ("Error al intentar abrir COM 1 " + Err.Description)
Frame1.Visible = True
Exit Sub
Salir:
Comsj.PortOpen = True
MsgBox ("Puerto COM 1 abierto OK")
Form1.Show
limpialista1 (List1.ListCount)

```

```

info = "Comunicacion con Puerto..... Ok" + Chr(9) + Chr(9) + CStr(Time) + Chr(9)
+ CStr(Format(Date, "dd/mm/yyyy"))
List1.AddItem info
End Sub

```

```

Public Sub seteo_at()
Dim comandos As Variant
limpialista1 (List1.ListCount)
info = "Sistema Iniciando y Seteando valor a contar....." + Chr(9) + CStr(Time) +
Chr(9) + CStr(Format(Date, "dd/mm/yyyy"))
List1.AddItem info
esactivado = False
If Len(Text4.Text) = 2 Then
dec = Right(Text4.Text, 1)
uni = Right(Text4.Text, 2)
sendmsj = uni + Chr$(13) + Chr$(10)
Comsj.Output = sendmsj
tiempo (2)
sendmsj = dec + Chr$(13) + Chr$(10)
Comsj.Output = sendmsj
Else
sendmsj = Text4.Text + Chr$(13) + Chr$(10)
Comsj.Output = sendmsj
End If
tiempo (3)
List2.AddItem sendmsj
sendmsj = "E" + Chr$(13) + Chr$(10)
Comsj.Output = sendmsj
tiempo (3)
List2.AddItem sendmsj
limpialista1 (List1.ListCount)
End Sub

```

```

Private Sub comsj_OnComm()

```

```

DoEvents
If Comsj.CommEvent = comEvReceive Then
    receivemsj = Comsj.Input
End If
DoEvents
If receivemsj <> "" And esactivado = False Then
    If receivemsj = "OK" Then
        Command5.Enabled = True
        esactivado = True
        Call okidoki(receivemsj)
    End If
End If

If receivemsj <> "" And escaner Then
    If receivemsj = "F" Then
        List4.AddItem "PROCESO FINALIZADO"
        Call finish
    End If
    If receivemsj = "EM" Then
        MsgBox "Problemas con el Motor"
    End If
    If receivemsj = "ES" Then
        MsgBox "Problemas con el Sensor"
    End If
End If

pepito = Len(receivemsj)
If esactivado And (pepito > 2 And pepito < 4) Then 'ojo enviar valores 001
trato (receivemsj)
Else
List3.AddItem receivemsj
End If
End Sub

```

```

Public Sub finish()
Label15.Caption = ""
Text4.Text = ""
escaner = False
paro = False
Command6.Visible
esactivado = False
End Sub

```

```

Public Sub okidoki(msj As Variant)
info = "Sistema seteado correctamente" + Chr(9) + Chr(9) + CStr(Time) + Chr(9) +
CStr(Format(Date, "dd/mm/yyyy"))
List1.AddItem info
List3.AddItem msj
Command6.Enabled = False
Command5.Enabled = True
End Sub

```

```

Public Sub escaneo()
sendmsj = "M" + Chr$(13) + Chr$(10)
tiempo (1)
Comsj.Output = sendmsj
List2.AddItem sendmsj
End Sub

```

```

Public Sub trato(recibido As String)
Label15.Caption = ""
cadena = recibido
Label15.Caption = cadena
End Sub

```

```

'=====LIMPIEZA=====
'=====
=====

```

```
Private Sub limpiarlista1(fila As Variant)
```

```
    If fila = 6 Then
```

```
        'Call Event_Succes_Log
```

```
        List1.Clear
```

```
    Else
```

```
        Exit Sub
```

```
    End If
```

```
End Sub
```

```
Private Sub limpiarlista54(fila5 As Variant, fila4 As Variant)
```

```
    If fila5 = 6 Then
```

```
        List5.Clear
```

```
        List6.Clear
```

```
    End If
```

```
    If fila4 = 6 Then
```

```
        Call Event_Intru_Log
```

```
        List4.Clear
```

```
    End If
```

```
End Sub
```

```
Private Sub Label7_Click()
```

```
End Sub
```

```
Public con As Connection
```

```
Public info As Variant
```

```
Public sendmsj As Variant
```

```
Public receivemsj As Variant
```

```
Public escaner As Boolean
```

```
Public escactivado As Boolean
```

```
Sub cerrar_con()
```

```
    If con <> "" Then
```

```
        con.Close
```



```

con = ""
End If

End Sub

Public Sub tiempo(k As Integer)
t = Timer: Do Until Timer > t + k: Loop
End Sub

Sub conexion()
Set con = New ADODB.Connection
con.ConnectionString = "DRIVER={MySQL ODBC 5.1 Driver};
SERVER=localhost; DATABASE=eloy; PWD=root; UID=root; OPTION=3"
con.Open
End Sub

Public cont As Integer
Public tiesto As Boolean

Private Sub Command1_Click()

Dim f As New MDIForm1
If (cont <= 2) Then
tiesto = False
Dim tabla As ADODB.Recordset
SQL = "SELECT * FROM user "
Set tabla = New Recordset
tabla.Open SQL, con, adOpenDynamic, adLockReadOnly
Do Until tabla.EOF
If (tabla!usu_usuario = Text1.Text) And (tabla!usu_clave = Text2.Text) Then
If tabla!usu_usuario = "administrador" Then
f.mman.Enabled = True
f.Show
Else
f.Show
End If
End If

```

```

        tiesto = True
    End If
    tabla.MoveNext
Loop
    If tiesto = False Then
        MsgBox "Error de Clave", vbCritical, "Clave Erronea"
        Text1.Text = ""
        Text2.Text = ""
        cont = cont + 1
        Text1.SetFocus
    Else
        Unload Me
    End If
Else
    MsgBox "Agotastes Numero de Intentos Validos", vbInformation, "Clave para
    Acceso"
    Unload Me
End If
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Call conexion
    cont = 0
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Call cerrar_con
End Sub

Private Sub Label1_Click()

```

```

End Sub
Private Sub Command1_Click()
    Dim tabla As ADODB.Recordset
    SQL = "SELECT * FROM user "
    Set tabla = New Recordset
        tabla.Open SQL, con, adOpenDynamic, adLockReadOnly
    Do Until tabla.EOF
        If (tabla!usu_usuario = MaskedTextBox4.Text) And (tabla!usu_clave =
MaskedTextBox6.Text) Or (tabla!usu_apellido = MaskedTextBox2.Text) Then
            MsgBox "Usuario y clave ya existen"
            limpia
            Me.MaskedTextBox1.SetFocus
        End If
        tabla.MoveNext
    Loop
tabla.Close

If MaskedTextBox2.Text <> "" And MaskedTextBox2.Text <> "" And MaskedTextBox4.Text
<> "" And MaskedTextBox6.Text <> "" Then
res = MsgBox("Confirma Realizar Insercion de Nuevo Usuario", vbYesNoCancel,
"Creacion de usuario")
    If res = 6 Then
        llena
    Else
        If res = 7 Then
            MsgBox "No se relizo el ingreso"
        End If
    End If
Else
MsgBox "Por Favor Llene los Campos Indispensables (*)", vbInformation, "Ingreso
de Usuarios"
End If
End Sub

```

```
Private Sub Command2_Click()
```

```
Call cerrar_con
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
conexion
```

```
Me.MaskedTextBox1.Text = ""
```

```
End Sub
```

```
Private Sub limpia()
```

```
MaskedTextBox1.Text = ""
```

```
MaskedTextBox2.Text = ""
```

```
MaskedTextBox3.Text = ""
```

```
MaskedTextBox4.Text = ""
```

```
MaskedTextBox5.Text = ""
```

```
MaskedTextBox6.Text = ""
```

```
End Sub
```

```
Public Sub llena()
```

```
Dim tabla As ADODB.Recordset
```

```
SQL = "select * from user"
```

```
Set tabla = New Recordset
```

```
tabla.Open SQL, con, 3, 3
```

```
tabla.AddNew
```

```
tabla!usu_nombre = MaskedTextBox1.Text
```

```
tabla!usu_apellido = MaskedTextBox2.Text
```

```
tabla!usu_direccion = MaskedTextBox3.Text
```

```
tabla!usu_celular = MaskedTextBox5.Text
```

```
tabla!usu_usuario = MaskedTextBox4.Text
```

```
tabla!usu_clave = MaskedTextBox6.Text
```

```
tabla.Update
```

```
tabla.Close
```

```
limpia
```

```
MsgBox "El registro a sido guardado satisfactoriamente", vbInformation,  
"Ingreso de Clientes"
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Call cerrar_con
```

```
End Sub
```

```
Private Sub MaskedTextBox1_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Me.MaskedTextBox2.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub MaskedTextBox2_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Me.MaskedTextBox3.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub MaskedTextBox3_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
Me.MaskedTextBox5.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub MaskedTextBox5_KeyPress(KeyAscii As Integer)
```

```
If Not (KeyAscii >= 48 And KeyAscii <= 57 Or KeyAscii = 8 Or KeyAscii = 13)
```

```
Then
```

```
KeyAscii = 0
```

```
MsgBox "Ingrese solo valores Numericos"
```

```
Me.MaskedTextBox5.SetFocus
```

```
End If
```

```
If KeyAscii = 13 Then
```

```
If Len(MaskedTextBox5.Text) = 11 Then
```

```
Me.MaskedTextBox4.SetFocus
Else
MsgBox "Este campo solicita 11 Digitos"
Me.MaskedTextBox5.SetFocus
End If
End If
End Sub
```

```
Private Sub MaskedTextBox4_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Me.MaskedTextBox6.SetFocus
End If
End Sub
```

```
Private Sub MaskedTextBox6_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Me.Command1.SetFocus
End If
End Sub
```

```
Private Sub Form_Load()
conexion
Option1.Enabled = True
End Sub
```

```
Private Sub limpia()
MaskedTextBox1.Text = ""
MaskedTextBox2.Text = ""
MaskedTextBox3.Text = ""
MaskedTextBox4.Text = ""
MaskedTextBox5.Text = ""
MaskedTextBox6.Text = ""
Combo1.Text = ""
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Call cerrar_con
```

```
End Sub
```

```
Private Sub Option1_Click()
```

```
MaskedTextBox5.Enabled = False
```

```
Combo1.Visible = True
```

```
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
On Error Resume Next
```

```
Select Case Button.Key
```

```
Case "Buscar"
```

```
MaskedTextBox5.Enabled = False
```

```
Frame1.Visible = False
```

```
Combo1.Visible = False
```

```
Call llena
```

```
Case "Eliminar"
```

```
Call elimina
```

```
limpia
```

```
Frame1.Visible = True
```

```
Combo1.Visible = True
```

```
Case "Actualizar"
```

```
If MaskedTextBox2.Text <> "" And MaskedTextBox4.Text <> "" And  
MaskedTextBox6.Text <> "" Then
```

```
res = MsgBox("Confirma Realizar Actualizacion", vbYesNo,  
"Actualizacion de usuario")
```

```
If (res = 6) Then
```

```
actualiza
```

```
Else
```

```
MsgBox "No se relizo el ingreso"
```

```
End If
```

```

Else
    MsgBox "Por Favor Llene los Campos Indispensables (*)", vbInformation,
"Actualizacion de Usuarios"
End If
limpia
Frame1.Visible = True
Combo1.Visible = True
Case "Salir"
    Unload Me
End Select
End Sub

```

```

Private Sub llena()
Dim tabla As ADODB.Recordset
Set tabla = New ADODB.Recordset
SQL = "select * from user"
tabla.Open SQL, con, 3, 3
Do Until tabla.EOF
    If Combo1.Text = tabla!usu_apellido Then
        Text7.Text = tabla!id_cliente
        MaskedTextBox1.Text = tabla!usu_nombre
        MaskedTextBox2.Text = tabla!usu_apellido
        MaskedTextBox3.Text = tabla!usu_direccion
        MaskedTextBox4.Text = tabla!usu_usuario
        MaskedTextBox5.Text = tabla!usu_celular
        MaskedTextBox6.Text = tabla!usu_clave
    End If
tabla.MoveNext
Loop
tabla.Close
If MaskedTextBox1.Text = "" Then
MsgBox "Registro no Existe", vbInformation
'MaskedTextBox2.Visible = False
Combo1.Visible = True

```



```
limpia
Frame1.Visible = True
End If
End Sub
```

```
Public Sub elimina()
Dim tabla As ADODB.Recordset
Set tabla = New ADODB.Recordset
SQL = "select * from user"
tabla.Open SQL, con, 3, 3
Do Until tabla.EOF
If tabla!id_cliente = CInt(Text7.Text) Then
    tabla.Delete
    MsgBox "Su registro a sido eliminado", vbInformation, "En su tabla de Usuarios"
End If
    tabla.MoveNext
Loop
tabla.Update
tabla.Close
End Sub
```

```
Public Sub actualiza()
Dim tabla As ADODB.Recordset
SQL = "select * from user"
Set tabla = New Recordset
tabla.Open SQL, con, 3, 3
Do Until tabla.EOF
If ((tabla!id_cliente) = CInt(Text7.Text)) Then
    SQLI = "replace into user(id_cliente, usu_nombre , usu_apellido, usu_direccion,
usu_celular, usu_usuario, usu_clave) values('" & CInt(Text7.Text) & "','" &
MaskedTextBox1.Text & "','" & MaskedTextBox2.Text & "','" & MaskedTextBox3.Text & "','"
& MaskedTextBox5.Text & "','" & MaskedTextBox4.Text & "','" & MaskedTextBox6.Text &
"')"
    con.Execute SQLI
    limpia
```

```

    MsgBox "Su registro ha sido modificado correctamente", vbInformation, "En la
tabla Usuario"
End If
    tabla.MoveNext
Loop
tabla.Close
End Sub
Option Explicit

' Opciones de seguridad de clave del Registro...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
    KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
    KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

' Tipos ROOT de clave del Registro...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1          ' Cadena Unicode terminada en valor nulo
Const REG_DWORD = 4      ' Número de 32 bits

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

```

```
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA"  
(ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long,  
ByVal samDesired As Long, ByRef phkResult As Long) As Long
```

```
Private Declare Function RegQueryValueEx Lib "advapi32" Alias  
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String,  
ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef  
lpcbData As Long) As Long
```

```
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As  
Long
```

```
Private Sub cmdSysInfo_Click()  
    Call StartSysInfo  
End Sub
```

```
Private Sub cmdOK_Click()  
    Unload Me  
End Sub
```

```
Private Sub Form_Load()  
    'Me.Caption = "Acerca de " & App.Title  
    ' lblVersion.Caption = "Versión " & App.Major & "." & App.Minor & "." &  
App.Revision  
    ' lblTitle.Caption = App.Title  
End Sub
```

```
Public Sub StartSysInfo()  
    On Error GoTo SysInfoErr  
  
    Dim rc As Long  
    Dim SysInfoPath As String
```

```
    ' Intentar obtener ruta de acceso y nombre del programa de Info. del sistema a  
partir del Registro...
```

```

If      GetKeyValue(HKEY_LOCAL_MACHINE,      gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
    ' Intentar obtener sólo ruta del programa de Info. del sistema a partir del Registro...
    ElseIf  GetKeyValue(HKEY_LOCAL_MACHINE,  gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
        ' Validar la existencia de versión conocida de 32 bits del archivo
        If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
            SysInfoPath = SysInfoPath & "\MSINFO32.EXE"

        ' Error: no se puede encontrar el archivo...
        Else
            GoTo SysInfoErr
        End If
    ' Error: no se puede encontrar la entrada del Registro...
    Else
        GoTo SysInfoErr
    End If

    Call Shell(SysInfoPath, vbNormalFocus)

Exit Sub
SysInfoErr:
    MsgBox "La información del sistema no está disponible en este momento",
vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef
As String, ByRef KeyVal As String) As Boolean
    Dim i As Long                ' Contador de bucle
    Dim rc As Long                ' Código de retorno
    Dim hKey As Long              ' Controlador de una clave de
Registro abierta
    Dim hDepth As Long           '

```

```

    Dim KeyValType As Long                ' Tipo de datos de una clave de
Registro
    Dim tmpVal As String                  ' Almacenamiento temporal para un
valor de clave de Registro
    Dim KeyValSize As Long               ' Tamaño de variable de clave de
Registro
    '-----
    ' Abrir clave de registro bajo KeyRoot {HKEY_LOCAL_MACHINE...}
    '-----
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Abrir
clave de Registro

    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError          ' Error de
controlador...

    tmpVal = String$(1024, 0)                ' Asignar espacio de variable
    KeyValSize = 1024                        ' Marcar tamaño de variable

Private Sub MDIForm_Unload(Cancel As Integer)
If Me.WindowState <> vbMinimized Then
    SaveSetting App.Title, "Settings", "MainLeft", Me.Left
    SaveSetting App.Title, "Settings", "MainTop", Me.Top
    SaveSetting App.Title, "Settings", "MainWidth", Me.Width
    SaveSetting App.Title, "Settings", "MainHeight", Me.Height
End If
End Sub

Private Sub mee_Click()
Dim f As New Form5
f.Width = 10155
f.Height = 6495
f.Left = (Screen.Width - f.Width) / 2
f.Top = 500
f.Show

```

End Sub

Private Sub mei_Click()

Dim f As New presental

f.Width = 6570

f.Height = 6090

f.Left = (Screen.Width - f.Width) / 2

f.Top = 500

f.Show

End Sub

Private Sub mhelp_Click()

' ShellExecute Me.hWnd, vbNullString, "C:\PROYECT\ver1.0.4\help\Ayuda.chm",
vbNullString, "C:\", SW_SHOWNORMAL

frmAbout.Show

End Sub

Private Sub mmin_Click()

Dim f As New Form2

' f.Width = 3990

' f.Height = 1110

' f.Left = (Screen.Width - f.Width) / 2

' f.Top = 500

' f.Show

Dim f As New Form3

f.Width = 5955

f.Height = 5175

f.Left = (Screen.Width - f.Width) / 2

f.Top = 500

f.Show

End Sub

Private Sub mmuser_Click()

Dim f As New Form6

f.Width = 5955

```

f.Height = 5175
f.Left = (Screen.Width - f.Width) / 2
f.Top = 500
f.Show
End Sub

```

```

Private Sub msal_Click()
Unload Form1
Unload Me
End Sub

```

```

Private Sub mscan_Click()
Form1.Width = 12585
Form1.Height = 8580
Form1.Left = (Screen.Width - Form1.Width) / 2
Form1.Top = 1000
Form1.Show
End Sub

```

Script SQL

MySQL-Front 3.2 (Build 13.6)

```

/!*!40101                                SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/!*!40101                                SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
/!*!40101                                SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/!*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/!*!40101 SET @OLD_SQL_MODE=@@SQL_MODE */;
/!*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES */;

```

```

/*!40101 SET NAMES latin1 */;
/*!40103 SET TIME_ZONE='SYSTEM' */;

# Host: localhost  Database: eloy
# -----
# Server version 5.0.45-community-nt

#
# Table structure for table user
#

CREATE TABLE `user` (
  `id_cliente` int(11) NOT NULL auto_increment,
  `usu_usuario` varchar(50) collate latin1_general_ci default NULL,
  `usu_clave` varchar(50) collate latin1_general_ci default NULL,
  `usu_nombre` varchar(50) collate latin1_general_ci default NULL,
  `usu_apellido` varchar(50) collate latin1_general_ci default NULL,
  `usu_direccion` varchar(70) collate latin1_general_ci default NULL,
  `usu_celular` varchar(11) collate latin1_general_ci default NULL,
  PRIMARY KEY (`id_cliente`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci ROW_FORMAT=DYNAMIC;

#
# Dumping data for table user
#

/*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

```



```

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

Programación PIC 16F870

Visualizador

```

*****
'* Name   : UNTITLED.BAS                               *
'* Author : [select VIEW...EDITOR OPTIONS]            *
'* Notice : Copyright (c) 2010 [select VIEW...EDITOR *
           : Options] *
'*       : All Rights Reserved                         *
'* Date   : 09/12/2010                                 *
'* Version : 1.0                                       *
'* Notes  :                                           *
'*       :                                           *
*****

```

```

INCLUDE "MODEDEFS.BAS"

define    LCD_DREG  PORTB
define    LCD_DBIT  4
define    LCD_RSREG PORTB
define    LCD_RSBIT 0
define    LCD_EREG  PORTB
define    LCD_EBIT  1

SENAL  VAR PORTC.1
CONT   VAR WORD
PULSO  VAR  PORTC.0
TX     VAR PORTC.2
RX     VAR PORTC.3
LED2   VAR PORTC.4
ARRAN  VAR BYTE
VALOR1 VAR BYTE
VALOR2 VAR BYTE
TOTAL  VAR WORD
NUM1   VAR BYTE
NUM2   VAR BYTE

```

```

NUM11  VAR BYTE
NAD VAR BYTE
SUM VAR WORD
TRISB=0
TRISC=%00000011
INICIO:
VALOR1=0
VALOR2=0
NUM1=0
NUM2=0
TOTAL=0
    LOW LED2
    HIGH SENAL
    LCDOUT $FE,1
    LCDOUT $FE,$80," TESIS DE GRADO "
    LCDOUT $FE,$C0,"SIST ELECTRONICO"
    LCDOUT $FE,$90,"CONTROL POR VOZ "
    LCDOUT $FE,$D0," UPS-GUAYAQUIL "
    PAUSE 3000

    LCDOUT $FE,1
    LCDOUT $FE,$80,"INTEGRANTES:  "
    LCDOUT $FE,$C0,"1. ELOY REA  "
    LCDOUT $FE,$90,"2. LUIS PINOS  "
    LCDOUT $FE,$D0,"          "
    PAUSE 3000

    LOW SENAL
    PAUSE 200
    HIGH SENAL
    PAUSE 200
    LOW SENAL
    LCDOUT $FE,1
OTRO:

```

```
LCDOUT $FE,$80,"SISTEM DE CONTEO"  
LCDOUT $FE,$C0,"  ACTIVADO  "  
LCDOUT $FE,$90,"SET.POINT:"  
LCDOUT $FE,$D0,"VALOR:"  
HIGH LED2  
; SEROUT TX,T2400,["1"]  
; PAUSE 100
```

INGRESO:

```
SERIN RX,T2400,VALOR1  
PAUSE 10
```

```
IF VALOR1="0" THEN NUM1=0 :GOTO VALOR  
IF VALOR1="1" THEN NUM1=1 :GOTO VALOR  
IF VALOR1="2" THEN NUM1=2 :GOTO VALOR  
IF VALOR1="3" THEN NUM1=3 :GOTO VALOR  
IF VALOR1="4" THEN NUM1=4 :GOTO VALOR  
IF VALOR1="5" THEN NUM1=5 :GOTO VALOR  
IF VALOR1="6" THEN NUM1=6 :GOTO VALOR  
IF VALOR1="7" THEN NUM1=7 :GOTO VALOR  
IF VALOR1="8" THEN NUM1=8 :GOTO VALOR  
IF VALOR1="9" THEN NUM1=9 :GOTO VALOR  
;IF VALOR="10" THEN NUM=10 :GOTO CUENTA  
GOTO INGRESO
```

VALOR:

```
LCDOUT $FE,$90,"SET.POINT:",DEC1 NUM1
```

DOS:

```
SERIN RX,T2400,VALOR2
```

```
PAUSE 10
```

```
IF VALOR2="0" THEN NUM2=0 :GOTO CUENTA
```

```
IF VALOR2="1" THEN NUM2=1 :GOTO CUENTA
IF VALOR2="2" THEN NUM2=2 :GOTO CUENTA
IF VALOR2="3" THEN NUM2=3 :GOTO CUENTA
IF VALOR2="4" THEN NUM2=4 :GOTO CUENTA
IF VALOR2="5" THEN NUM2=5 :GOTO CUENTA
IF VALOR2="6" THEN NUM2=6 :GOTO CUENTA
IF VALOR2="7" THEN NUM2=7 :GOTO CUENTA
IF VALOR2="8" THEN NUM2=8 :GOTO CUENTA
IF VALOR2="9" THEN NUM2=9 :GOTO CUENTA
GOTO DOS
```

```
.....
.....
```

```
;SE RECIBE EL DATO Y MUESTRA EN PANTALLA
```

```
CUENTA:
```

```
HIGH TX
```

```
LCDOUT $FE,$90,"SET.POINT:",DEC1 NUM1,DEC1 NUM2
```

```
PAUSE 1000
```

```
HIGH TX
```

```
SEROUT TX,T2400,["O"]
```

```
PAUSE 10
```

```
nUM1=NUM1*10
```

```
TOTAL=NUM1+NUM2
```

```
;GOTO INGRESO
```

```
INGRES:
```

```
SERIN RX,T2400,ARRAN
```

```
PAUSE 10
```

```
IF ARRAN="M" THEN GOTO CONTEO
```

```
GOTO INGRES
```

```
CONTEO:
```

```
IF PULSO=0 THEN GOTO SUMA
```

PAUSE 10
GOTO CONTEO

SUMA
IF PULSO=0 THEN GOTO SUMA
PAUSE 100
SUM= SUM+1
LCDOUT \$FE,\$D0,"VALOR:",DEC2 SUM
IF SUM= TOTAL THEN GOTO FIN
GOTO CONTEO

FIN:
HIGH TX
SEROUT TX,T2400,["F"]
PAUSE 10
LCDOUT \$FE,\$D0,"VALOR:",DEC2 SUM
PAUSE 1500
LCDOUT \$FE,1
LCDOUT \$FE,\$80," TESIS DE GRADO "
LCDOUT \$FE,\$C0,"SIST ELECTRONICO"
LCDOUT \$FE,\$90,"CONT FINALIZADO"
LCDOUT \$FE,\$D0," UPS-GUAYAQUIL "

END

Receptor – transmisor

INCLUDE "MODEDEFS.BAS"

;VARIABLES SERVO
DR VAR BYTE ;DERECHA
DL VAR BYTE ;DELANTE
CONTROL1 VAR PORTC.7
CONTROL2 VAR PORTB.3
;.....

SENAL VAR PORTC.6
RX2 VAR PORTB.7 ;PARA VISUALIZADOR
TX2 VAR PORTB.6 ;PARA VISUALIZADOR
TX VAR PORTB.4 ;PARA XBEE
RX VAR PORTB.5 ;PARA XBEE
VERDE VAR PORTC.0
AMARILLO VAR PORTC.1
ESTADO VAR BYTE
PARE VAR BYTE
ROJO VAR PORTC.2
MOTOR var PORTC.3
INI VAR PORTC.4 ;ES EL INTERRUPTOR DE ON
PULSO VAR PORTC.5 ;ENTRADA DEL SENDOR
PUL VAR WORD
PULO VAR WORD
ACEP VAR BYTE
VALOR1 VAR BYTE ;HASTA 255 MAX
VALOR2 VAR BYTE
VALOR3 VAR BYTE
TOTAL VAR WORD
X VAR BYTE
NUM1 VAR BYTE
NUM2 VAR BYTE
ACEPTO VAR BYTE
TRISB=0
DR=150
DL=150
PULSOUT CONTROL1,DR
PAUSE 10
PULSOUT CONTROL2,DL
PAUSE 10
INICIO:
NUM1=0
NUM2=0

```

LOW VERDE
HIGH TX
if (INI= 0 AND SENAL=0) THEN GOTO ENVIO ;INTERRUPTOR ON
ENCIENDE SISTEMAS
GOTO INICIO
ENVIO:
;IF INI=0 THEN GOTO ENVIO
;PAUSE 100
HIGH TX
SEROUT TX,T2400,["1"] ;DATO OK PARA PC
PAUSE 10
HIGH VERDE ; SISTEMA ON
INGRESO:
SERIN RX,T2400,VALOR1
PAUSE 100
PAUSE 10
IF VALOR1="0" THEN NUM1=0 :GOTO VALOR
IF VALOR1="1" THEN NUM1=1 :GOTO VALOR
IF VALOR1="2" THEN NUM1=2 :GOTO VALOR
IF VALOR1="3" THEN NUM1=3 :GOTO VALOR
IF VALOR1="4" THEN NUM1=4 :GOTO VALOR
IF VALOR1="5" THEN NUM1=5 :GOTO VALOR
IF VALOR1="6" THEN NUM1=6 :GOTO VALOR
IF VALOR1="7" THEN NUM1=7 :GOTO VALOR
IF VALOR1="8" THEN NUM1=8 :GOTO VALOR
IF VALOR1="9" THEN NUM1=9 :GOTO VALOR
;IF VALOR="10" THEN NUM=10 :GOTO CUENTA
GOTO INGRESO

VALOR:
SERIN RX,T2400,VALOR2
PAUSE 100
IF VALOR2="0" THEN NUM2=0 :GOTO CUENTA
IF VALOR2="1" THEN NUM2=1 :GOTO CUENTA

```

```

IF VALOR2="2" THEN NUM2=2 :GOTO CUENTA
IF VALOR2="3" THEN NUM2=3 :GOTO CUENTA
IF VALOR2="4" THEN NUM2=4 :GOTO CUENTA
IF VALOR2="5" THEN NUM2=5 :GOTO CUENTA
IF VALOR2="6" THEN NUM2=6 :GOTO CUENTA
IF VALOR2="7" THEN NUM2=7 :GOTO CUENTA
IF VALOR2="8" THEN NUM2=8 :GOTO CUENTA
IF VALOR2="9" THEN NUM2=9 :GOTO CUENTA
GOTO VALOR
CUENTA:
;ENVIO DATO AL MICRO VISUALIZADOR
HIGH TX2
HIGH TX2
HIGH TX2
SEROUT TX2,T2400,[VALOR2] ;PARA CORREGUIR UN ERROR QUE NO
LLEGA EL DATO
PAUSE 100
HIGH TX2
SEROUT TX2,T2400,[VALOR1]
PAUSE 100

HIGH TX2
SEROUT TX2,T2400,[VALOR2] ;TX2 ES EL VISUALIZADOR
PAUSE 100
;FIN DEL ENVIO
;.....
;.....
NUM1= NUM1*10
TOTAL=NUM1+NUM2
;ESPERO CONFIRMACION DE ENTREGA DEL VISUALIZADOR
INGRE:
SERIN RX2,T2400,ACEP
PAUSE 10
IF ACEP="O" THEN GOTO CUENTA_UNO

```



```

HIGH TX
GOTO INGRE
;ENVIO CONFIRMACION AL XBEE
CUENTA_UNO:
HIGH TX
;HIGH TX2
SEROUT TX,T2400,["O"]
;SEROUT TX2,T2400,["O"]
PAUSE 20
;ESPERO LA ORDEN ENCENDER MOTOR "M"
CUENTA_DOS:
SERIN RX,T2400,ACEPTO
IF ACEPTO="M" THEN GOTO CONTEO_1
PAUSE 10
GOTO CUENTA_DOS

CONTEO_1:
HIGH TX2
SEROUT TX2,T2400,["M"]
PAUSE 10
HIGH AMARILLO ;INDICA MOTOR TRABAJANDO
HIGH MOTOR ;ENCIENDE MOTOR

```

ANEXO B: MANUAL DE USUARIO

INGRESO AL SISTEMA

Al ejecutar el sistema nos aparecerá la pantalla de login la cual nos permite ingresar nuestro usuario y contraseña asignado.

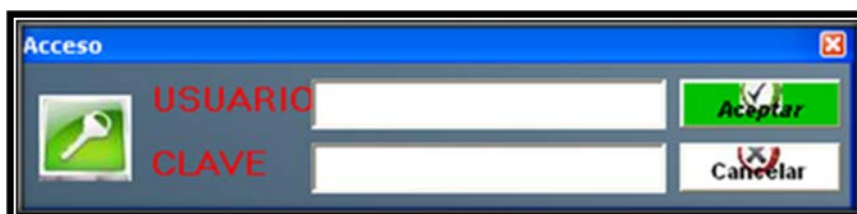


Figura B1 Pantalla de Login

Fuente: Autores

En la figura B1 Podemos ingresar con dos clases de usuario, el usuario regular y el administrador, el usuario administrador tendrá la posibilidad de ingresar al menú mantenimiento para poder ingresar, editar o eliminar un usuario del sistema, para el usuario regular la opción de mantenimiento le aparece bloqueado.

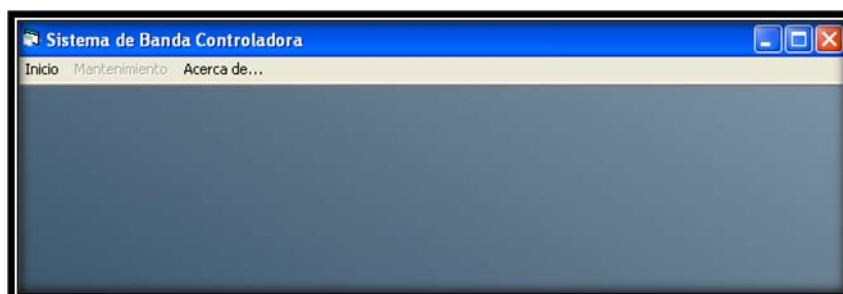


Figura B2 Menú con usuario regular

Fuente: Autores

En la figura B2 se muestra el menú con un usuario regular donde se verifica que está bloqueado el menú mantenimiento.



Figura B3 Menú usuario Mantenimiento

Fuente: Autores

La figura B3 nos muestra el menú con el usuario Administrador donde se aprecia que el menú Mantenimiento se encuentra activo.

MENÚ MANTENIMIENTO

Datos de Usuario

En el menú mantenimiento podremos verificar los datos del usuario y poder actualizar o eliminar dicho usuario



Figura B4 Menú mantenimiento Datos de Usuario

Fuente: autores

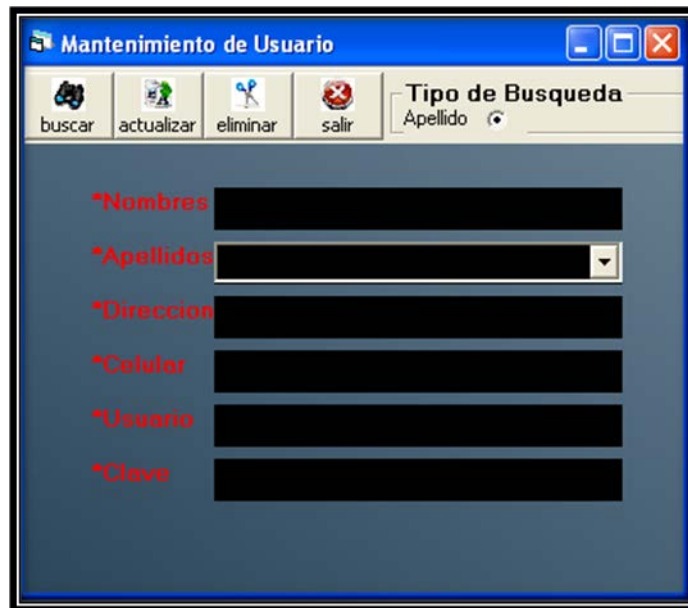


Figura B5 Mantenimiento de Usuario

Fuente: Autores

En el menú de mantenimiento de usuario (Figura B5) vemos los botones de Buscar, Actualizar, Eliminar y Salir, así como un combo de tipo de búsqueda el cual lo filtramos por apellidos de los empleados.

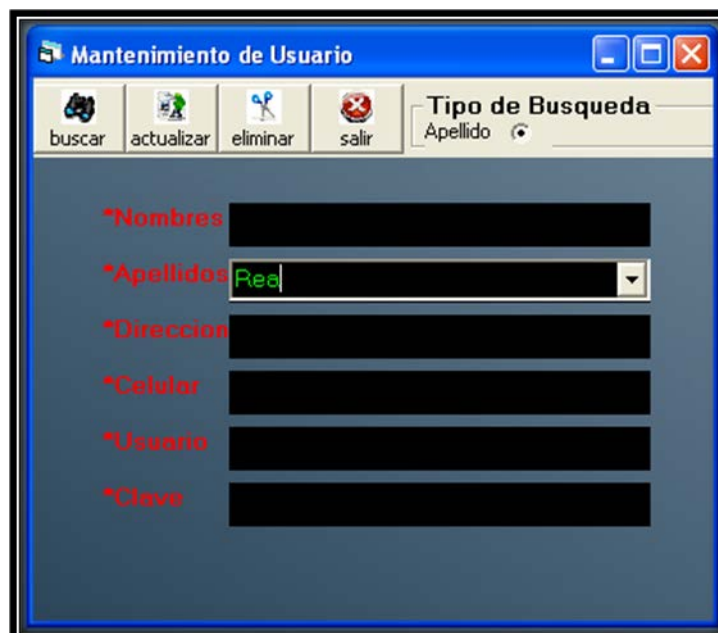
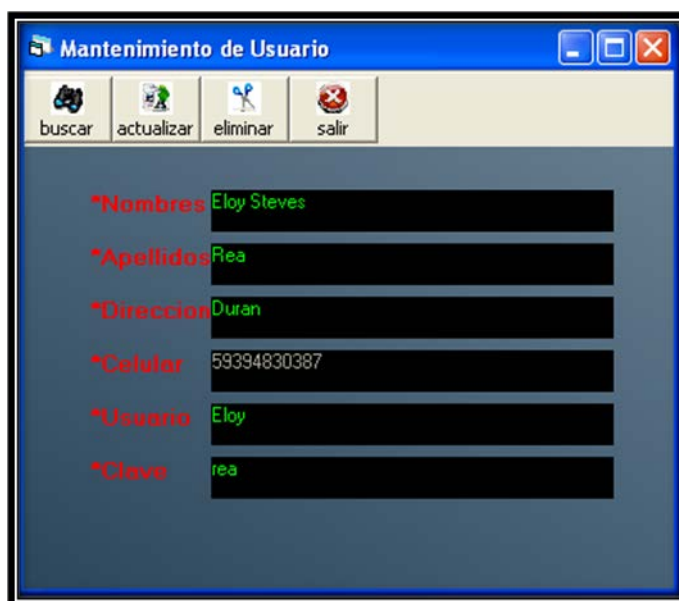


Figura B6 Búsqueda de empleado

Fuente: Autores

Para poder realizar cualquier operación en la casilla empleado, primero hay que buscarlo y para poder hacer la búsqueda en donde dice apellido dictamos el apellido del usuario y luego decimos buscar para que aparezcan los datos en las casillas correspondientes, también ahí nos aparece el usuario y la contraseña asignada al empleado, es una opción útil en caso de que el empleado no recuerde cual es su usuario o cual fue su contraseña que fue registrada (Figura B6).



The screenshot shows a window titled "Mantenimiento de Usuario" with a toolbar containing buttons for "buscar", "actualizar", "eliminar", and "salir". Below the toolbar, there are six input fields with labels in red and values in green: "Nombres" (Eloy Steves), "Apellidos" (Rea), "Direccion" (Duran), "Celular" (59394830387), "Usuario" (Eloy), and "Clave" (rea).

Figura B7 Datos empleado

Fuente: Autores

Cuando han aparecido los datos completos del empleado podremos usar los demás botones y en cada acción realizada nos aparecerá el mensaje correspondiente (Figura B7).



Figura B8 Mensaje Eliminación

Fuente: Autores

Cuando presionamos el botón eliminar nos aparece el mensaje indicando que el registro ha sido eliminado. Si se verifica en la opción de búsqueda ya no aparece el empleado que fue eliminado con anterioridad (Figura B8).

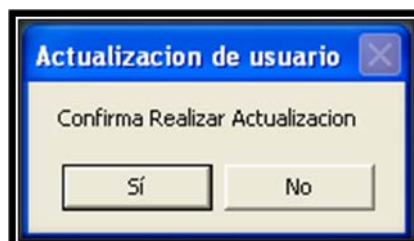


Figura B9 Confirmación Actualización

Fuente: Autores

Una vez que han aparecido los datos de los empleados después de la búsqueda, podremos modificar cualquier casilla de las que aparecen en la ventana, si queremos grabar los cambios realizados dictamos Actualizar y nos aparecerá un mensaje de que si queremos confirmar la modificación realizada (Figura B9).



Figura B10 Actualización empleado

Fuente: Autores

Si estamos conforme con la actualización dictamos si y los campos que fueron modificados se actualizaran en la base de datos del sistema, también aparecerá un mensaje indicándonos que el registro se ha modificado correctamente (Figura B10).

Ingreso de Usuario

Para ingresar un nuevo empleado al sistema nos dirigimos a la opción de mantenimiento y luego a Ingreso de usuario donde se nos desplegara una nueva ventana donde llenaremos los datos del nuevo usuario.



Figura B11 Ingreso de Usuario

Fuente: Autores

A screenshot of a software application window titled "Ingreso de Usuarios". The window contains a form with several input fields. The labels and their corresponding values are: "Nombres" (Marcelo), "*Apellidos" (Mendez), "Direccion" (Ciudadela IESS), "*Celular" (0938432345), "*User" (Marcelo), and "*Clave" (23821). At the bottom of the form, there are two buttons: "Guardar" (represented by a floppy disk icon) and "Salir" (represented by a red 'X' icon).

Figura B12 Ventana de Registro

Fuente: Autores

La casilla de Ingreso de usuario nos muestra los datos de los empleados a llenar, los cuales son los datos básicos como Nombres, Apellidos, Dirección, Celular, así también en esa misma ventana se le asigna el usuario y contraseña del empleado agregándolos en las casillas User y Clave (Figuras B11 – B12).

Debajo de esas ventanas encontramos los botones guardar y salir.

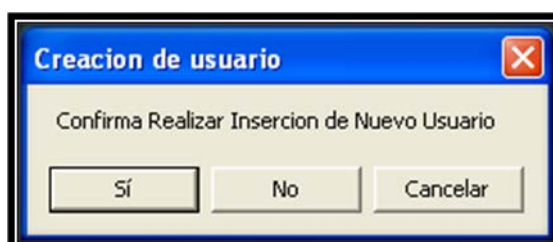


Figura B13 Confirmación ingreso nuevo usuario

Fuente: Autores

Una vez que se ha llenado todos los campos del registro del nuevo usuario se debe de dictar la palabra guardar para que el sistema ingrese la nueva información a la base de datos mostrándonos el mensaje de advertencia sobre si deseamos realizar la nueva inserción o no (Figura B13).

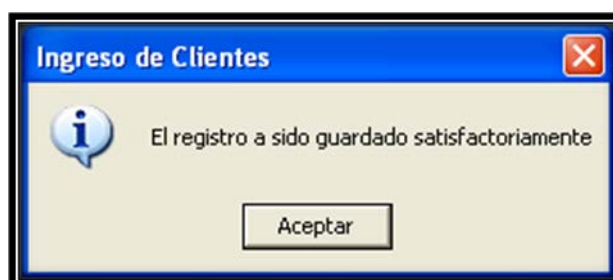


Figura B14 Registro guardado

Fuente: Autores

Si dictamos si en el cuadro anterior los datos ya se quedan registrados en nuestra base de datos y nos aparecerá un mensaje de confirmación indicándonos que el registro fue guardado satisfactoriamente, si se equivocaron en una información y desean rectificarla tendrán que ir a la opción de datos de usuario para realizar la modificación respectiva (Figura B14).

MENÚ INICIO

Dentro del menú inicio de la pantalla inicial del sistema tenemos la opción de control de la máquina transportadora.

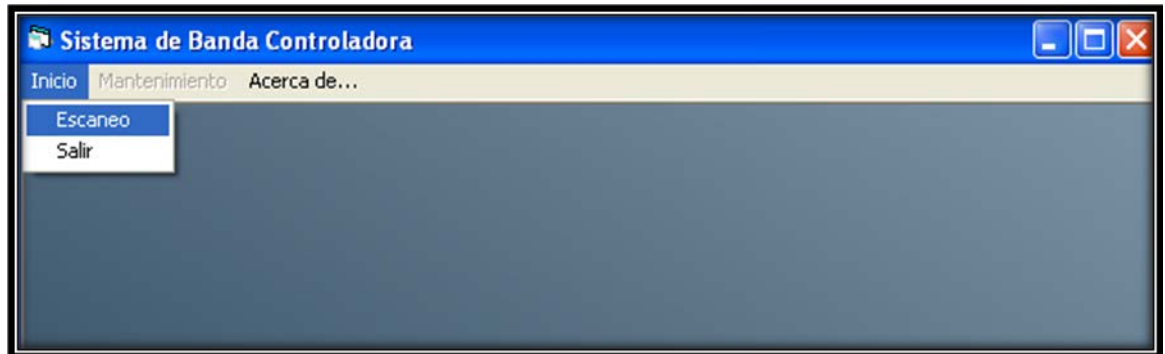


Figura B15 Menú principal inicio

Fuente: Autores

En el menú inicio (Figura B15) encontramos el submenú que contiene las opciones Escaneo y salir, si dictamos escaneo arrancamos el proceso de control de la máquina de banda transportadora.



Figura B16 Verificación Com1

Fuente: Autores

Antes de iniciar el sistema se verifica que haya conexión con el puerto de comunicación COM1 ya que ese es el puerto donde va a enviar y recibir la información el sistema (Figura B16).



Figura B17 Ingresos de valores numéricos

Fuente: Autores

Apenas iniciado el sistema nos aparece el mensaje para que coloquemos el valor numérico que se enviara a la máquina transportadora para saber cuántos elementos va a censar, una vez dado el valor se procede a habilitar el sistema completamente (Figura B17).

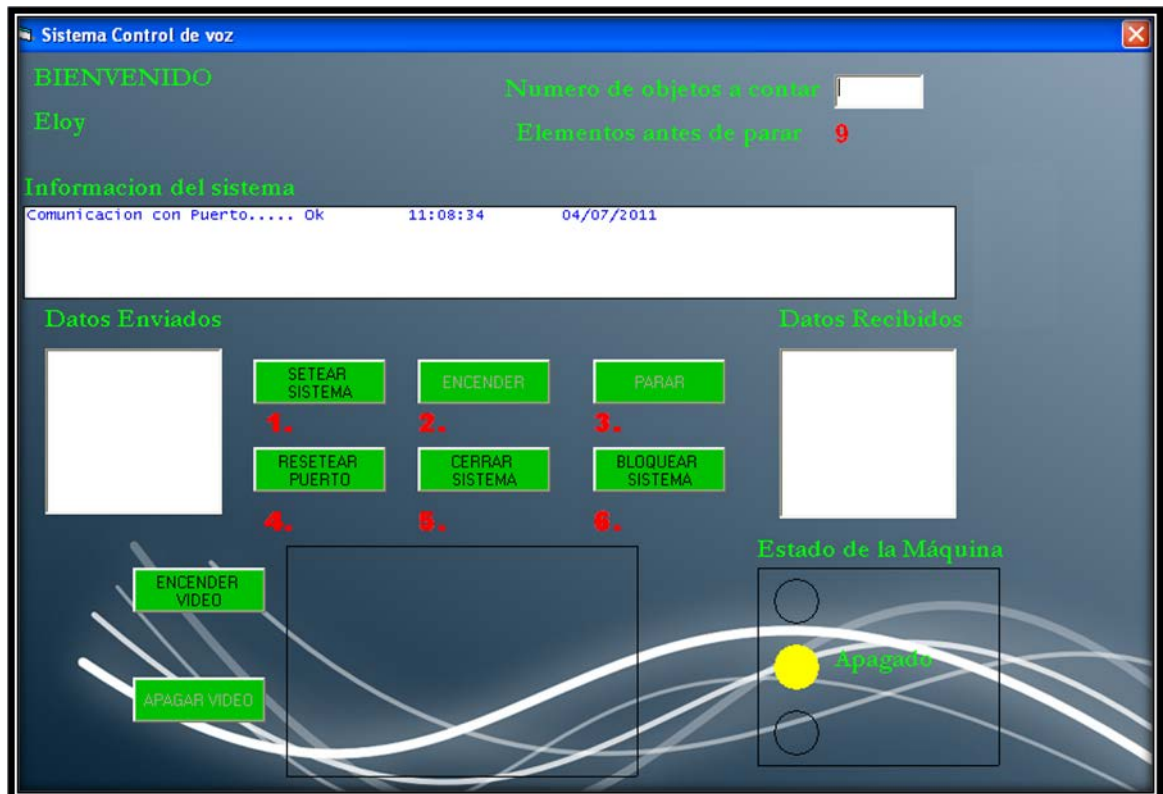


Figura B18 Pantalla de control

Fuente: Autores

La figura B18 nos presenta la pantalla de control de la máquina transportadora, en esa pantalla podremos dar por medio de comandos de voz las órdenes que se desean hacer como encender motor, apagar, bloquear, resetear, encender video, también nos

muestra pantallas informativas donde vemos el estado del puerto, los datos recibidos y enviados, los valores que se van a contar.

A continuación detallamos la función de cada uno de los componentes de esta pantalla:

Números de objetos a contar: en esa casilla debemos de dictar en número de elementos que se desea censar en la máquina transportadora.

Elementos antes de parar: nos muestra el avance de elementos contados por el sistema, en caso de error el sistema almacena ese elemento para que una vez solucionado no tenga que empezar a contar desde el principio sino desde donde se quedó antes del error.

Información del sistema: Va detallando los procesos que se está realizando en el momento de realizar las acciones en el sistema.

Datos enviados: nos muestra que datos están siendo enviado a través del puerto COM a la máquina transportadora.

Datos recibidos: Nos muestra los datos que están siendo recibidos a través del puerto COM por la máquina transportadora.

Setear sistema: Es el botón de inicio del sistema, al seleccionarlo envía un dato de verificación al sistema para empezar con el encendido del motor.

Encender: Enciende el motor de la máquina transportadora.

Parar: Apaga el motor de la máquina transportadora.

Resetear puerto: Resetea el puerto COM como precaución en caso de que el puerto se llene de datos erróneos y envíe comandos basura al sistema.

Cerrar sistema: Una vez concluido el proceso correctamente se debe de usar este comando para poder cerrar el sistema y liberar el puerto de comunicación COM1.

Bloquear sistema: En caso del que el usuario decida salir por un momento puede usar este botón para bloquear el sistema y solo puede ser reactivado cuando el usuario escriba la contraseña correspondiente.

Encender video: Muestra en la pantalla la cámara web para poder visualizar el funcionamiento de la máquina transportadora.

Apagar video: Desconecta la cámara web.

Estado de la máquina: Esta opción le indica al operador como está la situación de la máquina transportadora en ese momento, presenta los estados de encendido, apagado y error.

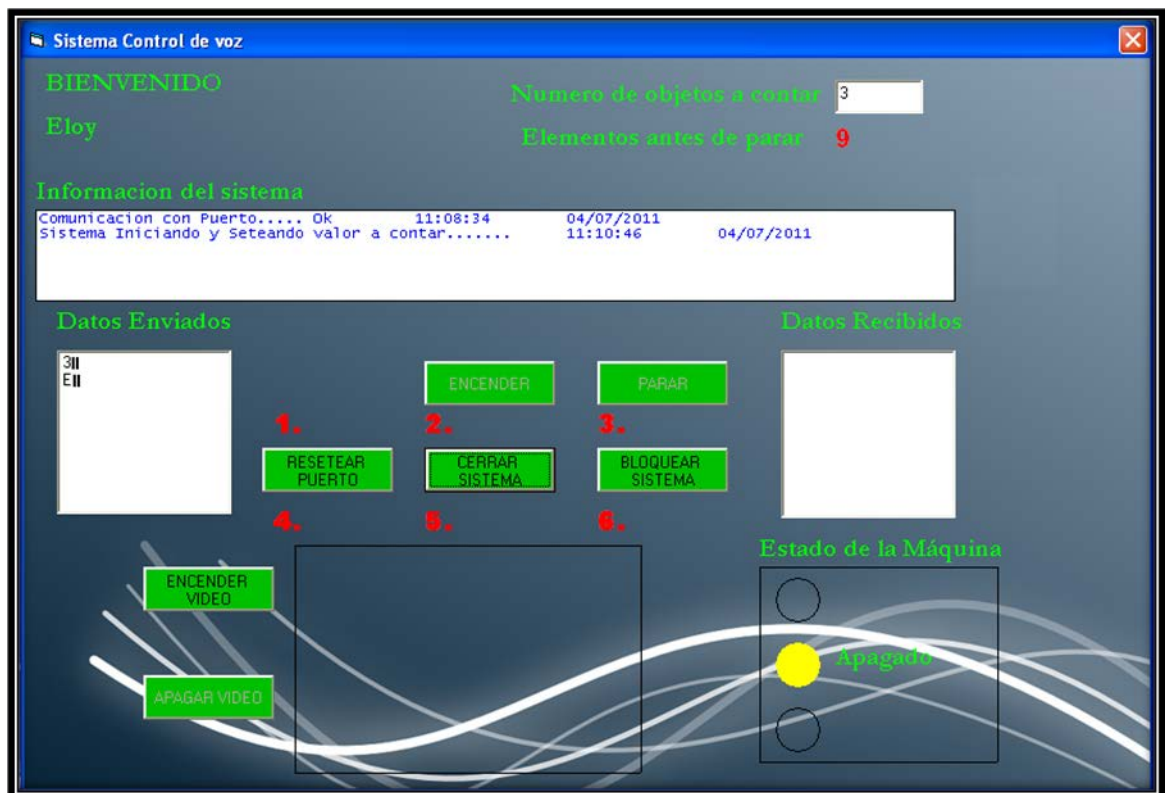


Figura B19 Funcionamiento del sistema

Fuente: Autores

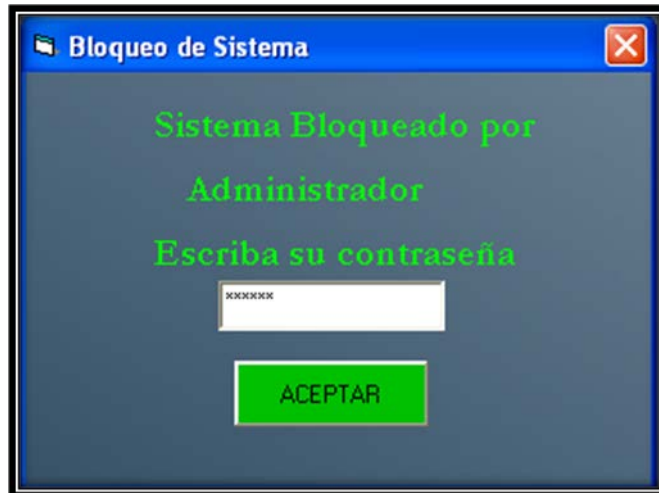


Figura B20 Bloqueo del sistema
Fuente: Autor

Cuando se dicta el comando de bloqueo la pantalla se cierra y aparece esta ventana que nos indica que el sistema está bloqueado, para volver a la pantalla inicial se debe de escribir la contraseña que usaron para registrarse y dictar aceptar (Figura B20).



Figura B21 Estado encendido
Fuente: Autores



Figura B22 Estado Error
Fuente: autores



Figura B23 Estado Apagado
Fuente: Autores

Las figuras B21, B22, B23 muestra las diferentes circunstancias que detecta la parte de estado de máquina transportadora, las cuales son Apagado cuando el motor no se ha encendido, Encendido cuando el motor ha iniciado la marcha o error cuando hay un error en el motor o en el sensor.



Figura B24 Cerrar sistema
Fuente: Autores

Una vez que ha concluido satisfactoriamente el proceso sin presentar error, diciendo cerrar sistema hacemos que se cierre el puerto de comunicación COM1 y a lo que decimos aceptar se cierra la ventana de control y vuelve al menú principal (Figura B24).

Menú acerca de...

Es una pantalla informativa con el nombre del proyecto, la versión y datos de los programadores, con el botón aceptar vuelve al menú principal (Figura B25).



Figura B25 Menú Acerca de
Fuente: Autores

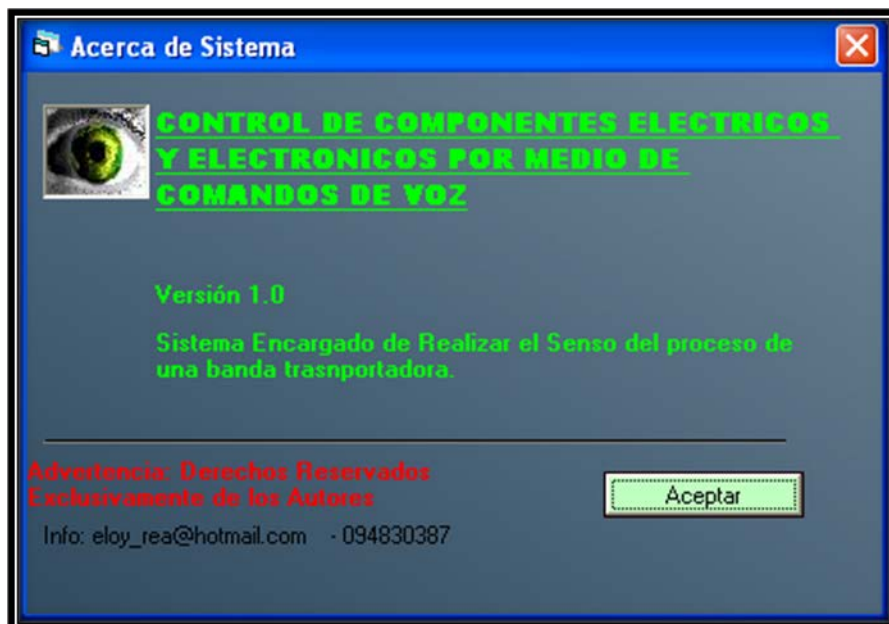


Figura B26 Acerca de.
Fuente: Autores

Ventanas de errores

El sistema cuenta con ventanas de errores que nos hacen saber cuando algo no está bien registrado en el sistema, a continuación veamos los tipos de errores presentados.



Figura B27 Error de Clave

Fuente: Autores

Al momento de ingresar el usuario y contraseña si dictamos una contraseña que no esté almacenada en la base de datos nos mostrará el mensaje de error de clave, para solucionar este error se debe de dictar la clave correcta (Figura B27).



Figura B28 Registro no existe

Fuente: Autores

En el menú mantenimiento si ponemos el apellido de un empleado y este no se encuentra registrado en la base de datos arrojará el mensaje de que el registro no existe, se debe de dictar un apellido que se encuentre registrado en el sistema (Figura B28).



Figura B29 Llenar campos

Fuente: Autores

Cuando se ingresa un nuevo usuario hay que llenar todos los campos que aparecen en la ventana, caso contrario si se pone grabar y no se han llenado los campos requeridos aparece el mensaje de error (Figura B29).



Figura B30 Error COM1

Fuente: Autores

Si el puerto COM1 al momento de arrancar el sistema no se encuentra activo o no está habilitado, presenta el mensaje de error, ya que el sistema necesita usar el puerto com1 para enviar y recibir los datos (Figura B30).



Figura B31 ingrese valor

Fuente: Autores

Para poder usar las funciones del sistema se debe de ingresar el valor a contar, el sistema no permitirá manipularlo sin antes haber ingresado un número (Figura B31).



Figura B32 Sistema Bloqueado

Fuente: Autores

Cuando el sistema está bloqueado la única forma de volver a la pantalla principal es diciendo la contraseña del usuario que se haya registrado con anterioridad, caso contrario mostrará el mensaje de que la contraseña ingresada no es correcta (Figura B32).

ANEXO C: GLOSARIO

PIC: El nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller (Controlador de Interfaz Periférico).

RS-232 : Es una interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (Equipo de Comunicación de datos), aunque existen otras situaciones en las que también se utiliza la interfaz RS-232.

SQL: (structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

CPU: También denominada como ordenador o computador, es una máquina electrónica que recibe y procesa datos para convertirlos en información útil. Una computadora es una colección de circuitos integrados y otros componentes relacionados que puede ejecutar con exactitud, sorprendente rapidez, y de acuerdo a lo indicado por un usuario o automáticamente por otro programa, una múltiple variedad de secuencias o rutinas de instrucciones que son ordenadas, organizadas y sistematizadas en función a una amplia gama de aplicaciones prácticas y precisamente determinadas.

BASE DE DATOS: es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

SENSOR: Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, etc.

COMUNICACIÓN INALÁMBRICA: La comunicación inalámbrica o sin cables es aquella en la que extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. En este sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales encontramos: antenas, computadoras portátiles, PDA, teléfonos móviles, etc.

TRANSMISOR: Transmisor en el área de comunicaciones es el origen de una sesión de comunicación. Un transmisor es un equipo que emite una señal, código o mensaje a través de un medio. Para lograr una sesión de comunicación se requiere: un transmisor, un medio y un receptor.

RECEPTOR: Un receptor es una persona o un equipo que recibe una señal, código o mensaje emitido por un transmisor o emisor.

SISTEMA OPERATIVO: Es el programa o conjunto de programas que efectúan la gestión de los procesos básicos de un sistema informático, y permite la normal ejecución del resto de las operaciones.

VSM (Virtual System Modelling): es el módulo de simulación de proteus.

CIRCUITO IMPRESO o PCB (del inglés *printed circuit board*), es un medio para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de *rutas o pistas* de material conductor, grabados en hojas de cobre laminadas sobre un sustrato no conductor, comúnmente baquelita o fibra de vidrio.

INTERFAZ: Conexión física y funcional entre dos aparatos o sistemas independientes.

MEGAPIXEL: Un **megapixel** (Mpx) equivale a 1 millón de píxeles, a diferencia de otras medidas usadas en la computación en donde se suele utilizar la base de 1024 para los prefijos, en lugar de 1000, debido a su conveniencia respecto del uso del sistema binario.

USB (Universal Serial Bus): En ordenadores, un bus es un subsistema que transfiere datos o electricidad entre componentes del ordenador dentro de un ordenador o entre ordenadores. Un bus puede conectar varios periféricos utilizando el mismo conjunto de cables.

IBM: El IBM Personal Computer, conocido comúnmente como el IBM PC, es la versión original y el progenitor de la plataforma de hardware compatible IBM PC. Es el IBM modelo 5150, y fue introducido el 12 de agosto de 1981. Fue creado por un equipo de ingenieros y de diseñadores bajo la dirección de Don Estridge del IBM Entry Systems Division.

EPROM: son las siglas de *Erasable Programmable Read-Only Memory* (ROM programable borrable). Es un tipo de chip de memoria ROM no volátil inventado por el ingeniero Dov Frohman. Está formada por celdas de FAMOS (Floating Gate Avalanche-Injection Metal-Oxide Semiconductor) o "transistores de puerta flotante", cada uno de los cuales viene de fábrica sin carga, por lo que son leídos como 1 (por eso, una EPROM sin grabar se lee como FF en todas sus celdas). Se programan mediante un dispositivo electrónico que proporciona voltajes superiores a los normalmente utilizados en los circuitos electrónicos. Las celdas que reciben carga se leen entonces como un 0.

Hyperterminal: Es un programa que se puede utilizar para conectar con otros equipos, sitios Telnet, sistemas de boletines electrónicos (BBS), servicios en línea y equipos host, mediante un módem, un cable de módem nulo o Ethernet.

Red Mesh: es una red en malla implementada sobre una red inalámbrica LAN (Red de área local).

ISO: Organización Internacional de Normalización

ANSI: Instituto Nacional Estadounidense de Estándares

WYSIWYG es el acrónimo de *What You See Is What You Get* (en inglés, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de

texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

Secure Sockets Layer (SSL; protocolo de capa de conexión segura).

ICSP: (In-Circuit Serial Programming) es una técnica de programación en circuito (ISP, In-System Programming) implementada en los microcontroladores de la marca Microchip que nos permite programar los microcontroladores en la propia placa de aplicación mediante una señal síncrona de datos serie y una señal de reloj.

BIBLIOGRAFIA

http://es.wikipedia.org/wiki/Visual_Basic

http://es.wikipedia.org/wiki/Puerto_serie

http://es.wikipedia.org/wiki/Microcontrolador_PIC

<http://es.wikipedia.org/wiki/SQL>

http://www.unicrom.com/Tut_Relé.asp

<http://www.vidadigitalradio.com/sensores-movimiento/>

<http://www.imagesco.com/articles/hm2007/SpeechRecognitionTutorial01.html>

<http://www.nuance.com/naturallyspeaking/products/editions/default.asp>

http://www.domodesk.com/product/14/14/15/1/MANDO_Activado_por_VOZ.htm

<http://www.decelectronics.com/html/XBEE/XBEE.htm>

<http://marload.blogspot.com/2010/01/módulo-rf-xbee-pro-900.html>

<http://www.alldatasheet.com>

<http://www.uv.es/marinjl/electro/reles.html>

<http://www.nichese.com/rele.html>

<http://mandobots.jimdo.com/tips-y-links/como-funciona-un-rele-de-5-pines/>

<http://www.bricolajecasero.com/electricidad/la-funcion-del-rele.php>

<http://www.taringa.net/posts/info/5800372/funcionamiento-del-rele.html>

<http://www.alegsa.com.ar/Dic/webcam.php>

<http://www.masadelante.com/faqs/webcam>

<http://www.pergaminovirtual.com/definicion/Webcam.html>

<http://www.definicion.org/webcam>

http://www.gepowercontrols.com/es/product_portfolio/control_automation/overloads/Electronic_Overload_Relay.html

http://es.wikipedia.org/wiki/Rel%C3%A9_T%C3%A9rmico

http://www.netcom.es/pepeocu/protecciones/6_2%20Reles%20termicos.htm

<http://garaje.ya.com/migotera/rele.htm>

<http://www.bricogeek.com/shop/178-pantalla-lcd-20x4-caracteres-negro-verde.html>

<http://www.buenastareas.com/ensayos/Uso-Del-Módulo-Lcd/1721169.html>

<http://www.forosdeelectronica.com/f24/control-display-lcd-microcontrolador-pic-201/>

<http://es.wikipedia.org/wiki/Led>

http://www.unicrom.com/Tut_diodo_led.asp

[http://www.radiovalvular.com/Diodo%20Emisor%20de%20luz\(let\).html](http://www.radiovalvular.com/Diodo%20Emisor%20de%20luz(let).html)

Libros

REYES, Carlos, *Microcontroladores PIC Programación en BASIC*, 2da Edición, Quito – Ecuador, 2006.