

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**



FACULTAD DE INGENIERÍAS

CARRERA:
INGENIERÍA EN ELECTRÓNICA CON MENCIÓN EN SISTEMAS
INDUSTRIALES

Tesis previa a la obtención del título de:
INGENIERO EN ELECTRÓNICA CON MENCIÓN EN
SISTEMAS INDUSTRIALES

TÍTULO:
“DISEÑO DE UN PROTOTIPO DE CONTROL DE ACCESO
APLICANDO TECNOLOGÍA RFID, CON PROTOCOLO RS485 Y
ESTANDAR ETHERNET”

AUTORES:
JAVIER DAVID NÚÑEZ MATAMOROS
CRISTHIAN ELÍAS GARCÍA ACOSTA

DIRECTOR:
ING. LUIS CÓRDOVA RIVADENEIRA

Guayaquil, Septiembre del 2012

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Javier David Núñez Matamoros y Cristhian Elías García Acosta declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestro derecho de propiedad intelectual correspondiente a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Los conceptos desarrollados, análisis realizados y las conclusiones del presente trabajo, son de exclusiva responsabilidad de los autores.

Guayaquil, Enero 1 del 2013

(f) _____
Javier David Núñez Matamoros.

(f) _____
Cristhian Elías García Acosta

AGRADECIMIENTO

Este proyecto es el resultado del esfuerzo conjunto de todos los que formamos el grupo de trabajo. Por eso agradecemos a nuestro director de tesis Ing. Luis Córdova Rivadeneira, a nuestros compañeros Robert Muñoz, Fernanda Olmedo, Madyson Espín, Jéssica Vásquez, José Acosta, Gloria Vera, Jéssica Vaca, Rommel Morales, Edison Uriña, Luciano Gómez, Jorge Reyes, César Cruz, quienes a lo largo de este tiempo han puesto a prueba nuestras capacidades y conocimiento en el desarrollo de este proyecto el cual ha finalizado llenando nuestras expectativas.

A nuestros padres quienes a lo largo de toda nuestra vida han apoyado y motivado nuestra formación académica, creyeron en nosotros en todo momento y no dudaron de nuestras habilidades.

A nuestros profesores a quienes les debemos gran parte de nuestro conocimiento, gracias a su paciencia y enseñanza, y finalmente un eterno agradecimiento a esta prestigiosa universidad, la cual abrió sus puertas a jóvenes como nosotros, preparándonos para un futuro competitivo y formándonos como personas de bien.

Javier David Núñez Matamoros.

Cristhian Elías García Acosta.

DEDICATORIA

Dedicamos este trabajo a nuestros padres y familiares

Ellos han sabido guiarnos y enseñarnos lo correcto.

También dedicamos este trabajo a todas aquellas personas
Que creyeron en nosotros y siempre nos brindaron su apoyo.

ÍNDICE GENERAL

	Pág.
DECLARATORIA DE RESPONSABILIDAD	II
AGRADECIMIENTO	III
DEDICATORIA	IV
CAPÍTULO 1	
PLANTEAMIENTO DEL PROBLEMA	
1.1 Antecedentes	19
1.2 Planteamiento del problema.	20
1.3 Justificación del proyecto	20
1.4 Objetivos	21
1.4.1 Objetivo general	21
1.4.2 Objetivos específicos	21
1.5 Explicación del proyecto	21
CAPÍTULO 2	
FUNDAMENTOS TEÓRICOS	24
2.1 Microcontroladores	24
2.2 Microcontrolador PIC16F877A	25
2.3 Microcontrolador PIC16F876	25
2.4 Módulo LCD 16 X4	27
2.4.1 Descripción general	27

2.4.2	Distribución de pines	28
2.5	Lector RFID ID12	29
2.5.1	Sensor ID12	30
2.5.2	Distribución de pines	30
2.5.3	Circuito base para el módulo RFID ID-12	31
2.6	Integrado DS1307	32
2.6.1	Descripción general	32
2.6.2	Diagrama de pines	33
2.6.3	Mapa de memoria	34
2.6.4	Segmentación	35
2.6.5	I ² C (Inter Integrated Circuit)	35
2.6.6	Declaración I2CWRITE y I2CREAD.	36
2.6.7	Relación entre la comunicación I ² C y Reloj Calendario DS1307.	37
2.7	MEMORY STICK DATALOGGER	38
2.7.1	Características	39
2.7.2	Operación del Memory Stick Datalogger	39
2.7.3	Set de comandos	39
2.7.4	Diagrama de pines	43
2.7.4.1	Modo UART	43
2.7.4.2	Modo SPI	43
2.7.5	Especificaciones Técnicas	44
2.7.6	Indicaciones led del Memory Sick Datalogger	44

2.8	Integrado SN75176	45
2.8.1	Descripción general	45
2.8.2	Diagrama de pines	46
2.9	Convertidor WIZNET	47
2.9.1	Características	47
2.9.2	Especificaciones técnicas	48
2.10	Software	50

CAPÍTULO 3

	DISEÑO ELECTRÓNICO DEL SISTEMA	56
3.1	Lector ID12 (Sensor RFID)	56
3.2	Memory Stick Datalogger	58
3.2.1	Formatear Pen Drive	58
3.2.2	Conexión del Memory Stick Datalogger	60
3.3	Esquema general de proyecto	65
3.4	Diagrama de bloques del proyecto	66
3.4.1	Estructura del “Módulo de Registro de Datos”	66
3.4.1.1	Conexiones DS1307	67
3.4.1.2	Conexiones Sensor ID12	67
3.4.1.3	Conexiones Memory Stick Datalogger	68
3.4.1.4	Conexiones LCD 20X4	68
3.4.1.5	Conexiones PIC Registro de Datos	69

3.4.1.6	Conexiones Pulsante Interno	70
3.4.1.7	Conexiones SN75176	70
3.4.1.8	Conexiones circuito de potencia	70
3.4.2	Esquema General del Módulo Registro de Datos	71
3.4.3	Estructura del “Módulo Base”	72
3.4.3.1	Conexiones Convertidor Serial (RS232 – Ethernet)	72
3.4.3.2	Conexiones PIC Módulo BASE	73
3.4.3.3	Conexiones SN75176	73
3.4.4	Esquema General del Módulo Base	74
3.5	Pruebas en Protoboard	75
3.6	Diseño de placas electrónicas	77
3.6.1	Diseño esquemático. Módulo Registro de Datos.	79
3.6.2	Diseño elaborado en Eagle. Módulo Registro de Datos.	80
3.6.3	Diseños PCB de Módulo Registro de Datos.	81
3.6.4	Diseño esquemático. Módulo Base	83
3.6.5	Diseño elaborado en Eagle. Módulo Base	84
3.6.6	Diseño PCB de Módulo Base	84
3.6.7	Diseño esquemático. Módulo RFID	85
3.6.8	Diseño elaborado en Eagle. Módulo RFID	86
3.6.9	Diseños PCB de Módulo RFID	86
3.7	Funcionamiento y pruebas generales del proyecto	87

CAPÍTULO 4

PROGRAMACIÓN Y SOFTWARE DEL SISTEMA	92
4.1 Programación del “Módulo Registro de Datos”	92
4.2 Diagrama de flujo “Módulo Registro de Datos”	103
4.3 Programa del “PIC16F876 del Módulo Registro de Datos”	104
4.4 Diagrama de flujo “PIC16F876 del Módulo Registro de Datos”	108
4.5 Programa del “Módulo Base”	109
4.6 Diagrama de flujo “Módulo Base”	112
4.7 Programa del “Módulo ID12”	113
4.8 Diagrama de flujo “Módulo ID12”	114
4.9 Software (Cardget)	115

CAPÍTULO 5

PRÁCTICAS DE LABORATORIO	129
5.1 Desarrollo de los módulos para prácticas	129
5.2 Comunicación Serial Asíncrona RS232	133
5.3 Uso del Memory Stick Datalogger	145
5.4 Comunicación serial RS485 (SN75176)	152
5.5 Uso del Wiznet –WIZ110SR	161
5.6 Comunicación Inalámbrica. Módulos XBEE	166

CAPÍTULO 6

COSTO DEL PROYECTO	173
6.1 Costo del Módulo Registro de Datos.	173
6.2 Costo del Módulo Base	174
6.3 Costo del Módulo RFID	174
6.4 Costo de accesorios y otros	175
6.5 Costo total del proyecto	176
CONCLUSIONES	177
BIBLIOGRAFÍA	179

ÍNDICE DE FIGURAS

		Pág.
Figura 1	Diagrama de pines PIC16F877A	25
Figura 2	Diagrama de pines PIC16F876	26
Figura 3	Display LCD 16 X4	27
Figura 4	Código de dirección LCD 16 X4	27
Figura 5	Tarjetas RFID	29
Figura 6	Sensor ID12	30
Figura 7	Configuración de pines del Sensor ID12	30
Figura 8	Conexiones de pines del Sensor ID12	31
Figura 9	DS1307	32
Figura 10	Diagrama de pines DS1307	33
Figura 11	Mapa de memoria del RTC DS1307	34
Figura 12	Segmentación del RTC DS1307	35
Figura 13	Interconexión de dispositivos	36
Figura 14	Sintaxis de los comandos usados para controlar I ² C	37
Figura 15	Memory Stick Datalogger	38
Figura 16	Dimensiones del módulo Memory Stick Datalogger	45
Figura 17	Aspecto físico del SN75176	45
Figura 18	Diagrama de pines del SN75176	46
Figura 19	Aspecto físico del Wiznet	47
Figura 20	Conector RJ45	48
Figura 21	Conector RS232 del Wiznet	49
Figura 22	Ventana de Inicio del programa	49

Figura 23	Ventana de instalación para el software Cardget	50
Figura 24	Ventana de Setup. Software Cardget	50
Figura 25	Ventana de Contrato de licencia. Software Cardget	51
Figura 26	Ventana de Información. Software Cardget	51
Figura 27	Ventana de Información del cliente. Software Cardget	52
Figura 28	Ventana de Carpeta de destino. Software Cardget	52
Figura 29	Ventana para instalar el programa. Software Cardget	53
Figura 30	Ventana de Instalando Cardget	53
Figura 31	Ventana de Finalización Cardget	54
Figura 32	Ejecución del software Cardget	54
Figura 33	Ventana de Filetransfer. Software Cardget	55
Figura 34	Ventana de SQL Server. Software Cardget	55
Figura 35	Ventana Nueva Conexión ID12	56
Figura 36	Ventana Conectar a ID12	56
Figura 37	Ventana Propiedades Comunicación del ID12	57
Figura 38	Pruebas de funcionamiento Sensor RFID	57
Figura 39	Ventana de comunicación del Hyperterminal RFID	58
Figura 40	Ventana Formato	58
Figura 41	Ventana Iniciar Formato	59
Figura 42	Ventana de Advertencia	59
Figura 43	Pruebas de funcionamiento con el Datalogger	60
Figura 44	Ventana Nueva Conexión Datalogger	61
Figura 45	Ventana Conectar a Datalogger	61
Figura 46	Ventana Propiedades Comunicación del Datalogger	62
Figura 47	Ventana Hyperterminal Datalogger	62

Figura 48	Ventana Propiedades: Datalogger	63
Figura 49	Ventana Configuración ASCII: Datalogger	63
Figura 50	Ventana Hyperterminal Datalogger ingresando datos al pendrive	64
Figura 51	Esquema General del Proyecto	65
Figura 52	Diagrama de bloques del Módulo de Registro de Datos	66
Figura 53	Conexiones DS1307	67
Figura 54	Conexiones Sensor ID12	67
Figura 55	Conexiones Memory Stick Datalogger	68
Figura 56	Conexiones LCD 20 X4	68
Figura 57	Conexiones pic Registro de Datos (PIC16F877A)	69
Figura 58	Conexiones Pulsante Interno	70
Figura 59	Conexiones SN75176	70
Figura 60	Conexiones circuito de potencia	70
Figura 61	Esquema General del Módulo Registro de Datos	71
Figura 62	Diagrama de bloques del Módulo Base	72
Figura 63	Conexiones WIZNET	72
Figura 64	Conexiones PIC Módulo Base	73
Figura 65	Estructura SN75176	73
Figura 66	Esquema General del Módulo Base	74
Figura 67	Pruebas en Protoboard (LCD 20 X 4)	75
Figura 68	Pruebas en Protoboard (Memory Stick y Sensor ID12)	75
Figura 69	Pruebas en Protoboard (Comunicación con laptop)	76
Figura 70	Pruebas en Protoboard (Módulos Registro de Datos)	76
Figura 71	Ventana PCB del software EAGLE	78

Figura 72	Diseño Esquemático. Módulo Registro de Datos	79
Figura 73	Diseño Elaborado en Eagle. Módulo Registro de Datos	80
Figura 74	Diseño PCB Módulo Registro de Datos (1)	81
Figura 75	Diseño PCB Módulo Registro de Datos (2)	81
Figura 76	Diseño PCB Módulo Registro de Datos (3)	82
Figura 77	Diseño PCB Módulo Registro de Datos (4)	82
Figura 78	Diseño esquemático. Módulo Base	83
Figura 79	Diseño Elaborado en Eagle. Módulo Base	84
Figura 80	Diseño PCB Módulo Base	84
Figura 81	Diseño esquemático. Módulo RFID	85
Figura 82	Diseño Elaborado en Eagle. Módulo RFID	86
Figura 83	Diseño PCB Módulo RFID	86
Figura 84	Módulo Registro de Datos	87
Figura 85	Diagrama de conexiones del Módulo Registro de Datos	88
Figura 86	Módulo Base	88
Figura 87	Router	88
Figura 88	Fuente de alimentación	89
Figura 89	Pulsantes	89
Figura 90	Módulo RFID	89
Figura 91	Elementos de Control de acceso	90
Figura 92	Prototipo de Control de Acceso (1)	90
Figura 93	Prototipo de Control de Acceso (2)	91
Figura 94	Prototipo de Control de Acceso (3)	91
Figura 95	Ícono del software CardGet	115
Figura 96	Ventana de contraseña	115

Figura 97	Seleccionar Gateway	115
Figura 98	CardGet. Mensaje de error	116
Figura 99	CardGet. Menú contextuales	116
Figura 100	CardGet. Bitácora de Accesos	117
Figura 101	CardGet. Administración. Usuario	117
Figura 102	CardGet. Administración. Usuario. Nuevo Usuario	118
Figura 103	CardGet. Administración. Usuario. Reinicio de clave.	118
Figura 104	CardGet. Administración. Usuario. Lista de usuarios	119
Figura 105	CardGet. Archivo Excel generado para Lista de usuarios	119
Figura 106	CardGet. Administración. Usuario. Eliminación de usuarios	120
Figura 107	CardGet. Administración. Horario. Nuevo Horario	120
Figura 108	CardGet. Administración. Horario. Nuevo Horario	121
Figura 109	CardGet. Administración. Materia. Nueva materia	121
Figura 110	CardGet. Administración. Materia. Lista de Materia	122
Figura 111	CardGet. Administración. Materia. Eliminar Materias	122
Figura 112	CardGet. Administración. Colaborador. Puertos disponibles	123
Figura 113	CardGet. Administración. Colaborador. Nuevo colaborador	123
Figura 114	CardGet. Administración. Colaborador. Lista de Colaboradores	123
Figura 115	CardGet. Administración. Colaborador. Eliminar Colaborador	124
Figura 116	CardGet. Administración. Sincronización de Mantenimiento	124
Figura 117	CardGet. Administración. Carrera. Nueva Carrera	125
Figura 118	CardGet. Administración. Carrera. Lista Carreras ingresadas	125
Figura 119	CardGet. Administración. Carrera. Eliminar carrera	126
Figura 120	CardGet. Administración. Período. Nuevo período	126
Figura 121	CardGet. Administración. Período. Eliminar período	126

Figura 122	CardGet. Configuración. Dispositivo. Nuevo dispositivo	127
Figura 123	CardGet. Configuración. Dispositivo. Listado RFID	127
Figura 124	CardGet. Configuración. Dispositivo. Eliminar dispositivo	128
Figura 125	CardGet. Configuración. Herramientas de Diagnóstico	128
Figura 126	Diseño en PROTEUS. Tarjeta para prácticas	130
Figura 127	Diseño en PROTEUS. Tarjeta para prácticas Vista 3D	131
Figura 128	Módulo de tarjeta para prácticas	132

ÍNDICE DE TABLAS

		Pág.
Tabla 1	Elementos del modulo Registro de Datos	22
Tabla 2	Elementos del modulo Registro Base	22
Tabla 3	Diagrama de pines del LCD 16X4	28
Tabla 4	Diagrama de pines del Sensor ID12	31
Tabla 5	Diagrama de pines del DS1307	33
Tabla 6	Comandos usados para controlar el Memory Stick Datalogger	42
Tabla 7	Reporte de errores	42
Tabla 8	Distribución pines del Memory Stick Datalogger Modo UART	43
Tabla 9	Distribución pines del Memory Stick Datalogger en Modo SPI	43
Tabla 10	Especificaciones técnicas del Memory Stick Datalogger	44
Tabla 11	Estado del LED en el Memory Stick Datalogger	44
Tabla 12	Características generales del Wiznet	48
Tabla 13	Pines de conexión para RJ45	48
Tabla 14	Descripción de pines del puerto serial	49
Tabla 15	Precio de elementos del Módulo Registro de Datos	173
Tabla 16	Precio de elementos del Módulo Base	174
Tabla 17	Precio de elementos del Módulo RFID	174
Tabla 18	Precio de accesorios y otros	175
Tabla 19	Precio de Módulos de Practicas	176
Tabla 20	Costo Total del proyecto	176

AÑO	TÍTULO	ALUMNO/S	DIRECTOR DE TESIS	TEMA DE TESIS
2012	INGENIERO ELECTRÓNICO CON MENCIÓN EN SISTEMAS INDUSTRIALES	NÚÑEZ MATAMOROS Javier David GARCÍA ACOSTA Cristhian Elías	CORDOVA RIVADENEIRA Luis	“DISEÑO DE UN PROTOTIPO DE CONTROL DE ACCESO APLICANDO TECNOLOGÍA RFID, CON PROTOCOLO RS485 Y ESTANDAR ETHERNET”

ABSTRACT

La presente tesis: “DISEÑO DE UN PROTOTIPO DE CONTROL DE ACCESO APLICANDO TECNOLOGÍA RFID, CON PROTOCOLO RS485 Y ESTANDAR ETHERNET”, las partes principales que la componen son: cuatro módulos registradores de datos, un módulo base y un módulo RFID y dos módulos de prácticas. El microcontrolador PIC procesa la información de los elementos externos, el sensor RFID es el encargado de leer las tarjetas RFID, el memory stick datalogger almacena los horarios de acceso al personal, el CI. SN75176 se encarga de establecer la comunicación RS485, el DS1307 genera la fecha y hora actual, todos estos son elementos indispensables en este proyecto.

La necesidad de nuestro proyecto surge de mejorar los sistemas de control de acceso y este sistema representarlo a escala, esto lo hace usando la tecnología RFID y los módulos se encargan de dar la apertura a una área específica en base a un horario previamente establecido en cada uno de ellos.

Este proyecto se complementa con las tarjetas de prácticas, en los cuales se pueden realizar un estudio de los diferentes elementos usados en nuestra tesis y de esta manera dar al estudiante una base para generar futuros proyectos.

PALABRAS CLAVES

Prototipo / Control de acceso / tecnología / RFID / RS485 / Ethernet / Memory Stick Datalogger / Microcontrolador / DS1307 / Cardget / LCD 20X4 / ID12 / WIZNET / PENDRIVE / SN75176

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes:

Por lo general, existen ambientes que necesitan métodos para monitorizar y controlar las entradas del personal a sus trabajos con el objetivo de realizar el control de sus horas laborales, controlar bienes del establecimiento y administrar accesos para personal específico; dentro de esos métodos vemos desde los registros escritos firmados por los empleados, hasta complejos sistemas con sensores biométricos, lo que hace ver la necesidad que tienen algunos lugares de mantener un control a sus empleados.

Podemos citar el caso de la Universidad Politécnica Salesiana, el cual posee un control de acceso ubicado en diferentes lugares del establecimiento distantes de las aulas de clase y laboratorios, aquí se puede notar que al realizar el control de ingreso genera retrasos en el tiempo de inicio de clases, desde que el personal docente realiza su marcación hasta que llega al aula, además este proceso de marcación no garantiza que el personal ingresa a cumplir sus obligaciones, otro punto a recalcar es que constantemente hay que buscar al encargado de las llaves para abrir las diferentes áreas de trabajo.

Esta manera de administración del edificio no garantiza el cuidado de los bienes de los diferentes lugares de la Universidad, tampoco que las horas de clase impartidas por los docentes sean completas, he inclusive toda la responsabilidad de estos lugares puede recaer sobre el encargado de piso.

Además mencionamos que en los sistemas biométricos de control de acceso se necesita grabar la información individualmente en cada módulo, lo que implica que si existen módulos en lugares distintos, la persona debe ingresar sus datos en cada uno de ellos; otro punto a considerar es que los lectores de huella tienen que estar en contacto directo constantemente con el usuario, siendo propensos a daños del equipo.

Finalmente podemos acotar que la Universidad cuenta con un sistema de acceso, no apropiado a sus necesidades de establecimiento educativo, y que aplicando nuevas tecnologías podemos mejorar el sistema actual para obtener una mejor administración del edificio.

1.2 Planteamiento del problema.

¿Existe en el ámbito industrial de la ciudad de Guayaquil, un control de acceso autónomo, personalizado y portátil, capaz de almacenar registros de entrada sin depender de un centro único de monitorización?

1.3 Justificación del proyecto.

Este proyecto es necesario para la Universidad Politécnica Salesiana, en el cual se propone una solución a escala (prototipo) en base a los conocimientos adquiridos, además de la labor investigativa acerca de nuevas tecnologías RFID, capaz de reemplazar los sensores biométricos, por tarjetas codificadas únicas e irrepetibles, de esta manera el personal contará con una de ellas y tendrá acceso a lugares previamente especificados.

Así como también elementos electrónicos modernos capaces de administrar datos, manejar grandes cantidades de memoria, que garantizarán la confiabilidad de la información almacenada en ellos, capaces de realizar sus funciones sin depender de una red estructurada entre los mismos, he inclusive reemplazar módulos averiados sin interferir en el funcionamiento del sistema.

También permitirá disminuir el tiempo de registro del personal que tendrá acceso a diferentes áreas, ya que conociendo el código de las tarjetas RFID, se pueden almacenar en la memoria de cada uno de los módulos diseñados, e inclusive permite tener un stock de tarjetas pregrabadas en el sistema para el nuevo personal, evitando los registros individuales por cada estación, que se realiza normalmente en los lectores biométricos y de esa manera volver más eficiente la distribución de las mismas cuando se debe cubrir una gran demanda en poco tiempo.

Además de lo mencionado, para el estudio de los equipos usados en el control de acceso, se diseñaron tarjetas para prácticas, en el cual se manejan todos los tipos de comunicaciones vistos en el proyecto principal, de esta manera los estudiantes tendrán más accesos al conocimiento de equipos actuales y podrán complementar nuestra investigación.

1.4 Objetivos

1.4.1 Objetivo General

- Desarrollar un prototipo de control de acceso, utilizando los conocimientos aprendidos de microcontroladores PIC, comunicación serial, tecnologías de tarjetas RFID y el uso del estándar Ethernet para redes de computadoras.

1.4.2 Objetivos Específicos

- Diseñar y elaborar módulos (módulo registro de datos) capaces de controlar el acceso por medio de tarjetas RFID de manera independiente, tomando en cuenta un horario específico.
- Diseñar y construir un módulo capaz de servir de interfaz entre los módulos puerta y el PC, utilizando el estándar Ethernet (TCP/IP).
- Diseñar y desarrollar módulos de práctica para centralizar los temas usados en el proyecto final.

1.5 Explicación del proyecto.

El sistema que se desarrollará como un prototipo, será capaz de explicar los tipos de comunicaciones ya mencionadas:

Módulo de Registro de Datos:

Estos módulos “**MÓDULO DE REGISTRO DE DATOS**”, constará de algunos elementos entre los cuales tenemos:

	Descripción
PIC 16F877A	PIC REGISTRO. Este microcontrolador es aquel que posee toda la programación que se encarga de detectar las tarjetas RFID, así como también interactuar con el DS1307 (comunicación I2C).
PIC 16F876	PIC16F876. Este microcontrolador se encarga de darle una dirección única a cada módulo de registro para poder diferenciarlos de los otros, además de encargarse de transmitir y recibir información de la red.
LCD 20X4	Visualizador de información, es un indicador que presenta en la primera línea la materia, el nombre de la persona y en la segunda línea la fecha y hora, cabe indicar que también mostrará otros mensajes como “Acceso Denegado”, “Bienvenido”, etc. dependiendo de las condiciones.

Sensor ID12	Se encarga de leer los datos de la tarjeta RFID y los envía al microcontrolador para ser procesados.
Memory Stick Datalogger	Será el encargado de guardar los datos de fecha, hora, ubicación y el nombre de la persona (siendo estos datos ficticios), a la que se encuentra asignada la tarjeta, para alimentar el registro de información (deberá tener una capacidad de memoria suficiente para almacenar datos).
DS1307	Reloj en tiempo real es un integrado capaz de dar la fecha y hora como su nombre lo indica en tiempo real, el cual va a ser necesario para saber los tiempos de registro.
SN75176	Comunicación RS485
Circuito de potencia	Es el encargado de activar un relé para conectar cualquier carga deseada, cuyo tiempo de activación será de 10 segundos.

Tabla 1 Elementos del módulo Registro de Datos

Fuente: Autores

Módulo Base

Este módulo será capaz de relacionar los datos dentro de una red. El cual constará de los siguientes elementos:

	Descripción
PIC 16F877A	PIC BASE. Este microcontrolador es el encargado de controlar el flujo de información entre la red485 (módulos registradores de datos) y el router D - LINK (Red Ethernet)
SN75176	Comunicación RS485
W5100	Módulo WIZNET. Este convertidor con ayuda del PIC16F877A será el encargado de entablar la comunicación entre la computadora y el módulo base para poder enviar los datos referentes al control de acceso, el cual maneja el protocolo Ethernet.

Tabla2 Elementos del módulo Registro Base

Fuente: Autores

Según los elementos explicados en los párrafos anteriores:

El LCD en su segunda fila presenta la fecha y la hora en tiempo real gracias al integrado DS1307. El “**Módulo de Registro de Datos**” cuenta con un lector RFID, cuando una persona acerca su tarjeta RFID única, el lector sensara los códigos de la

misma y verificará según el programa dentro del microcontrolador las respectivas condiciones:

Si la persona está en el horario correcto, el LCD presenta en la primera fila el nombre de la persona con su respectiva materia, los datos de la fecha y hora. Todos estos datos se quedarán almacenados en un pendrive con ayuda del Memory Stick Datalogger, actúa la etapa de potencia y desbloquea la chapa magnética para permitir el acceso. Si el personal no está en el horario correcto, el LCD presentara el mensaje ACCESO DENEGADO y la etapa de potencia no actuará.

Cabe recalcar que los datos de fecha y hora se quedaran almacenados, (sólo cuando el acceso del personal es correcto), en los módulos gracias al pendrive que trabaja conjuntamente con el Memory Stick Datalogger.

El encargado de controlar los módulos de acceso tendrá todos los datos del personal y así poder revisar los datos que los módulos en cada aula le envían y que así mismo se encuentran almacenados en el pendrive y poder manipularlos en el software, por lo tanto también se podrá variar los horarios en el software del computador y grabarlos en el Memory Stick para así al colocarlos en los módulos de cada clase, este actualizará su horario, y así mismo actualizar la fecha y la hora siendo esta parte bidireccional, todo esto se realiza a través de la red Ethernet con sus respectivos dispositivos.

El funcionamiento en el prototipo es el siguiente:

Cuando una persona desea habilitar una puerta, éste pasa su tarjeta por el lector para abrir la puerta, en ese momento se marca su entrada (tomando en cuenta que está en el horario correcto).

En ese momento si cierra o no la puerta, ya no importa porque ya ingresó su hora de entrada.

Si la puerta está cerrada la persona puede salir de la habitación, ya que en la parte posterior de la puerta existirá un pulsante, para abrir la puerta sin necesidad del ingreso de una tarjeta.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS

En este capítulo se desarrolla la parte teórica de los diferentes dispositivos que se usarán en el proyecto.

2.1 Microcontroladores.

Se puede definir al microcontrolador como un dispositivo capaz de desarrollar órdenes almacenadas en su memoria. Este circuito integrado se lo puede comparar con un computador ya que posee tres bloques funcionales tales como: unidad central de procesos, memoria, periféricos de entrada y salida.

La unidad central de proceso es aquel que interpreta las instrucciones almacenadas en la memoria del programa y procesa los datos ingresados por los periféricos de entrada. Como resultado los periféricos de salida se comunican con el medio exterior para alertar, comunicar e informar al usuario.

“Microcontroladores, como ya se ha comentado previamente, un sistema completo con unas prestaciones limitadas que no pueden modificarse y que puede llevar a cabo las tareas para las que ha sido programado de forma autónoma.”¹

En el mercado existen millones de Microcontroladores de diferentes fabricantes, de allí la importancia de escoger el más acertado para su propósito dependiendo de sus características y prestaciones tales como procesamiento de datos, cantidad de periféricos de entrada y salida, consumo, cantidad y tipo de memoria, etc.

La compañía Microchip Technology Inc. se encarga de elaborar la familia de Microcontroladores denominados PIC (Controlador de Interfaz Periférico), el presente proyecto se basará en los dispositivos PIC16F877A y PIC16F876.

1 http://www.unicrom.com/Tut_PICs1.asp / Autor: Max Web Portal / Año 2012

2.2 Microcontrolador PIC16F877A

Este microcontrolador es un dispositivo principal en nuestro proyecto, ya que en él se grabará el programa que ejecutará la secuencia para lectura de tarjetas RFID, almacenamiento de datos en el memory stick, he inclusive será el que genera el algoritmo (secuencia de pasos lógicos) para cumplir lo planteado.

Entre las características del PIC tenemos:

- Memoria de programa tipo Flash 8K x 14
- Memoria de Datos 368 bytes
- Memoria EEPROM 256 bytes
- El Integrado posee 40 pines (33 pines de entrada y salida)
- Soporta Xtal 20 Mhz.
- Voltaje de operación: 2.0V hasta 5.5V
- 1 Convertidor A/D de 10 bits (8 canales)
- 2 módulos CCP (Captura, comparador, PWM)
- 1 Módulo I²C
- 1 USART (Puerto serie)
- 2 Timers de 8 bits
- 1 Timer 16 bits

A continuación en la Figura 1 se presenta el diagrama de pines del PIC, el cual resume algunas características explicadas anteriormente.

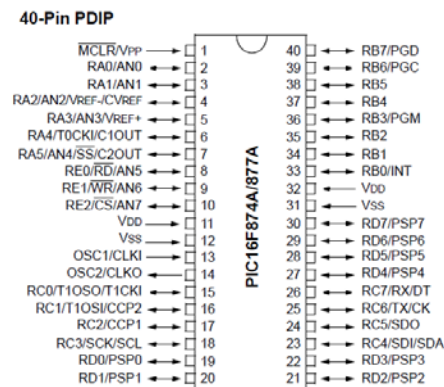


Figura 1 Diagrama de pines 16F877A / Año 2012

Fuente: <http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>

2.3 Microcontrolador PIC16F876

Este microcontrolador será el encargado de almacenar una dirección única para cada módulo registro de datos, también ayuda a la transmisión de información a través de la red RS485 que se presenta en el proyecto.

Entre las características del PIC tenemos:

- Procesador de arquitectura RISC avanzada.
- Memoria de Datos 368 bytes
- Memoria EEPROM 256 bytes
- El Integrado posee 28 pines (33 pines de entrada y salida)
- Soporta Xtal 20 Mhz.
- Voltaje de operación: 2.0V hasta 5.5V
- 1 Puerto Serie Síncrono Master (MSSP) con SPI e I²C.
- 2 módulos CCP (Captura, comparador, PWM)
- 1 Módulo I²C
- 1 USART (Puerto serie)
- 2 Timers de 8 bits
- 1 Timer 16 bits

A continuación se presenta el diagrama de pines del PIC en la Figura 2, el cual resume algunas características explicadas anteriormente

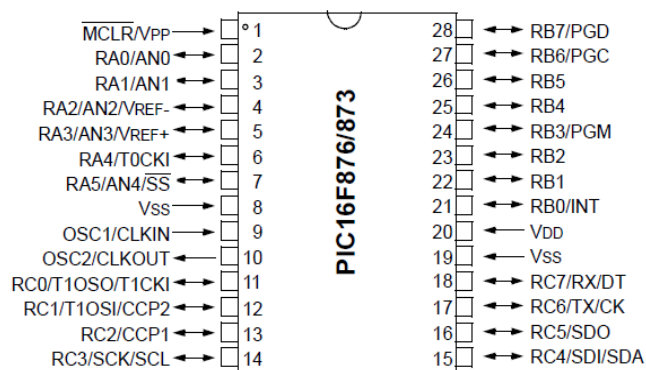


Figura 2 Diagrama de pines 16F876 / Año 2012

Fuente: <http://ww1.microchip.com/downloads/en/devicedoc/30292c.pdf>

2.4 Módulo LCD 16 X4

El módulo LCD 16 X 4 es el que presenta mensajes definidos en el programa con esto se puede observar la secuencia del programa e interactuar con el operador.

Entre los mensajes que se visualizan están: “Acceso Denegado”, “Bienvenido”, la fecha y hora, nombre de la persona quién ingresa, todo esto dependiendo de las condiciones en que esté el sistema.

2.4.1 Descripción general

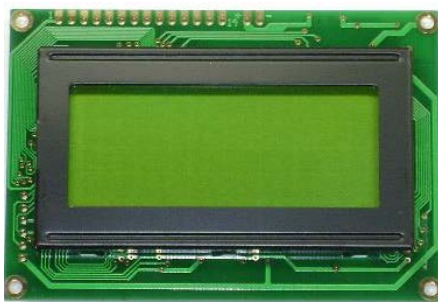


Figura 3 Display LCD 16 X4/ Año 2012
Fuente: <http://www.idtelectronics.com/products/lcd-products.html>

La pantalla de cristal líquido o LCD (LiquidCrystal Display) es un dispositivo microcontrolador de visualización gráfica para la presentación de caracteres o símbolos.

Como su nombre lo indica contiene 4 filas por 16 caracteres, en donde cada caracter tiene 5 x 8 puntos.

El módulo LCD es considerado un periférico de salida ya que presenta los resultados del proceso realizado por el PIC, como indica la figura 2.3. Cabe indicar que con ayuda de comandos del programa, se puede establecer una comunicación con 4 bits (pines 7, 8, 9,10) no necesariamente se usan los 8 bits (pin 7 al 14) y esto ayuda mucho para disminuir la cantidad de pines que se usarán en el microcontrolador.

A continuación se presenta la Figura 4 indicando el código de dirección del LCD para ubicar cualquier caracter en diferentes posiciones.

DISPLAY CHARACTER ADDRESS CODE:																
Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F
DD RAM Address	10	11														1F
DD RAM Address	50	51														5F

Figura 4 Código de dirección LCD 16 X4/ Año 2012
Fuente: <http://www.datasheetcatalog.org/datasheet/vishay/016m004b.pdf>

2.4.2 Distribución de pines

Contiene 16 pines distribuidos de la siguiente manera:

Pines 1 y 2 de alimentación el pin 3 controla el contraste con ayuda de un potenciómetro para que limite la corriente los pines 4, 5 y 6 son de comandos, también cuenta con 8 bits de datos que están representados desde los pines 7 al 14 siendo el primero el menos significativo los pines 15 y 16 indican la intensidad de luz del display. Para una mayor comprensión revisar la Tabla 3.

Pin	Nombre	Función	Descripción
1	Vss	Alimentación	GND (Negativo de la fuente)
2	Vdd	Alimentación	+5V (Positivo de la fuente)
3	Vo	Ajuste de contraste	Normalmente se conecta a un potenciómetro a través del cual se aplica una tensión variable entre 0 y 5 V que permite regular el contraste del cristal líquido. Por lo general afecta a la intensidad de los caracteres.
4	RS	Comando	Selección del registro de control/registro de datos RS = 0 Selección del registro de control RS = 1 Selección del registro de datos
5	R/W	Comando	Señal de lectura / escritura R/W = 0 El módulo LCD es escrito R/W = 1 El módulo LCD es leído
6	E	Comando	Señal de activación del módulo LCD E = 0 Módulo desconectado E = 1 Módulo conectado
7	D0	I/O	Dato LSB (Bit menos significativo)
8	D1	I/O	Dato
9	D2	I/O	Dato
10	D3	I/O	Dato
11	D4	I/O	Dato
12	D5	I/O	Dato
13	D6	I/O	Dato
14	D7	I/O	Dato MSB (Bit más significativo)
15	A	Ánodo	En estos pines se ubica una resistencia para controlar la intensidad de luz del LCD
16	K	Cátodo	

Tabla 3 Diagrama de pines del LCD 16X4/ Año 2012

Fuente: <http://www.idtelectronics.com/products/lcd-products.html>

2.5 LECTOR RFID ID12

“La identificación por radio-frecuencia o RFID es un término genérico para denominar las tecnologías que utilizan ondas de radio para identificar automáticamente personas u objetos. Existen varios métodos de identificación, pero el más común es almacenar un número de serie



Figura 5 Tarjetas RFID/ Año 2012
Fuente: <http://11870.com/pro/avizor/media>

que identifique a una persona u objeto, y quizás otra información en una etiqueta RFID, compuesta por un microchip conectado a una antena. Dicha antena permite que el chip transmita la información de identificación a un lector, el cual convierte las ondas de radio reflejadas por la etiqueta RFID en información digital que luego se puede transmitir a sistemas informáticos que puedan procesarla.”²

Baja frecuencia (120 ~ 135 KHz). La banda tiene una gran demanda en el mercado, el uso de restricciones el rendimiento no es de impacto ambiental, de bajo costo, por lo general menor que la distancia máxima de 60 cm de reconocimiento, utilizados principalmente en el control de acceso.

Alta frecuencia (10 ~ 15 Mhz). El rango de funcionamiento es inferior a 100 cm y es utilizado en sistemas de gestión y logística.

UHF (850 ~ 960 Mhz). Aquí la distancia de máximo reconocimiento es de hasta 10 metros, pero se ve afectado por condiciones ambientales, su precio es elevado y como ejemplo se lo usa en la identificación de contenedores, vehículos y otros.

Microondas (2,45 ~ 5,8 GHz). La banda de reconocimiento llega a distancias de 100 metros, pero su precio es el más elevado, utilizado principalmente en sistemas de transporte inteligente.

2 <http://www.rfidjournal.com/faq/> Autor: RFID Journal/ Año 2012

2.5.1 Sensor ID12

Los módulos usados son los ID-12, de estos existen también el ID-2 e ID-20, estos módulos poseen una antena interna y poseen rangos de lectura de 12 cm a 16 cm. respectivamente. Los módulos ID-12, si se le incluye una antena externa, puede tener alcances de hasta 25 cm. La Figura 6 presenta al sensor ID12.



Figura 6 Sensor ID12/ Año 2012

Fuente: http://www.id-innovations.com/httpdocs/EM%20module%20SERIES%202007-10-9_wfinal%20v22.pdf

Los lectores poseen los siguientes tipos de formatos de datos:

- ASCII,
- Wiegand26,
- Magnética ABA track2.

2.5.2 Distribución de pines

A continuación se explica en la Figura 7, la configuración de pines para el elemento

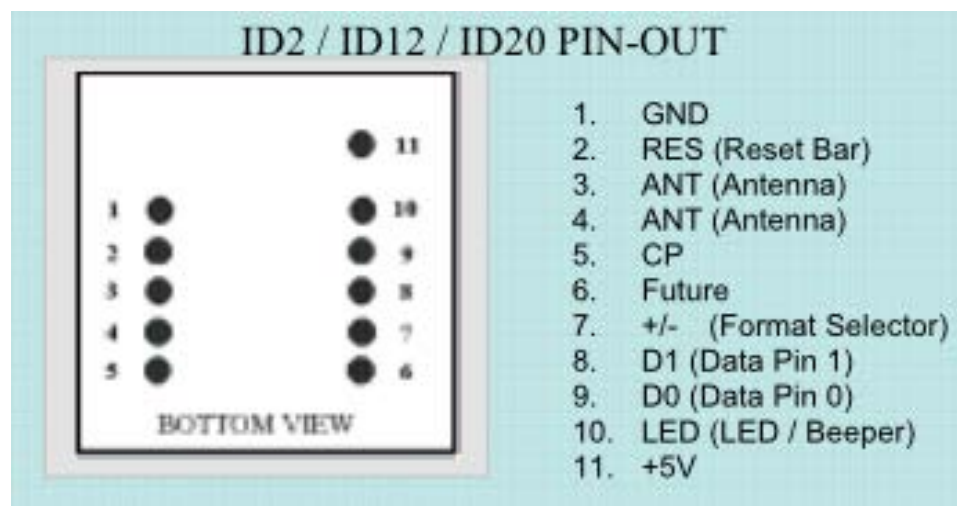


Figura 7 Configuración de pines del Sensor ID12/ Año 2012

Fuente: http://www.id_innovations.com/httpdocs/EM%20module%20SERIES%202007-10-9_wfinal%20v22.pdf

La tabla 4 explica de una manera más detallada el funcionamiento de cada pin del sensor ID12.

Pin	Nombre	Función	Descripción
1	Vss	Alimentación	GND
2	Vdd	Reset	+5V
3	Antena	Para antena externa y capacitor de afinación	Con esta antena se puede aumentar el alcance del módulo hasta los 25 cm.
4	Antena	Para antena externa y capacitor de afinación	Con esta antena se puede aumentar el alcance del modulo hasta los 25 cm.
5	-	No usado	
6	Future	Future	Future
7	Selector de Formato (+/-)	Puente a GND.	Para conversión de datos Negativo.
8	Data 1	CMOS	Señal de Reloj
9	Data 0	Datos en TTL	Datos
10	Zumbador	Salida	Indicación de Tarjeta Pasada.
11	Alimentación	+5v	+5v

Tabla 4 Diagrama de pines del Sensor ID12 / Año 2012

Fuente: http://www.id-innovations.com/httpdocs/EM%20moudule%20SERIES%202007-10-9_wfinal%20v22.pdf

2.5.3 Circuito base para el módulo RFID ID-12

En la Figura 8 se puede observar las conexiones para el sensor ID12, el conjunto de los elementos formados por U1, D1, C1 y C2, establecen el voltaje a 5 V. Prácticamente el sensor internamente hace el procedimiento de reconocimiento de la tarjeta RFID a 125 KHz, con ayuda del pin 10 y con una etapa amplificadora Q1, activa un sonido beeper y el led D2.

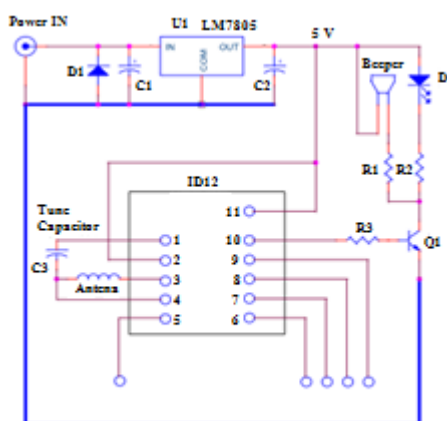


Figura 8 Conexiones de pines del Sensor ID12 / Año 2012

Fuente: http://www.id_innovations.com/httpdocs/EM%20moudule%20SERIES%202007-10-9_wfinal%20v22.pdf

2.6 Integrado DS1307

El integrado en mención es el encargado de generar la fecha y hora del sistema, esto en conjunto con el cristal de cuarzo, es una parte importante del proyecto ya que con esta fecha y hora trabajará el programa principal.

2.6.1 Descripción general



Figura 9 DS1307

Fuente: Autores

Reloj de tiempo real (RTC), cuenta segundos, minutos, horas, fecha del mes, mes días de la semana y hasta el año 2.100. Este CI es un reloj de tiempo real exacto, el cual automáticamente, mantiene el tiempo y la fecha actual, incluyendo compensación para meses con menos de 31 días y

saltos de año. Se puede observar en la Figura 9 la presentación de dicho dispositivo.

La dirección y los datos son transferidos vía serial por 2-wire, en bus bi-direccional.

También el reloj opera en formato de 24 horas o en formato de 12 horas AM/PM. Un cristal de cuarzo estándar, de bajo costo, a 32.768kHz entre los pines 1 y 2 para proveer tiempo base exacto.

Opcionalmente se le puede conectar al pin 3, baterías de respaldo de 3 volt, asegurando que se mantendrá el tiempo a la fecha aunque esté desconectada la fuente de tensión del circuito principal. El circuito integrado automáticamente detecta que se ha removido la energía en el circuito principal y se conectan las baterías de respaldo cuando es requerido, el pin 4 estará conectado a tierra y el pin 5 a positivo a una fuente de 5 voltios, el voltaje mínimo al que trabaja es 4.5v y el máximo 5.5v.

El pin 7 es una salida de colector abierto, que puede ser programada para hacer “flash” cada 1Hz. Esto permite la colocación de un led como indicador de segundos en aplicaciones de reloj. El circuito integrado también tiene 56 bytes de memoria RAM para propósito general, el cual puede ser usado como memoria extra por si es requerido.

El pin 6 (SCL) y el pin 5 (SDA) van conectados al circuito en donde se transferirá la información de manera serial, por medio del protocolo I2C.

2.6.2 Diagrama de pines

A continuación se describe el diagrama de pines del DS1307 para un mayor entendimiento de este circuito integrado, mejor conocido como RTC (Real Time Clock), observar la Figura 10.

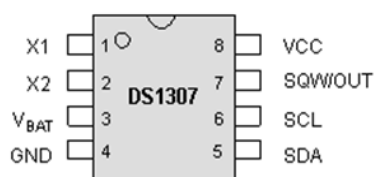


Figura 10 Diagrama de pines DS1307
Fuente: Autores

A continuación se explica de una manera más detallada la distribución de pines con sus respectivas características (Tabla 5).

Pin	Nombre	Función	Descripción
1	X1	XTAL	Pines para conectar XTAL de cuarzo estándar externo de 32.768 KHz. Los capacitores para estabilizar el circuito oscilador se incluye internamente.
2	X2		
3	VBAT	Batería	Pin para conectar opcionalmente una batería de 2.0v a 3.5v. Normalmente el DS1307 operará con su fuente del pin Vcc. En ausencia de dicha tensión o cuando su nivel caiga por debajo V _{BAT} , el DS1307 empezará a trabajar con la batería.
4	VSS	Alimentación	GND.
5	SDA	Pines de comunicación	Pines de interfaz I ² C.
6	SCL		
7	SOUT	Onda Cuadrada	Por este pin el integrado puede obtener una onda cuadrada de cuatro frecuencias: 1 Hz, 4.096 KHz, 8.192 KHz ó 32.768 KHz. Se configura con el registro de control. Es un pin de drenador abierto y por tanto necesitará de una pull up si se usa.
8	VDD	Alimentación	+5V.

Tabla 5 Diagrama de pines del DS1307 / Año 2012

Fuente: <http://www.sparkfun.com/datasheets/Components/DS1307.pdf>

2.6.3 Mapa de memoria

* Realiza conteos de segundos, minutos, horas, días, fecha, mes, año, hasta el 2100.

* Opera como esclavo.

* El acceso a la memoria se obtiene mediante la aplicación de un START CONDITION y proporcionar un código de identificación del dispositivo seguido por un registro de dirección.

En la Figura 11 se realiza un mapa de memoria del integrado.

	7	6	5	4	3	2	1	0	
00H	SEGUNDOS								00 a 59
01H	MINUTOS								00 a 59
02H	HORAS								01 a 12 00 a 23
03H	DIA								1 a 7
04H	FECHA								01 a 28/29 01 a 30 01 a 31
05H	MES								01 al 12
06H	AÑO								00 a 99
07H	CONTROL								
08H :: 3FH	RAM 56 X 8								

Figura 11 Mapa de memoria del RTC DS1307 / Año 2012
Fuente: <http://www.sparkfun.com/datasheets/Components/DS1307.pdf>

2.6.4 Segmentación

A continuación en la Figura 12, se explica de una manera más detallada la segmentación del integrado, es decir la característica de cada bit y la manera de poder manipular estos datos.

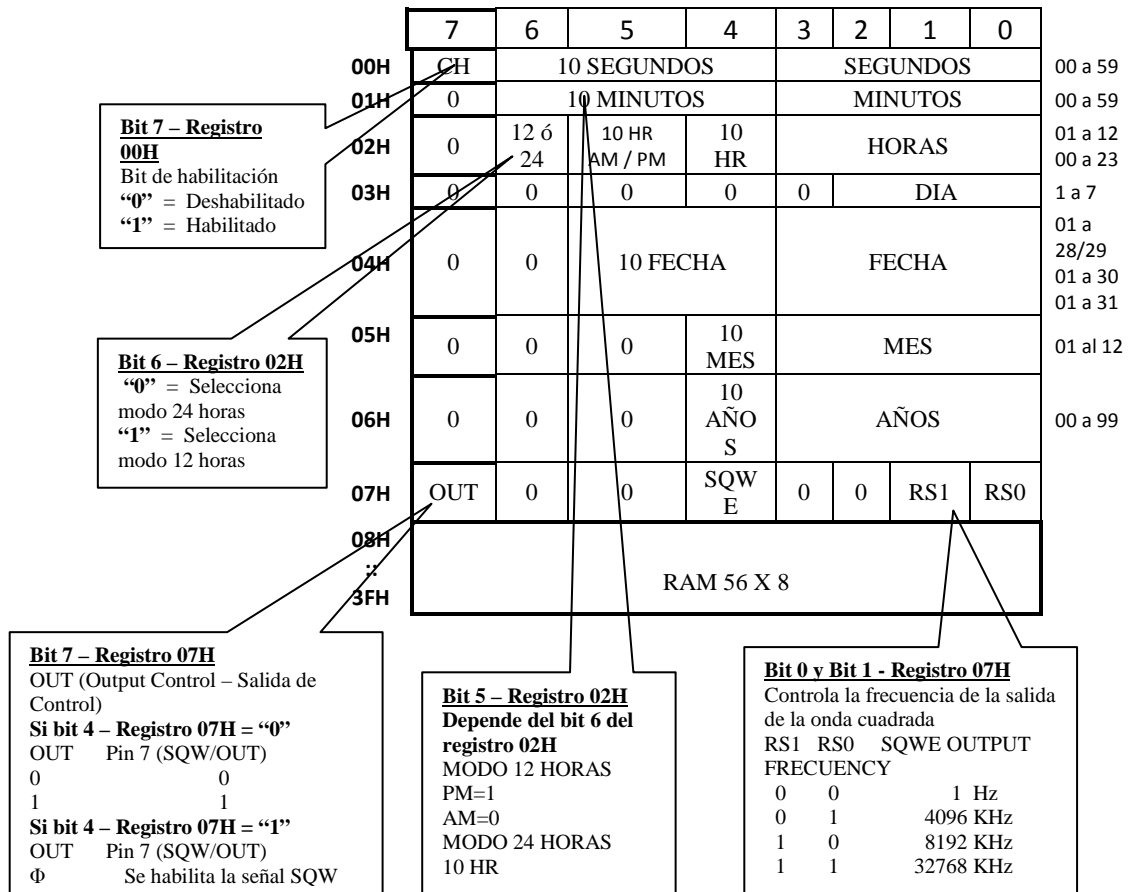


Figura 12 Segmentación del RTC DS1307 / Año 2012
Fuente: <http://www.sparkfun.com/datasheets/Components/DS1307.pdf>

2.6.5 I²C (Inter Integrated Circuit)

Se trata de un protocolo serie desarrollado por Philips Semiconductors usado por muchos integrados para comunicarse entre ellos, para su funcionamiento requiere sólo dos líneas, una de reloj (SCL) y otra de datos (SDA) junto a dos resistencias de pull-up con cada una de estas líneas.

Existen todo tipo de circuitos integrados con un bus I2C, termómetros, memorias, relojes de tiempo real, drivers de Display, etc. y en nuestro caso el sensor de ultrasonidos que vamos a utilizar entrega sus mediciones a través de este bus, como indica la Figura 13.

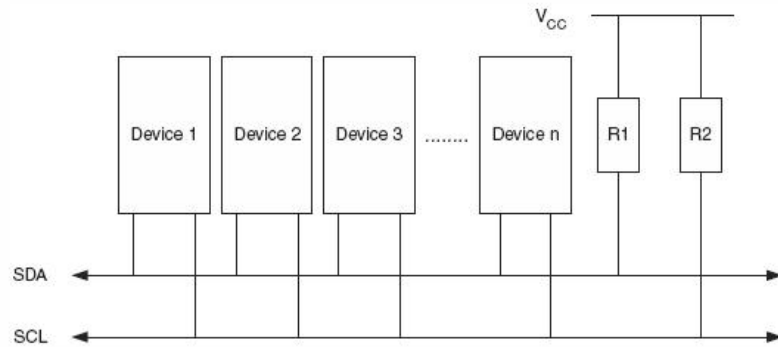


Figura 13 Interconexión de dispositivos / Año 2012
Fuente: <http://jmnlab.com/i2c/i2c.html>

Por lo que conocer como funciona el bus y aprender a usarlo es interesante en esto de la robótica.

“En los AVR en protocolo I2C lo encontramos con el nombre de TWI, nos permite conectar hasta 128 integrados al bus (límite capacitancia del bus 400 pF) usando sólo dos líneas y añadiendo unas resistencias de pull-up.”³

2.6.6 Declaración I2CWRITE y I2CREAD.

Hay declaraciones para escribir y leer datos en un chip EEPROM serial usando una interfaz I²C de 2 hilos, funcionan en modo I²C Master y también puede ser utilizado para comunicarse con otros dispositivos de interfaz I²C como sensores de temperatura, reloj calendarios, conversores A/D, etc.

3 <http://jmnlab.com/i2c/i2c.html> / **Autor:** Club de Informáticay Robótica / Año 2012

Las declaraciones de la Figura 14, son usadas en la programación de los PIC.

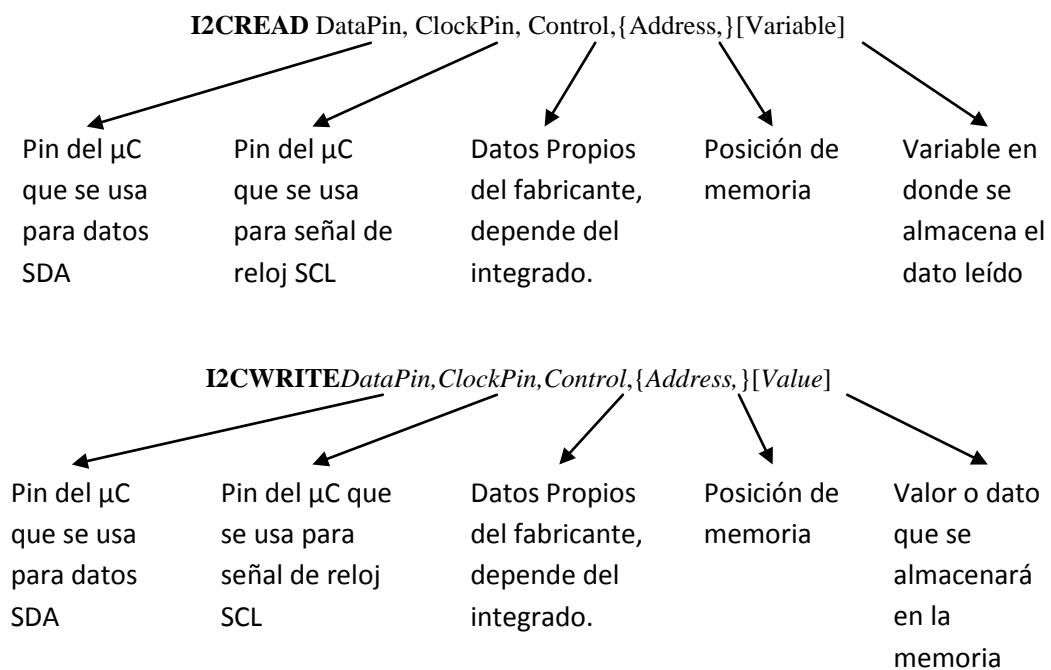


Figura 14 Sintaxis de los comandos usados para controlar I²C
Fuente: Autores

2.6.7 Relación entre la comunicación I²C y reloj calendario DS1307.

Como habíamos dicho anteriormente el bus I²C, permite la comunicación con algunos dispositivos como las memorias 24CXX, los procesadores de señal, codificadores de video, sensores de temperatura, RTC (reloj en tiempo real), etc. Por esta razón, para nuestro proyecto se usará el RTC DS1307 encargado de generar la hora y la fecha cuando nuestro proyecto así lo requiera

Entre unos aspectos a tomar en cuenta tenemos:

- ✓ Como dato importante el byte de control es %11010000, (propio del fabricante), y su modo de grabación de datos es en sistema hexadecimal.
- ✓ Como podrán observar la batería es lo que le mantiene en funcionamiento al RTC cuando no hay alimentación DC, por tal razón cuando apagamos todo el circuito, y luego lo volvemos a prender, notamos que el reloj no se ha desiguado, pero si retiramos la batería, el reloj se detiene cuando cortamos la alimentación del circuito, y cuando se lo vuelve a conectar, sigue corriendo

el tiempo pero continúa en el segundo que se quedó en el instante que se le cortó la alimentación.

- ✓ Para el desarrollo del programa se usará los comando explicados anteriormente I2CREAD y I2CWRITE

2.7 MEMORY STICK DATALOGGER

El Memory Stick Datalogger es un registrador de datos USB el cual permite conectar un dispositivo de almacenamiento masivo USB, como una unidad de disco USB, para sus proyectos con microcontrolador. El Vinculum IC /Firmware en el módulo de registro de datos maneja el sistema de archivos del Memory Stick, para que pueda compartir los archivos con su PC.

Todo este control mediante simples comandos seriales. Se puede observar en la Figura 15, una foto del dispositivo en mención.



Figura 15 Memory Stick Datalogger/ Año 2012

Fuente: <http://www.parallax.com/tabid/768/ProductID/434/Default.aspx>

“El corazón de la Memory Stick Datalogger es el vínculo incrustado USB Host Controller IC por FTDI (disponible a través de paralaje). Este circuito integrado permite la implementación de la funcionalidad USB Controladora de host dentro de los productos sin tener que lidiar con el bajo nivel PROTOCOLO USB. Además de proporcionar la interfaz de host, la Vinculum también maneja el sistema de archivos FAT (File Allocation Table – Tabla de Asignación de Arhivos) a través del firmware actualizable en memoria flash.”⁴

4 <http://www.robotshop.com/content/PDF/memorystickdataloggerv1.1-27937.pdf/>

Autor: RobotShop Distributor. Inc / **Año:** 2012

2.7.1 Características:

- Conexión directa a los dispositivos serie.
 - ✓ Adición de la función de red rápida y sencilla.
 - ✓ Proveer de personalización de Firmware.
- La estabilidad y fiabilidad del sistema mediante el uso de W5100 chip de hardware.
- Soporta conexión PPPoE.
- Permite la configuración de serie - con el comando simple y fácil.
- Soporta contraseña de la Seguridad.
- Programa de configuración de herramientas.
- Interfaz Ethernet 10/100 y un máximo de 230 Kbps Interfaz de serie.
- RoHS.

2.7.2 Operación del Memory Stick Datalogger

La parte principal del memory stick datalogger o el registro de datos es el USB Host Controller IC (Circuito integrado controlador Host USB). Permite al usuario controlar el memory stick con ayuda de comandos serie, los archivos deben estar en formato FAT, con esto pueden ser leídos y escritos por un computador. Estos archivos creados pueden ser importados como hojas de cálculo, donde se acceden a los datos de una manera fácil para el usuario.

2.7.3 Set de comandos

Hay que recordar que este módulo se lo controla con ayuda del Vinculum VNC1L siendo este el primero de la familia FTDI Vinculum adherido a un dispositivo controlador.

A continuación en la Tabla 6, se detallan comandos y respuestas del módulo que se usan para controlar el memory stick datalogger.

Extendend ASCII Comand for Terminal Mode	Shortened Hexadecimal Command for microprocessor mode	Función	Respuesta
Switching between Shortened and Extended Command sets			
'SCS' <cr>	\$10,\$0D	Switches to the shortened command set	This will return the prompt '>',\$0D to indicate that the device is in shortened command set mode
'ECS' <cr>	\$11,\$0D	Switches to the extended command set	This will return the prompt 'D: >',\$0D to indicate that the device is in extended command set mode
Responses to indicate if disk is online			
<cr>	\$0D	Check if online	This will return the appropriate prompt or 'no disk' message for the current command set
Response to Check if online for Extended Command Mode		If no valid disk is found	'No Disk', \$0D
		If a valid disk is found	'D: >',\$0D
Response to Check if online for Short Command Mode		If no valid disk is found	'ND', \$0D
		If a valid disk is found	'>',\$0D
Directoryoperations			
'DIR' <cr>	\$01,\$0D	Lists the current directory	A list of the names and directory names are returned. Each entry is terminated by \$0D. A directory entry has <sp>'DIR' after the name and before the \$0D.
'DIR' <sp><name><cr>	\$01,\$20,<name>,\$0D	Lists the name followed by the size. Use this before doing a the read to know how many bytes to expect.	\$0D,<name><sp><size in hex (4 bytes) LSB frst> \$0D
'DLD' <sp><name><cr>	\$05,\$20,<name>,\$0D	Delete directory	Deletes the directory <name> from the current directory <prompt> \$0D

'MKD' <sp><name><cr>	\$07,\$20,<name> ,\$0D	Make directory	Creates a new directory <name> in the current directory <prompt> \$0D
'CD' <sp><name><cr>	\$02,\$20,<name> ,\$0D	The current directory is changed to the new directory <name>	<prompt> \$0D
'CD' <sp>'..'<cr>	\$02,\$20,\$2E,\$2E,\$0D	Move up one directory level	<prompt> \$0D
File operations			
'RD' <sp><name><cr>	\$04,\$20,<name>,\$0D	Read the <name>	This will send back the entire the in binary to the monitor. The size should first be found by using the 'RD' <sp><name><cr>command so that the expected number of bytes is known. <prompt>\$0D
'RDF' <sp><size in hex (4 bytes)><cr>	\$0B,\$20, size in hex (4 bytes),\$0D	Read the data of <size in hex (4 bytes)> from the current open file.	This will send back the requested amount of data to the monitor. <prompt>\$0D
'DLF' <sp><name><cr>	\$07,\$20,<name>\$0D	Delete file <name>	This will delete the file from the current directory and free up the FAT sectors. <prompt>\$0D
'WRF' <sp><size in hex (4 bytes)><cr><data bytes of size><cr>	\$08,\$20,size in hex (4 bytes), \$0D, \$data, \$0D	Writes the data of <size in hex (4 bytes)> to the end of the current open file.	<prompt>\$0D
'OPW' <sp><name><cr>	\$09,\$20,<name>, \$0D	Open a file for writing to with 'WRF'	<prompt>\$0D
'OPR' <sp><name><cr>	\$0E,\$20,<name>, \$0D	Open a file for reading to with 'RDF'	<prompt>\$0D
'CLF' <sp><name><cr>	\$0A,\$20,<name>, \$0D	Closes a file for writing	<prompt>\$0D
'REN' <sp><orig name><sp><new name><cr>	\$0C,\$20,<orig name>, \$20, <new name><cr>	Rename a file or directory	<prompt>\$0D
'FS'<cr>	\$12,\$0D	Returns free space in bytes on disk	<free space in hex (4 bytes) LSB first>\$0D
Commands for UART monitor mode only			
'SBD' <sp><divisor (3 bytes)LSB first><cr>	\$14,\$20, divisor (3bytes) LSB first>,\$0D	Set Baud Rate (See Baud Rate Table)	<prompt>\$0D
Power Management Commands			
'SUD'<cr>	\$15,\$0D	Suspend the disk when not in use to conserve power. The disk will be woken up automatically the	<prompt>\$0D

		next time a disk command is sent to it.	
'WKD'<cr>	\$16,\$0D	Wake Disk and do not put it into suspend when not in use.	<prompt>\$0D
'SUM'<cr>	\$17,\$0D	Suspend Monitor and stop clocks	<prompt>\$0D
Debug commands			
'SD'<sp><sector number in ASCII hex><cr>	\$0,\$20,... \$0D	Sector dump. This is used for debug purposes and may be removed. e.g. 'SD 0000 <cr>' will dump sector 0000.'SD 0010'<cr> will dump sector 16 decimal.	Send back 512 bytes from the sector specified in HEX converted to ASCII. Every 16 bytes is followed by a \$0D. <prompt>\$0D
'IDD'<cr>	\$0F, \$0D	Identify disk drive. This will display information about the attached disk.	Send IDD data block and then <prompt>\$0D
'FWV'<cr>	\$01, \$0D	Get firmware versions	Display the version number of the main firmware and the reprogramming firmware in the VNCL1 'MAIN x.xx'\$0D 'RPRG x.xx'\$0D Then <prompt>\$0D

Tabla 6 Comandos usados para controlar el Memory Stick datalogger / Año 2012

Fuente: <http://www.parallax.com/Portals/0/Downloads/docs/prod/comm/VDAPFirmwareSpec.pdf>

También se presenta una tabla de errores que se obtiene como resultado cuando se manipula el módulo memory stick.

Error	Command Mode	Result
If command is unrecognised	Extended Command set	'Bad Command ', \$0D
	Shortened Command set	'BC ', \$0D
If command fails	Extended Command set	'Command Failed ', \$0D
	Shortened Command set	'CF ', \$0D

Tabla 7 Reporte de errores / Año 2012

Fuente: <http://www.parallax.com/Portals/0/Downloads/docs/prod/comm/VDAPFirmwareSpec.pdf>

2.7.4 Diagrama de pines

El Memory Stick Datalogger puede ser usado de dos maneras, ya sea por modo UART (Transmisor-Receptor Asíncrono Universal), o por SPI (Serial Peripheral Interfaz por sus siglas en inglés), siendo el primero usado para nuestro proyecto, ya que los PIC cuentan con un módulo UART incorporado.

2.7.4.1 Modo UART

Como se observa el gráfico para usarla con esta conexión UART, se necesita ubicar el jumper del lado derecho, tomar en cuenta que los Microcontroladores que se usarán contiene un módulo UART, con esto se lo relaciona y es el modo de comunicación que se usará en el proyecto.

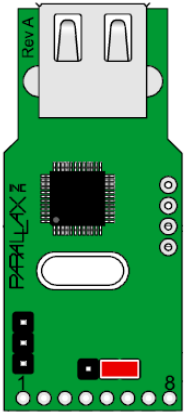
	Pin	Nombre	Descripción
	1	VSS	Conectar el sistema a negativo (tierra).
	2	RTS#	Respuesta de envío (Conectar a MCU CTS).
	3	VDD	Conectar a +5V.
	4	RXD	Recepción de datos (Conectar a MCU TXD).
	5	TXD	Transmisión de datos (Conectar a MCU RXD).
	6	CTS#	Clear to Send (Conectar a MCU RTS).
	7	NC	No Conectado.
8	RI#	Indicador Ring (Making this input low resumes from suspends).	

Tabla 8 Distribución de pines para controlar el Memory Stick datalogger en Modo UART/ Año2012
Fuente: www.parallax.com/Portals/Downloads/prod/comm/MemoryStickDataloggerV1.pdf

2.7.4.2 Modo SPI

Otra forma de conectar el memory stick es usando la comunicación SPI, tal como indica el ítem descripción en la tabla siguiente.

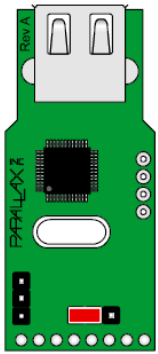
	Pin	Nombre	Descripción
	1	V _{SS}	Conectar el sistema a negativo (tierra).
	2	SDO	Salida de datos serial.
	3	V _{DD}	Conectar a +5V.
	4	SDI	Entrada Datos Serial.
	5	SCLK	Entrada de reloj serial.
	6	CS	Entrada Selección de chip.
	7	NC	No Conectado.
8	NC	No Conectado.	

Tabla 9 Distribución de pines para controlar el Memory Stick datalogger en Modo SPI / Año 2012
Fuente: <http://www.parallax.com/Portals/Downloads/docs/comm/MemoryStickDataloggerV1.1.pdf>

2.7.5 Especificaciones técnicas

A continuación se da especificaciones técnicas para poder usar el módulo de una manera adecuada, sin ser víctima de sobre voltaje o comportamientos erróneos del módulo.

Símbolo	Significado	Mínimo	Típico	Máximo	Unidad
V _{DD}	Voltaje de poder	4.75	5.0	5.25	V
-	Temperatura	- 65°	-	- 65°	C
-	Temperatura de operación	0°	-	- 70°	C
I _{VDD}	Corriente de la fuente	-	25	-	mA
I _{VDD}	Corriente en standby	1	2	2	mA

Tabla 10 Especificaciones técnicas para controlar el Memory Stick datalogger / Año 2012

Fuente:

<http://www.parallax.com/Portals/0/Downloads/docs/prod/comm/MemoryStickDataloggerV1.1.pdf>

2.7.6 Indicaciones led del Memory Stick Datalogger

El módulo tiene un LED indicador, en el cual se ve reflejado los diferentes estados en que se encontrará el Memory Stick durante el envío de información o respuestas del mismo, todo esto se explica en la tabla siguiente.

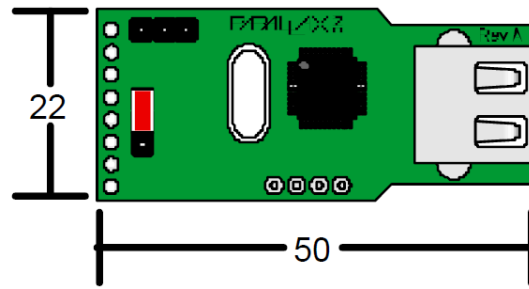
Observación	Respuesta
Power ON	LED Green y LED Red flash alternadamente por 2 segundos. Esto se repite hasta que se actualice.
Inicialización del disco USB	LED Green ON y LED Red OFF.
Lectura del disco USB	LED Green OFF y LED Red ON.
Remover el disco USB	LED Green OFF y LED Red OFF.
Lectura del disco USB	LED Green OFF y LED Red ON.
Envío de comandos desde software hasta el Puerto del disco USB	LED Green OFF y LED Red Flash.
Envío de comandos desde software hasta el Puerto del disco USB removido	LED Green OFF y LED Red OFF.

Tabla 11 Estado del LED en el Memory Stick Datalogger / Año 2012

Fuente:

<http://www.parallax.com/Portals/0/Downloads/docs/prod/comm/MemoryStickDataloggerV1.1.pdf>

Para poder ubicarlo en las placas electrónicas, se debe conocer las medidas exactas para tomar las consideraciones necesarias. Tomando en cuenta la Figura 16.



units = mm

Figura 16 Dimensiones del módulo memory stick datalogger / Año 2012

Fuente:

<http://www.parallax.com/Portals/0/Downloads/docs/prod/comm/MemoryStickDataloggerV1.1.pdf>

2.8 Integrado SN75176

El SN75176 se encarga de generar los niveles de voltaje necesarios para la transmisión de la red RS485, y con ello llevar a cabo la recepción y transmisión de datos en el sistema.

2.8.1 Descripción general

Para lograr la comunicación entre módulos, y así poder formar una red donde todos los módulos se encuentran en paralelo, se ha usado este integrado, en la cual elabora una interfaz del tipo RS-485, para su diseño, donde se utilizan dos circuitos integrados con la matricula SN75176 de Texas Instruments, uno es para la recepción de datos y otro para la transmisión.

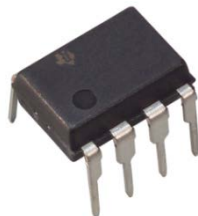


Figura 17 Aspecto físico del SN75176/ Año 2012

Fuente: <http://www.sigmaelectronica.net/sn75176-p-572.html>

Estos dispositivos se encargan de hacer la conversión entre los niveles TTL del microcontrolador y las señales del tipo diferencial que se utilizan el bus RS-485.

Vale la pena decir que en el controlador de transmisión se agregó una línea de habilitación, esto se debe a que todas las salidas de los Microcontroladores están conectadas a la línea de recepción del ordenador, así cada uno está siempre deshabilitado para enviar datos y solo se habilitará en el momento en que deba hacer una transmisión, evitando así conflictos o choques de información en la línea o bus de datos.

2.8.2 Diagrama de pines

En las terminales VCC y GND se encuentra la alimentación del circuito, que este caso es de +5V.

La terminal R0 y DI recibe un nivel lógico TTL si y solo si la línea RE se habilita y como se puede observar es con un '0' lógico.

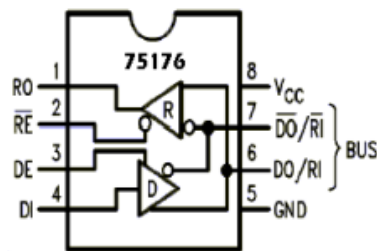


Figura 18 Diagrama de pines del SN75176/ Año 2012
Fuente: <http://www.sigmaelectronica.net/sn75176-p-572.html>

Las terminales D0 y $\overline{D0}$ reciben también el nombre de A y B y son sobre estas líneas las que forman el Bus de Transmisión y Recepción.

Como se puede observar, en la Figura 18, cada chip consta de un transmisor y un receptor, si las terminales RE (Pin 2) y DE (Pin 3) se unen entre sí con un solo bit se puede controlar el flujo de la información.

2.9 Convertidor WIZNET

El WIZNET es un elemento relativamente nuevo que se encarga de generar el protocolo (conjunto de reglas) para llevar la información desde una red RS485 hasta a una red Ethernet.

2.9.1 Características

El módulo WIZ110SR basado en el chip W5100 funciona como un Gateway que convierte desde el protocolo RS-232 al protocolo TCP/IP y viceversa.

“Permite entre otras cosas dar conectividad Ethernet a los equipos que no lo poseen, pero sin lugar a dudas el mayor uso que se le da es por medio de la creación de puertos COM virtuales.”⁵

En la Figura 19 se observa el aspecto físico del Wiznet para poder tener una mejor referencia.



Figura 19 Aspecto físico del Wiznet/ Año 2012

Fuente: <http://www.olimex.cl/pdf/WizNet/Guia%20MCI%20-%20Conversor%20Serial%20Ethernet%20WizNet%20v1.1.pdf>

Entre las características principales de WizNet se tienen:

- ✓ Módulo todo incluido para RS-232 y Ethernet.
- ✓ Simple y rápida implementación de la red.
- ✓ Alta estabilidad y confiabilidad usando el chip WizNet W5100, con un poderoso stack TCP/IP totalmente diseñado por hardware.
- ✓ Fácil y poderoso programa de configuración por medio del puerto serial.
- ✓ Interfaz Ethernet 10/100Mbps, Interfaz Serial Max 230Kbps.

5 <http://www.olimex.cl/pdf/WizNet/Guia%20MCI%20-%20Conversor%20Serial%20Ethernet%20WizNet%20v1.1.pdf>
Autor: MCI Electronics / **Año:** 2012

Las especificaciones generales se muestran a continuación:

Arquitectura	TCP / IP	W5100
	PHY	Incluido en W5100
	Serial	RS-232
	MCU	GC89L591A0-MQ44I (compatible con 80C52) 62K Bytes in-system programmable(ISP) flash
Network	Interfaz	10/100 Mbps con detección automático. Conector RJ-45.
	Protocolo	TCP, UDP, IP, ARP, ICMP, MAC, PPPoE
Serial	Señales	TXD, RXD, RTS, CTS, GND
	Parámetros	Parity : None, Even, Odd Data bits : 7,8 Flow Control : RTS/CTS, XON/XOFF
	Velocidad	Up to 230Kbps
Dimensiones		75mm X 45mm (PCB ancho X largo)
Voltaje de entrada		5V
Consumo de Potencia		DC 5V, Under 180mA
Temperatura		0°C ~ 80°C (Operando) -40°C ~ 85°C (Embalado)
Humedad		10 ~ 90%

Tabla 12 Características generales del Wiznet/ Año 2012

Fuente:

<http://www.olimex.cl/pdf/WizNet/Guia%20MCI%20Convertor%20Serial%20Ethernet%20WizNet%20v1.1.pdf>

2.9.2 Especificaciones técnicas

- MCU 8051,
- FLASH 62 KB (MCU interna),
- SRAM 16 KB (MCU interna),
- EEPROM 2 KB (MCU interna).

El módulo tiene dos interfaces de comunicación con el exterior, uno de ellos es el denominado RJ45, como indica la figura.

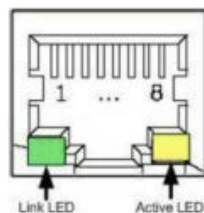


Figura 20 Conector RJ45 / Año 2012

Pin	Signal
1	TX+
2	TX-
3	RX+
6	RX-

Tabla 13 Pines de conexión para RJ45 / Año 2012

Fuente: <http://www.olimex.cl/pdf/WizNet/Guia%20MCI%20Convertor%20Serial%20Ethernet%20WizNet%20v1.1.pdf>

Otro interfaz se basa físicamente en el conector DB9, el cual tiene las siguientes características.

Pin Number	Signal	Description
1	NC	Not Connected
2	RxD	Receive Data
3	TxD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	NC	Not Connected

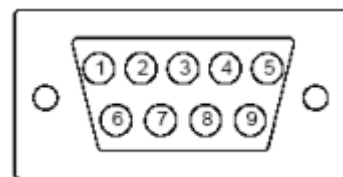


Tabla 14 Descripción de pines del puerto serial / Año 2012

Figura 21 Conector RS232 del Wiznet / Año 2012

Fuente:http://www.olimex.cl/pdf/WizNet/Guia%20MCI%20_%20Conversor%20Serial%20Ethernet%20WizNet%20v1.1.pdf

Para poder testear si el módulo Wiznet funciona apropiadamente, viene con un programa adicional que puede ser usado para hacer ping, y con el cual se lo configura para diferentes propósitos.

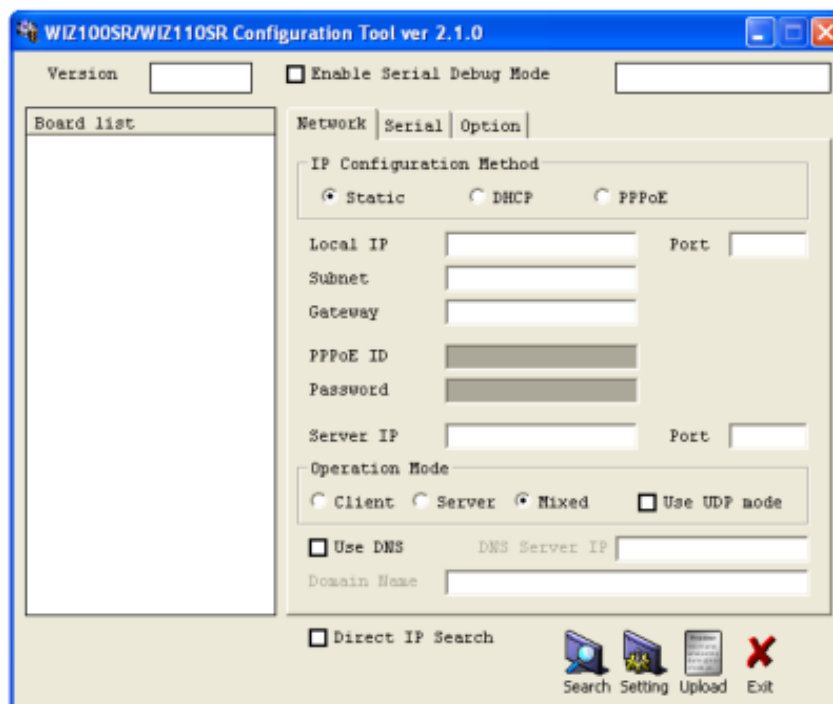


Figura 22 Ventana de Inicio del programa / Año 2012

Fuente:http://www.olimex.cl/pdf/WizNet/Guia%20MCI%20_%20Conversor%20Serial%20Ethernet%20WizNet%20v1.1.pdf

2.10 Software

Comenzamos buscando siguiendo la ruta Equipo/Clase 1(F:) /DISK y ejecutamos el ícono Setup como indica la Figura 23.

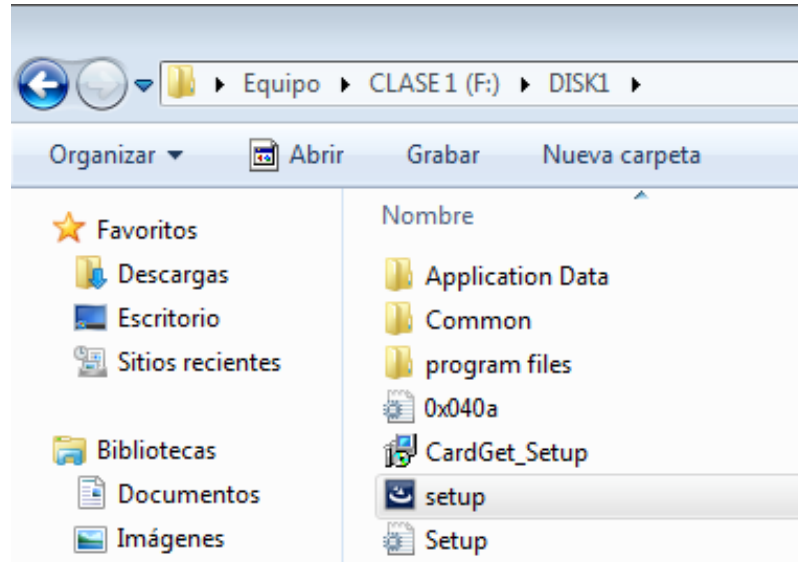


Figura 23 Ventana de instalación para el software Cardget
Fuente: Autores

Después de esto, se muestra la siguiente ventana de la Figura 24, dando la bienvenida para instalar el software, para esto se da un clic en Siguiente.

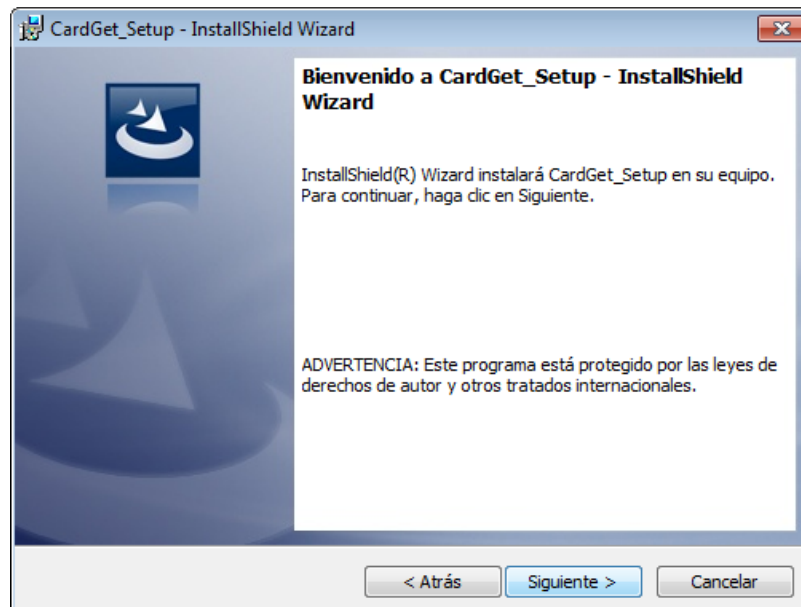


Figura 24 Ventana de Setup. Software Cardget
Fuente: Autores

En la siguiente ventana, en la Figura 25, se tiene que aceptar los términos del contrato de licencia. Para continuar con el proceso de instalación.

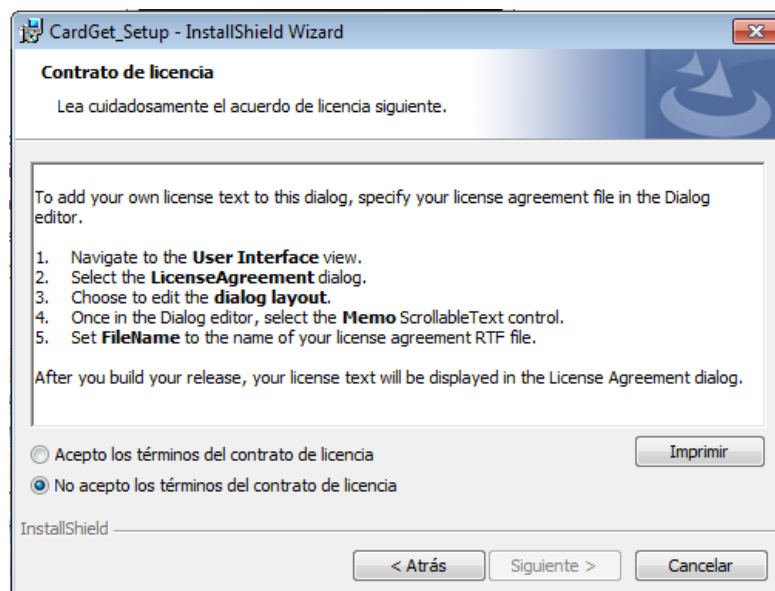


Figura 25 Ventana de Contrato de licencia. Software Cardget
Fuente: Autores

En la siguiente ventana (Figura 26) se muestra información de última hora necesaria para la instalación del software, se debe dar un clic en Siguinte.

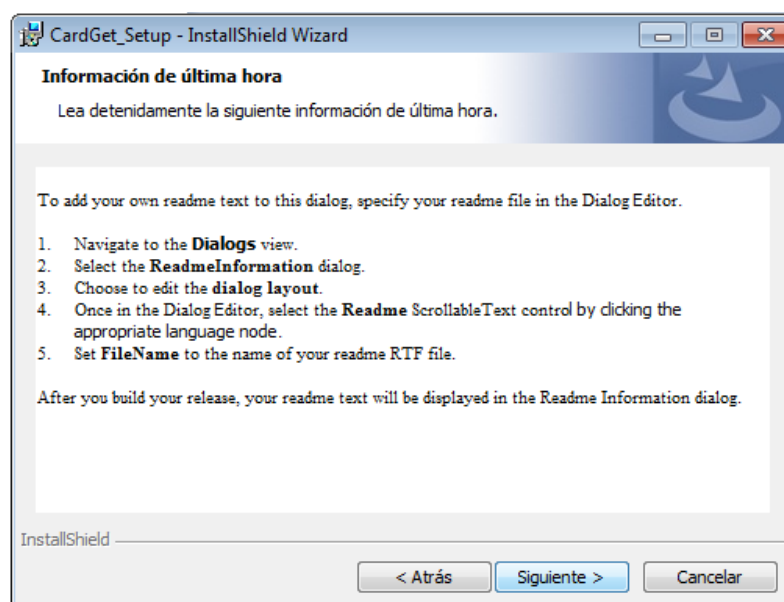


Figura 26 Ventana de Información. Software Cardget
Fuente: Autores

Continuamos ingresando un Nombre de usuario y una Organización como datos sugeridos en la instalación, esto es indicado en la Figura 27.

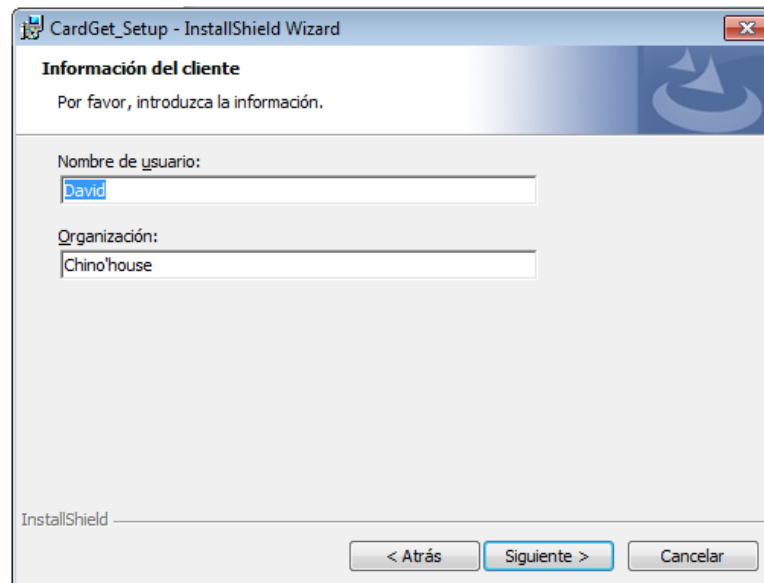


Figura 27 Ventana de Información del cliente. Software Cardget
Fuente: Autores

Después se presenta la ruta en donde será instalado el programa Cardget en nuestro caso es C:/Program Files/RFID/TESIS/, según como indica la Figura 28, presionamos el ícono Siguiete.

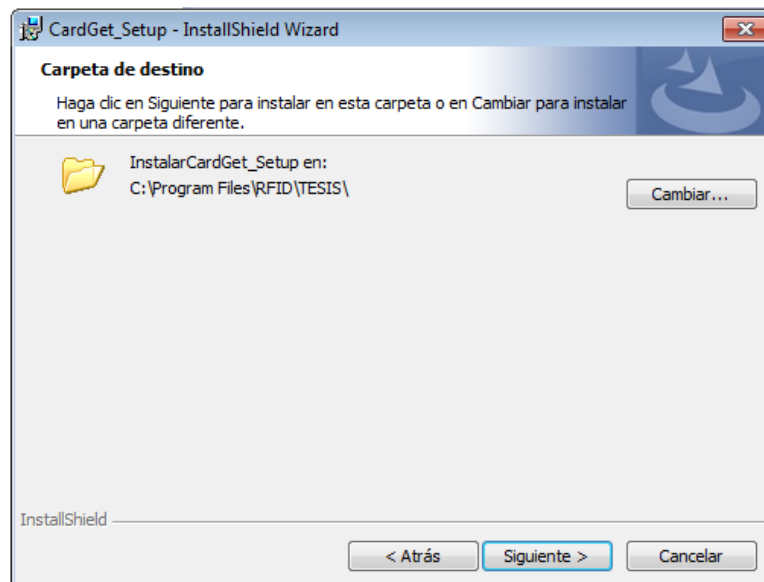


Figura 28 Ventana de Carpeta de destino. Software Cardget
Fuente: Autores

En estas instancias el sistema operativo está en la última fase y preparado para instalar el software, como se presenta en la Figura 29. Presionamos el ícono Instalar.

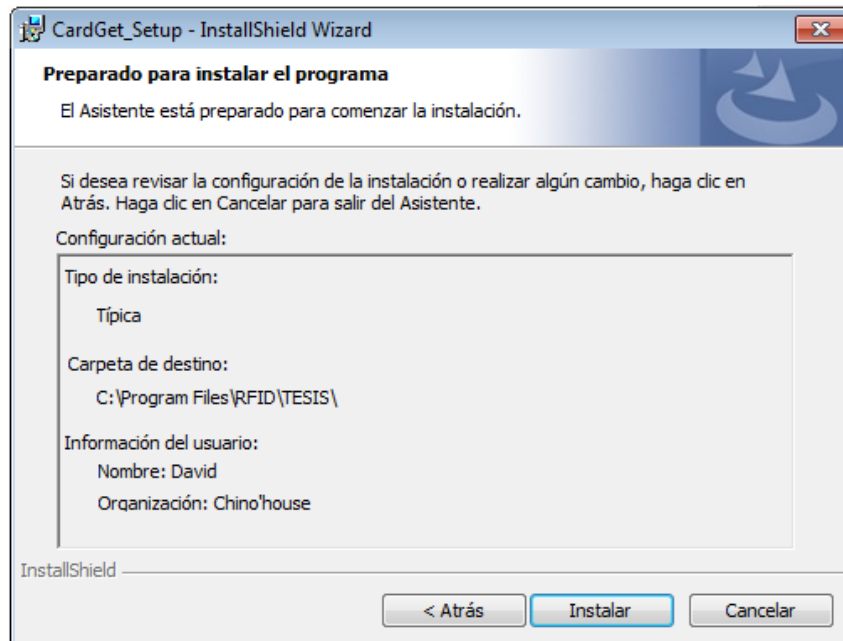


Figura 29 Ventana de Preparado para instalar el programa. Software Cardget
Fuente: Autores

Luego de esto se inicia la instalación del software como se observa en la Figura 30.

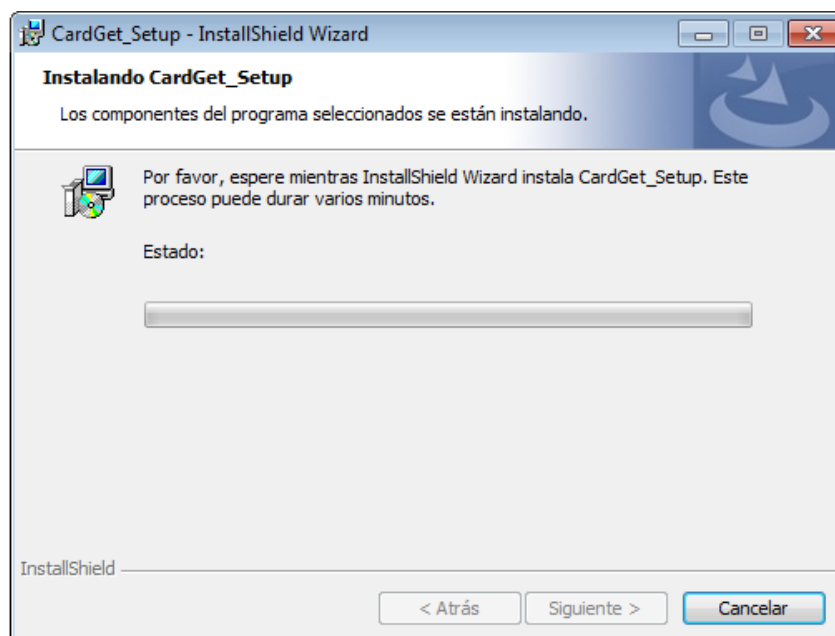


Figura 30 Ventana de Instalando Cardget
Fuente: Autores

En este paso prácticamente ya está instalado el software y pide la ejecución del mismo.

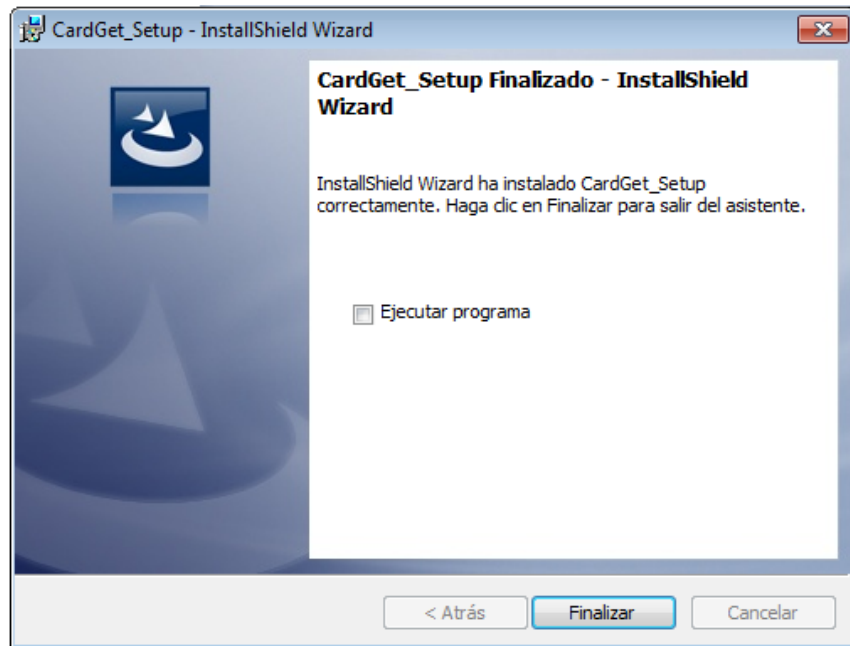


Figura 31 Ventana de Finalización Cardget
Fuente: Autores

Para esto nos dirigimos al ícono ejecutable, C:\Program Files\RFID\TESIS, editar CardGet.exe haciendo clic derecho en Editar, tal y como se observa en la Figura 32.

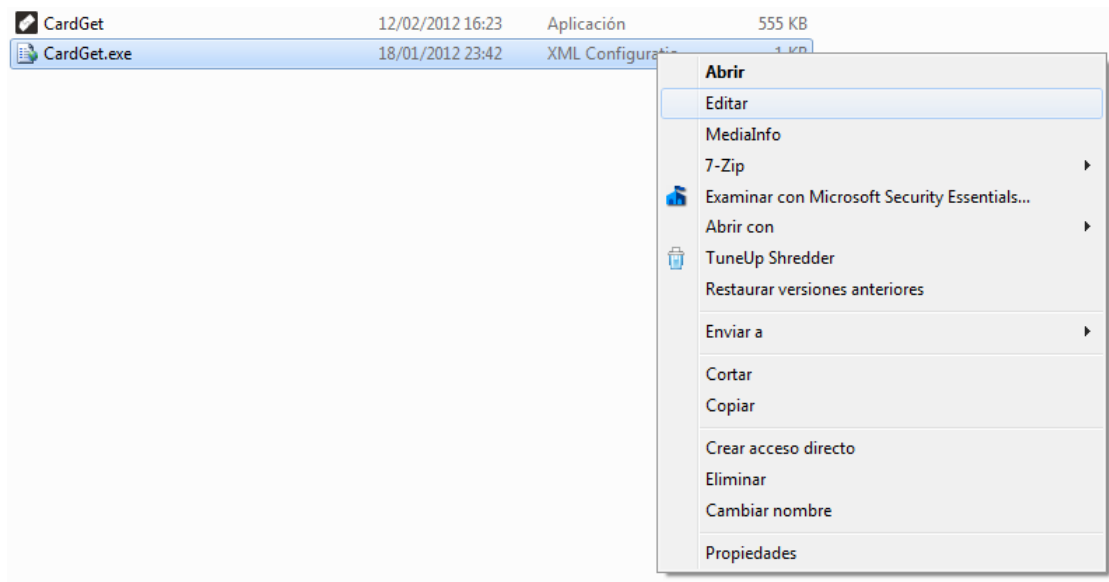


Figura 32 Ejecución del software Cardget
Fuente: Autores

A continuación se debe realizar un cambio, para esto nos dirigimos a la siguiente ruta C:\Program Files\RFID\TESIS, presionamos clic derecho y editar Cardget, tal y como se observa en la Figura 33, se de cambiar la dirección de Filetransfer por la que se instaló.

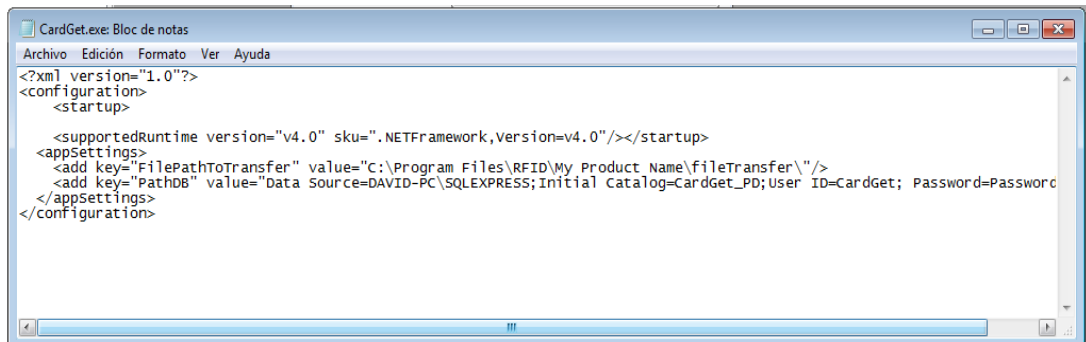


Figura 33 Ventana de Filetransfer. Software Cardget
Fuente: Autores

También se debe variar la dirección DataSource por la que indica SQL Server, esto se demuestra en la Figura 34.



Figura 34 Ventana de SQL Server. Software Cardget
Fuente: Autores

Cabe indicar que los requerimientos para la instalación del software Cardget son:

- Microsoft Office 2010
- SQL Server 2008 R2

CAPÍTULO 3

DISEÑO ELECTRÓNICO DEL SISTEMA

En este capítulo se explicará las pruebas de funcionamiento de nuevos dispositivos, que se usan en el proyecto como son el lector ID12 y el Memory Stick Datalogger. Así como también el desarrollo del proyecto a nivel de hardware.

3.1 Lector ID12 (Sensor RFID)

Para comprobar el funcionamiento de este dispositivo, se tiene que establecer una comunicación serial (por medio del software Hyperterminal), siguiendo los siguientes pasos:

Generar una nueva conexión (Hyperterminal), elegir un nombre y un ícono cualesquiera, como indica la Figura 35.

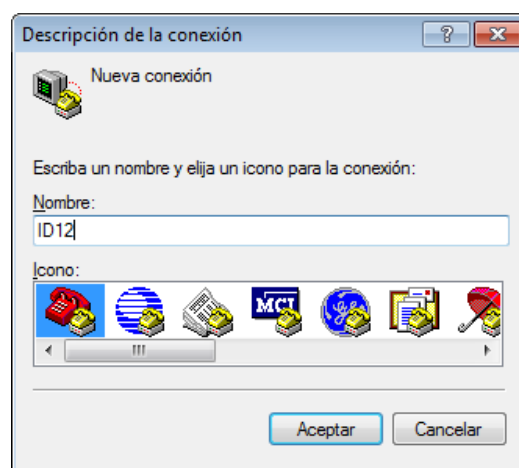


Figura 35 Ventana Nueva Conexión ID12
Fuente: Autores



Figura 36 Ventana Conectar a ID12
Fuente: Autores

Se escoge el puerto de comunicación, para nuestro ejemplo fue el COM4, tal como se observa en la imagen siguiente.

Se configura el puerto a los siguientes valores mostrados en la Figura 37.

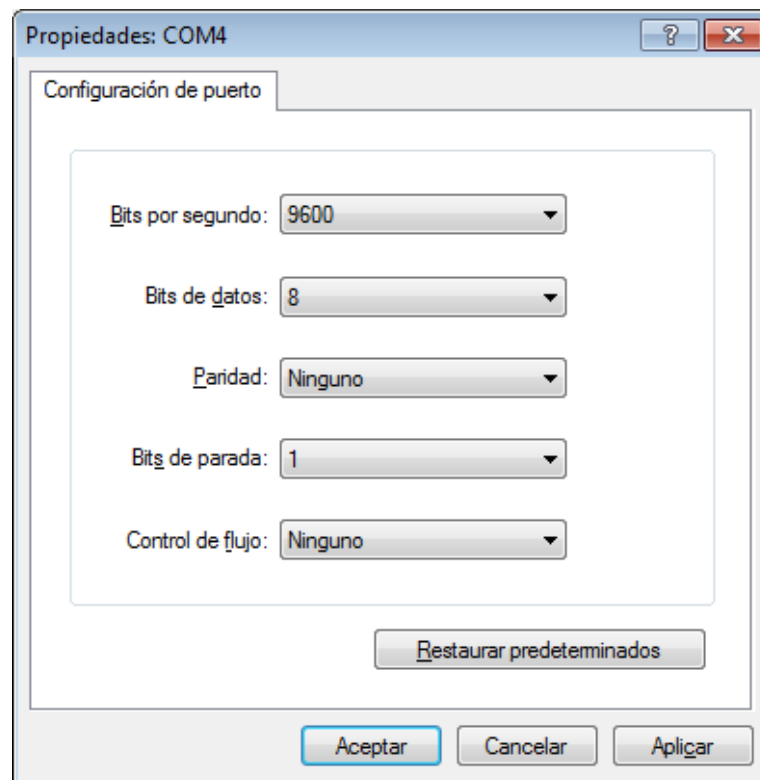


Figura 37 Ventana *Propiedades Comunicación del ID12*
Fuente: Autores

Después de esto, se establecera comunicación con el puerto designado (COM4) y se conectarán los elementos como indica la Figura 38.

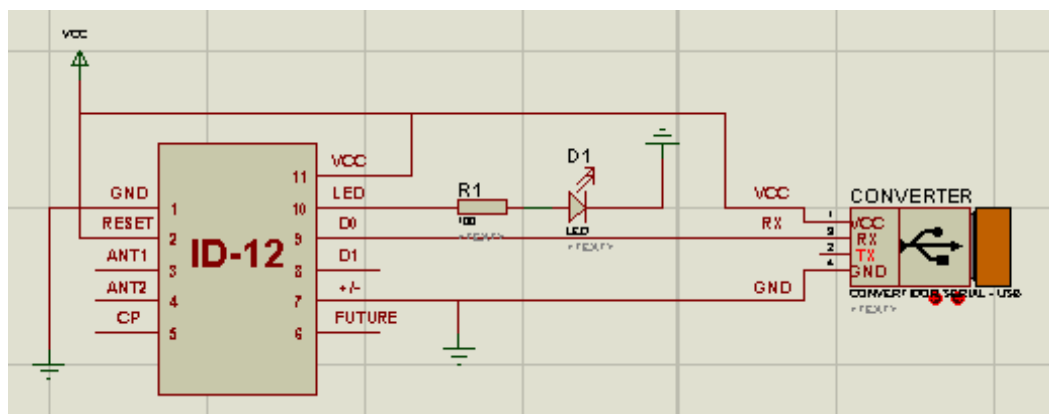


Figura 38 Pruebas de funcionamiento Sensor RFID
Fuente: Autores

Después de esto se verifican las tarjetas RFID, pasando las mismas por el sensor ID12, de tal forma que en la pantalla del Hyperterminal, se presenten los códigos de las tarjetas escogidas.

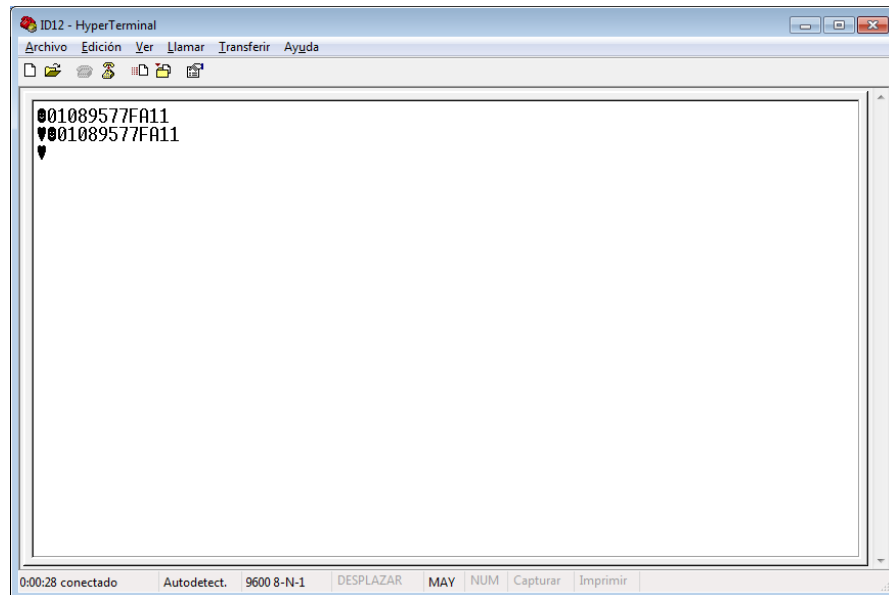


Figura 39 Ventana de comunicación del Hyperterminal RFID

Fuente: Autores

3.2 Memory Stick Datalogger

A continuación se describen los pasos para configurar el Memory Stick Datalogger, luego de esto se lo usará en el proyecto para los fines especificados.

3.2.1 Formatear Pendrive

Para iniciar las pruebas con el memory stick datalogger, se debe usar un pendrive previamente formateado siguiendo los pasos adjuntos:

Clic derecho sobre el ícono KINGSTON (F), se presenta un menú contextual como indica la Figura 40, y se elige la opción *Formatear*.

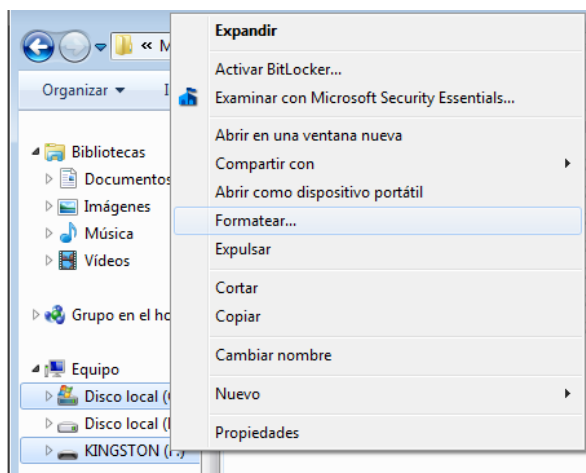


Figura 40 Ventana *Formato*

Fuente: Autores

La siguiente ventana me indica las propiedades para formatear dicho pendrive, en donde se elige *Capacidad*, *Sistema de archivo*, *Tamaño de la unidad de asignación*, escogiendo los valores que me indica la Figura 41. Luego se da un clic en *Iniciar*

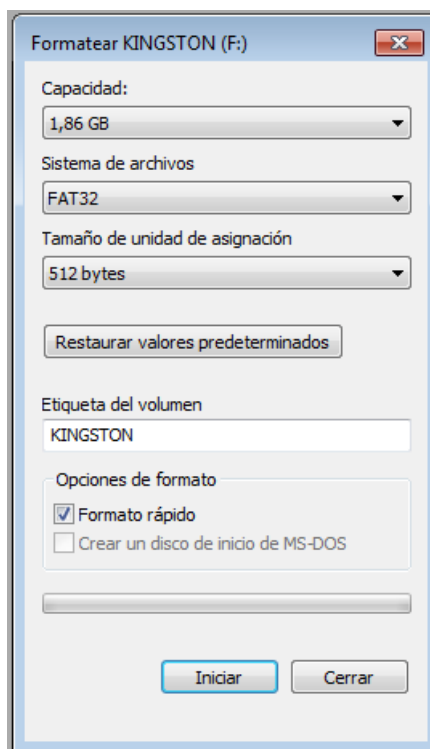


Figura 41 Ventana Iniciar formato
Fuente: Autores

Luego de esto, saldrá la siguiente ventana, indicando que se perderán los datos almacenados en el pendrive, presionamos *Aceptar*, en este momento el pendrive ya está con formato FAT32.

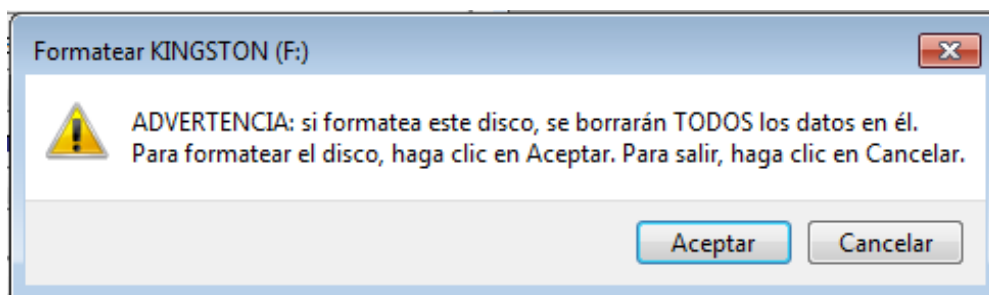


Figura 42 Ventana de Advertencia
Fuente: Autores

3.2.2 Conexión del Memory Stick Datalogger

A nivel hardware la conexión de los dispositivos será como se muestra en la Figura 43, en este podemos encontrar el convertidor USB a Serial, un pendrive y el memory stick datalogger

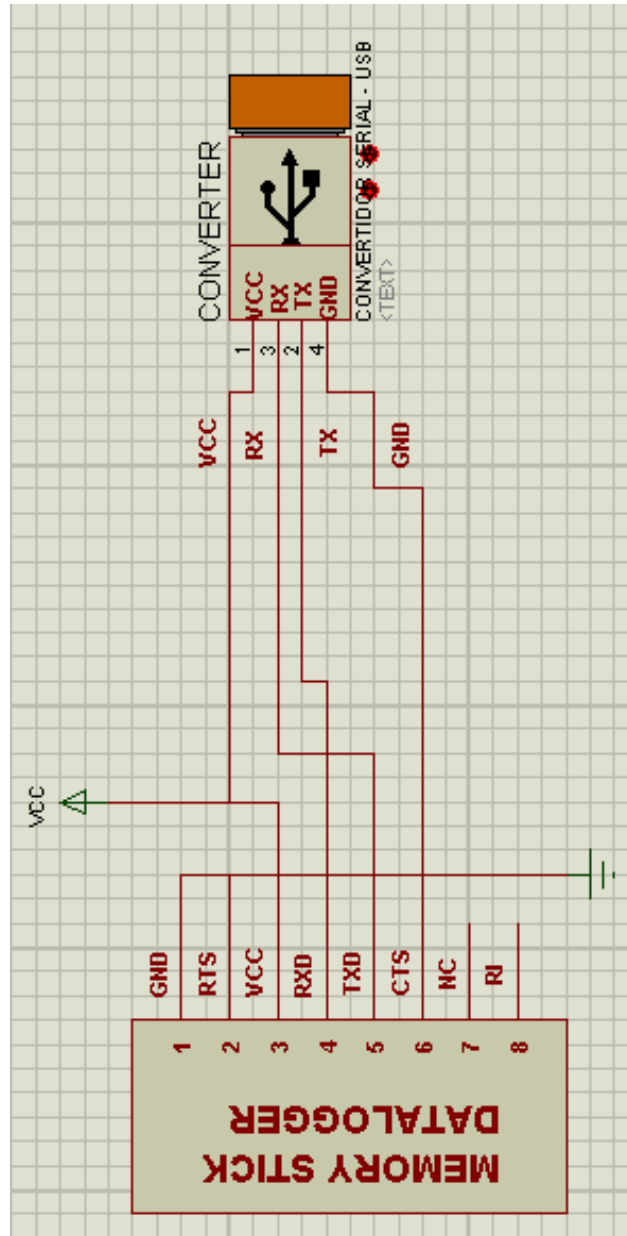


Figura 43 Pruebas de funcionamiento con el Datalogger
Fuente: Autores

Para comprobar el funcionamiento del memory stick datalogger, se tiene que establecer una comunicación serial (por medio del software Hyperterminal), siguiendo los siguientes pasos:

Generar una nueva conexión (Hyperterminal), elegir un nombre y un ícono cualesquiera, como indica la Figura 44.

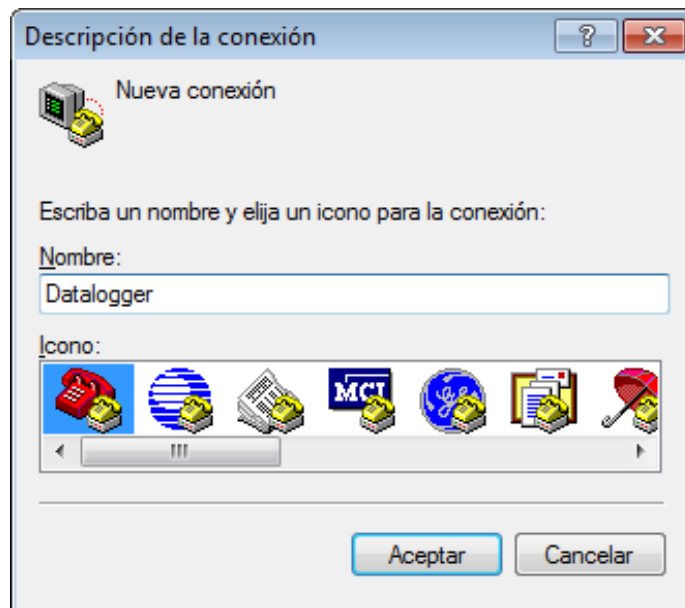


Figura 44 Ventana *Nueva Conexión Datalogger*
Fuente: Autores

Se escoge el puerto de comunicación, para nuestro ejemplo fue el COM4, tal como se observa en la imagen siguiente.

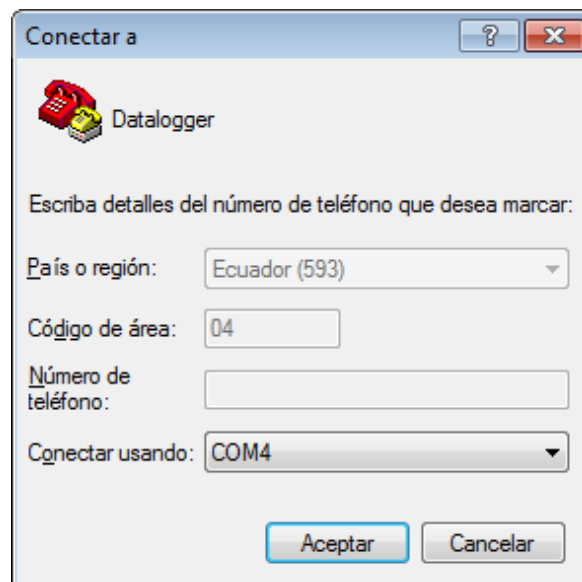


Figura 45 Ventana *Conectar a Datalogger*
Fuente: Autores

Se configura el puerto a los siguientes valores mostrados:

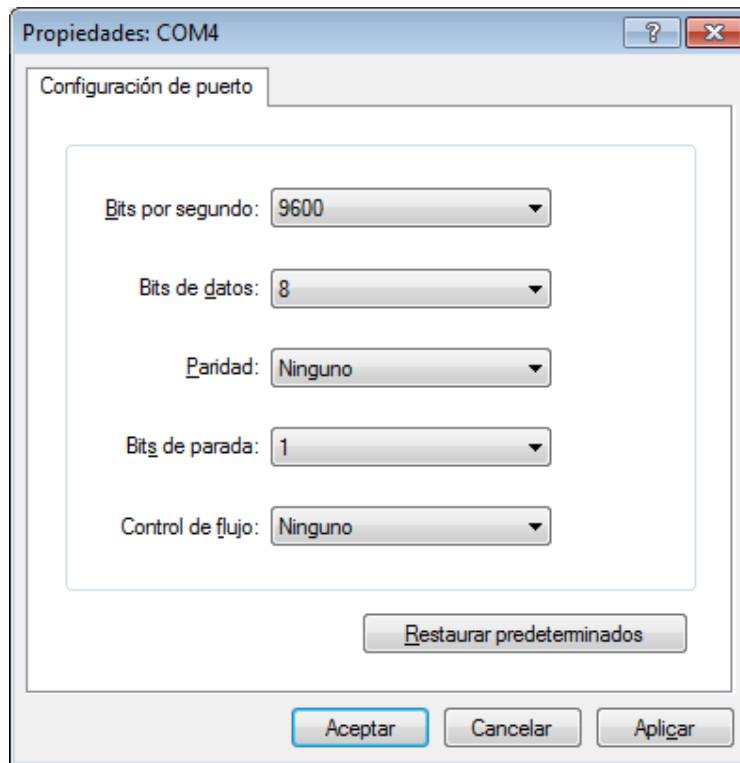


Figura 46 Ventana *Propiedades Comunicación del Datalogger*
Fuente: Autores

Después de configurar el puerto, nos presenta la siguiente ventana Figura 47, con el estado *Conectado*, nos dirigimos a la siguiente ruta *Archivo / Propiedades* para establecer configuraciones más avanzadas para este tipo de dispositivo.

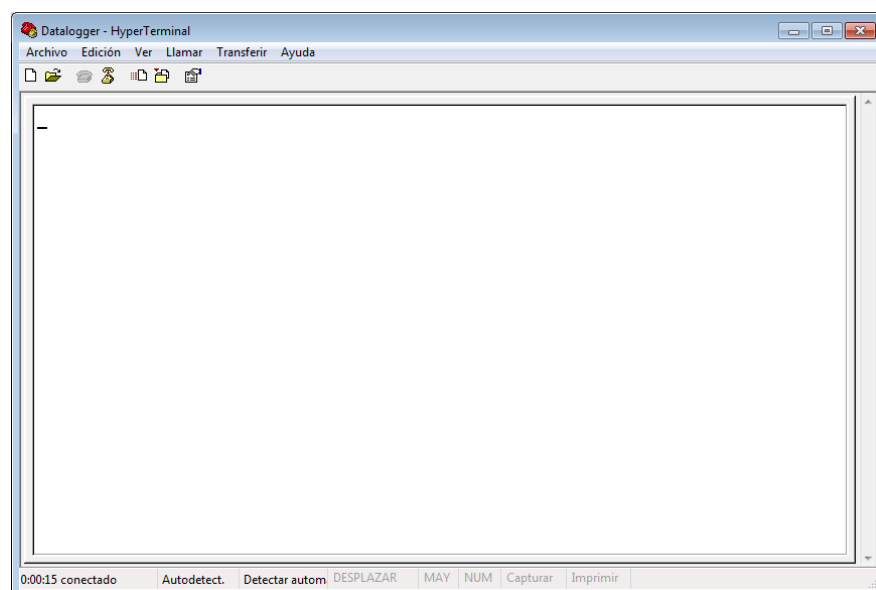


Figura 47 Ventana Hyperterminal Datalogger
Fuente: Autores

La nueva ventana se presenta a continuación y nos ubicamos en la pestaña de *Configuración*, como indica el gráfico de la Figura 48. Aquí seleccionamos *Configuración ASCII*.

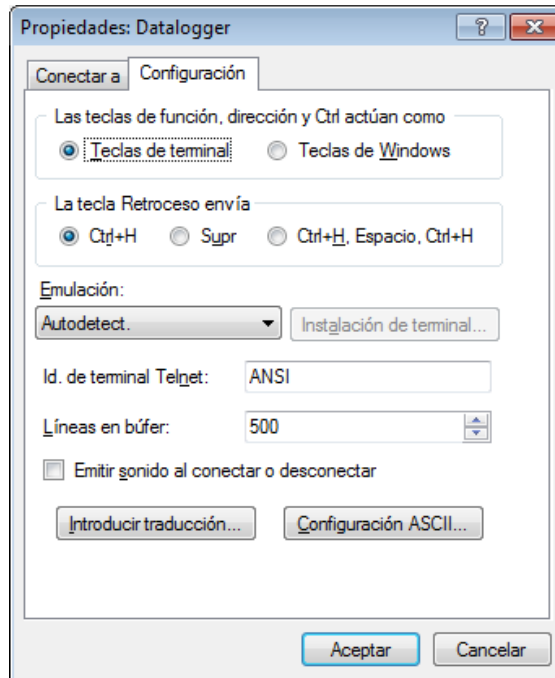


Figura 48 Ventana *Propiedades: Datalogger*

Fuente: Autores

En esta ventana de configuración ASCII, elegimos las opciones tal y como se muestra en la Figura 49. Después de esto se concluye con la parte de software.

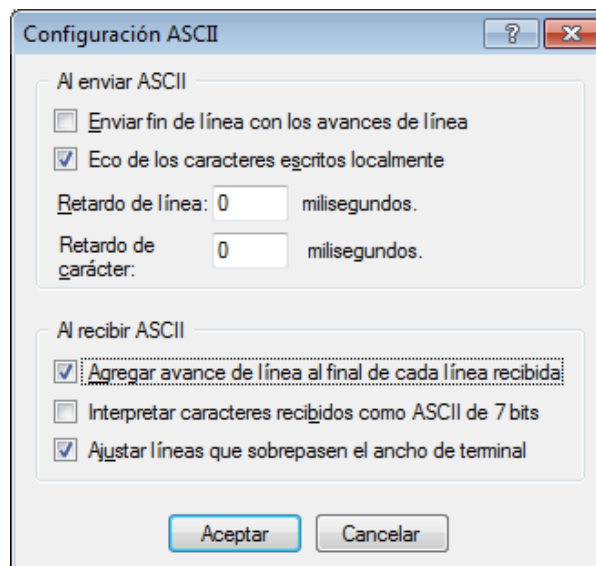
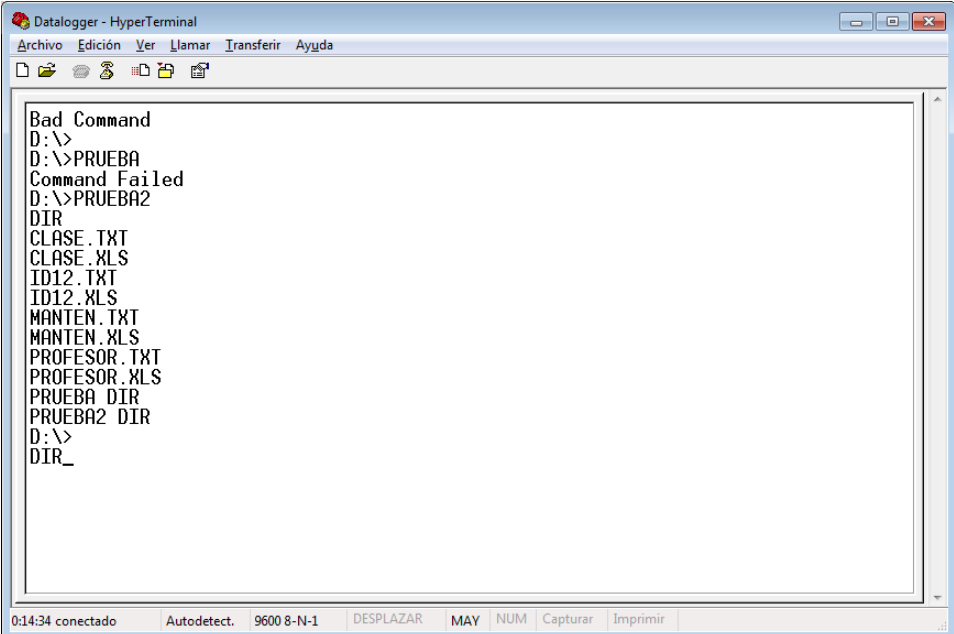


Figura 49 Ventana *Configuración ASCII: Datalogger*

Fuente: Autores

Relacionando la parte del hardware y software, se procede a interactuar con comandos establecidos y explicados en el capítulo 2, a continuación como ejemplo se establecerá comunicación y se creará una carpeta con nombre PRUEBA, esto se puede apreciar en la Figura 50.



```
Datalogger - HyperTerminal
Archivo  Edición  Ver  Llamar  Transferir  Ayuda

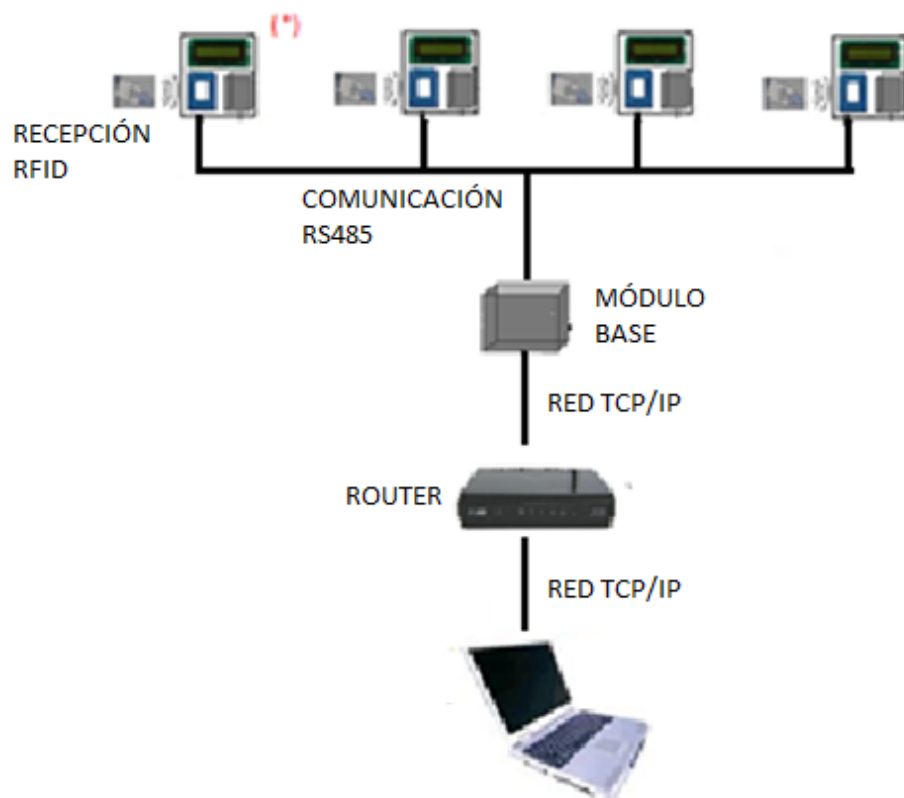
Bad Command
D:\>
D:\>PRUEBA
Command Failed
D:\>PRUEBA2
DIR
CLASE.TXT
CLASE.XLS
ID12.TXT
ID12.XLS
MANTEN.TXT
MANTEN.XLS
PROFESOR.TXT
PROFESOR.XLS
PRUEBA DIR
PRUEBA2 DIR
D:\>
DIR_

0:14:34 conectado  Autodetect.  9600 8-N-1  DESPLAZAR  MAY  NUM  Capturar  Imprimir
```

Figura 50 Ventana Hyperterminal Datalogger con datos ingresados al pendrive
Fuente: Autores

3.3 Esquema general del proyecto

A continuación se detalla de manera general el hardware del proyecto, podemos observar en la Figura 51, los módulos de registro de datos (4), el módulo base relacionados con la red RS485, así como también el router el cual se conecta a una red TCP/IP configurada en una laptop. En la siguiente sección se explica de manera detallada cada estructura del esquema general del proyecto.



(* En los módulos de registro de datos se presentan comunicación I2C (DS1307), Comunicación SPI (Memory Stick Datalogger)

Figura 51 Esquema general del proyecto
Fuente: Autores

3.4 Diagrama de bloques del proyecto

Después de haber descrito de manera general los integrados, en forma teórica (capítulo 2) y módulos que se van a utilizar para desarrollar el proyecto en la sección anterior, se puede establecer que el proyecto consta de dos partes denominados:

- Módulo Registro de Datos,
- Módulo Base.

A continuación un diagrama de bloques de cada una de las partes en que comprende el proyecto.

3.4.1 Estructura del “Módulo Registro de Datos”

Se puede observar en la Figura 52, que los módulos registro de datos tendrán un lector RFID ID12, capaz de recibir información de las tarjetas para que sean procesadas por el microcontrolador PIC16F877A, aquí se encuentra el programa principal de todo el proyecto, éste también interactúa con el DS1307 por medio de la comunicación I²C, para poder entregar y recibir la fecha y hora de marcación del personal, el LCD 16 X4 es relacionado también con este PIC, y envía la información al usuario. El otro PIC16F876 es el encargado de establecer comunicación con el módulo base ayudado por el SN75176. Como se puede observar el memory stick datalogger interactúa con los dos PIC, ya que recibe datos del PC y de las acciones externas (marcación de entrada), pero no al mismo tiempo.

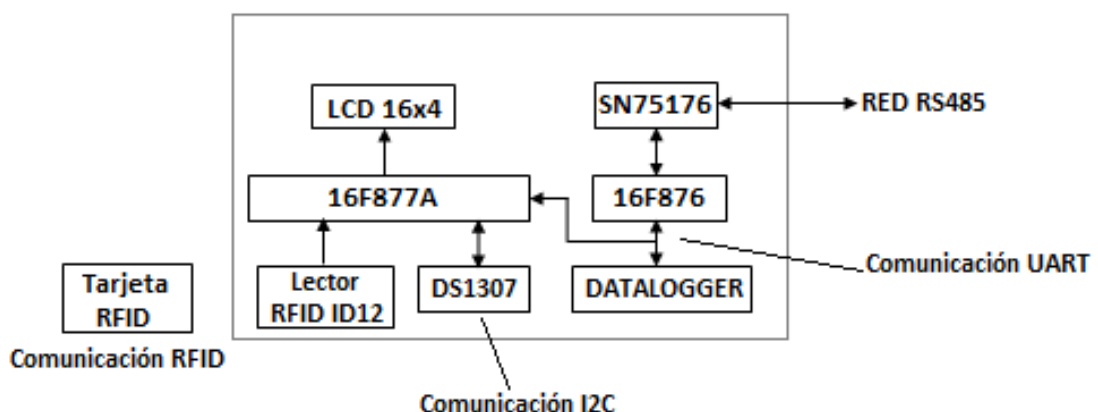


Figura 52 Diagrama de bloques del Módulo Registro de Datos

Fuente: Autores

A continuación se detalla de manera minuciosa la estructura interna del módulo registro de datos.

3.4.1.1 Conexiones DS1307

Entre el pin 1 y el pin 2 se coloca un cristal de 32876 KHz (cristal de cuarzo), en el pin 3 se ubica una pila de serie CR2032 (en la placa se coloca esta pila usando un socket para la batería), esto sirve para mantener los datos de fecha y hora cuando el módulo este desconectado. El pin 5 (SDA) y el pin 6 (SCL) son los pines encargados de la comunicación con el pic registro de datos usando el bus I²C explicado en capítulos anteriores, estos son conectados con resistencias de pull up de 10KΩ cada pin como indica la Figura 53.

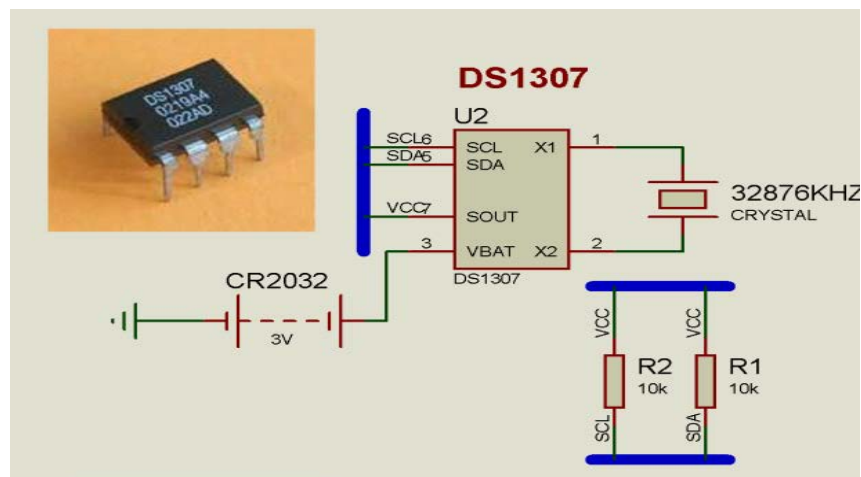


Figura 53 Conexiones DS1307
Fuente: Autores

3.4.1.2 Conexiones Sensor ID12

En el sensor ID12 (Figura 54), existen algunos pines que no se conectan entre ellos (pin 3, pin 4, pin 5, pin 6, pin 8). Los pines 1 y 7, son conectados a GND. Los pines 2 y 11 van conectados a VCC. El pin 9 entrega el código de la tarjeta RFID. El pin 10 activa un sonido cuando esto sucede. .

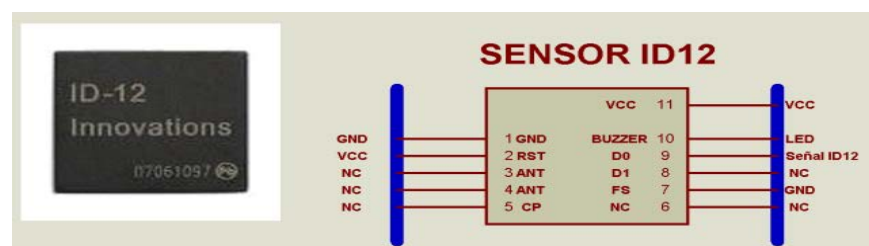


Figura 54 Conexiones Sensor ID12
Fuente: Autores

3.4.1.3 Conexiones Memory Stick Datalogger

Siguiendo con las partes del módulo registro de datos, se explica las conexiones del memory stick datalogger, los pines 1 y 3 son encargados de la alimentación GND y VCC respectivamente. Los pines 2, 7 y 8 son pines que no son usados en el proyecto, pero es preferible conectarlos a GND. El pin 6 es NC (No Conexión). Aquí los pines que se encargan de la comunicación con el pic registro de datos son los pines 4 (IN_LOGGER) y 5 (OUT_LOGGER).

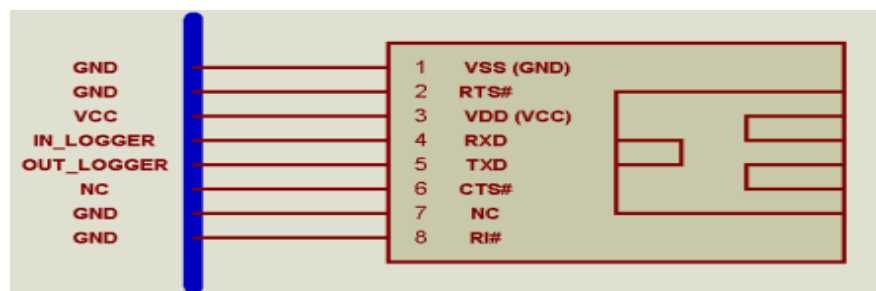


Figura 55 Conexiones Memory Stick Datalogger
Fuente: Autores

3.4.1.4 Conexiones LCD 20X4

El elemento más común del módulo es el LCD 20X4, se encarga de presentar los mensajes y ser intermediario entre el usuario y el proyecto, para su conexión se usan sólo 4 bits de datos (D4, D5, D6, D7) que corresponden a los pines 11, 12, 13, 14 respectivamente. Los pines de alimentación son el pin 1 (GND) y el pin 2 (VCC). Los pines 4 (RS), 5 (RW), 6 (E) son pines de control que se conectan al pic registro de datos.

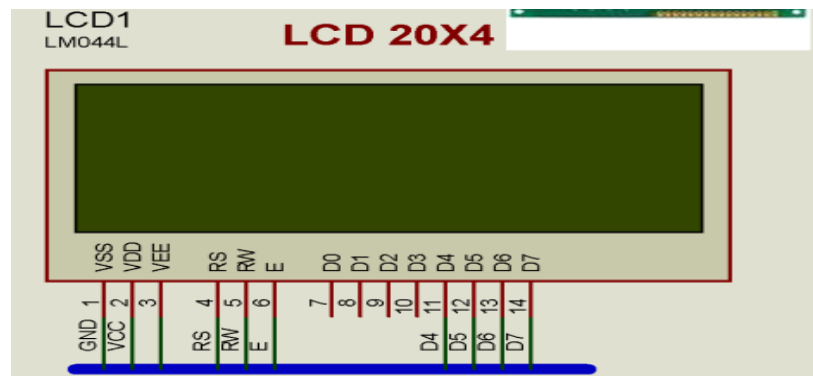


Figura 56 Conexiones LCD 20 X 4
Fuente: Autores

3.4.1.5 Conexiones PIC REGISTRO DE DATOS

Denominado como pic registro de datos, como su nombre lo indica es un PIC16F877A capaz de controlar todo el flujo de información ya que aquí se aloja el programa principal, desarrollado en lenguaje de alto nivel PIC BASIC explicado en el capítulo siguiente.

Aquí se anotan conexiones importantes del PIC en mención:

Pines de comunicación con LCD 20 X4	Pines 27(D4), 28(D5), 29(D6), 30(D7)
Pines de habilitación y RS del LCD 20 X4	Pines 19(D0), 20(D1)
Pines de comunicación con DS1307	Pines 18 (SCL), 23 (SDA)
Pines de aviso con PIC16F876	Pines 17 (SCL2), 33 (SDA2)
Pines de comunicación con sensor RFID	Pin 21 (Señal ID12)
Pines de comunicación con Memory Stick	Pines 15 (C0), 22 (D3)
Pines habilitación circuito fuerza	Pin 16 (C1)
Pines X1 – X2 (Cristal de cuarzo)	Pines 13 (OSC1), 14 (OSC2)

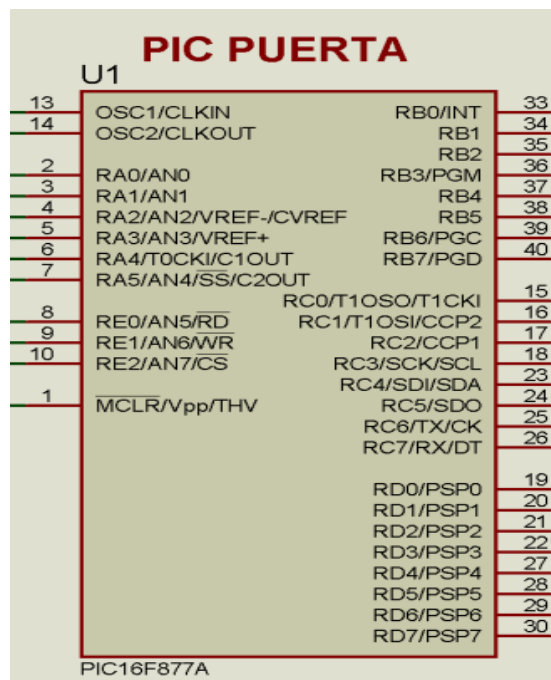


Figura 57 Conexiones pic Registro de Datos (PIC16F877A)

Fuente: Autores

3.4.1.6 Conexión Pulsante Interno

Otro de los elementos internos es un pulsante interno como indica la Figura 58, que sirve para activar la etapa de potencia (el cual en el proyecto activar un relé, que energizara una chapa magnética) desde el interior de la puerta, en cualquier momento.



Figura 58 Conexiones Pulsante Interno
Fuente: Autores

3.4.1.7 Conexiones SN75176

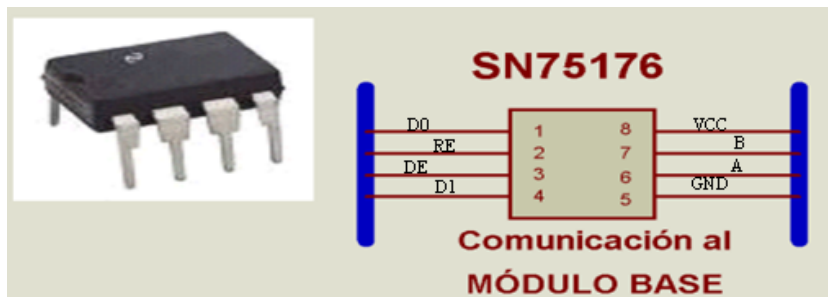


Figura 59 Conexiones SN75176
Fuente: Autores

El integrado capaz de enlazar la red entre los módulos de registro de datos y el módulo base es el SN75176.

Este se encarga del hardware de la red RS485, todos los módulos registro de datos y el módulo base están conectados en paralelo, cuando se envía información para que un módulo registro de datos específico, éste es enviado en sí para todos los módulos, pero sólo responde uno, ya que cada módulo cuenta con una dirección diferentes entre ellos.

3.4.1.8 Conexiones Circuito de Potencia

En la etapa de potencia interviene un relé el cual a su vez puede accionar un elemento de fuerza (chapas magnéticas), con lo cual abrirían las puertas para permitir el acceso.



Figura 60 Conexiones circuito de potencia
Fuente: Autores

3.4.2 Esquema General de Módulo Registro de Datos

Se realiza un esquema general de los componentes del “Módulo Registro de Datos” para tener una idea más clara de los elementos usados, esto se puede observar en la Figura 61.

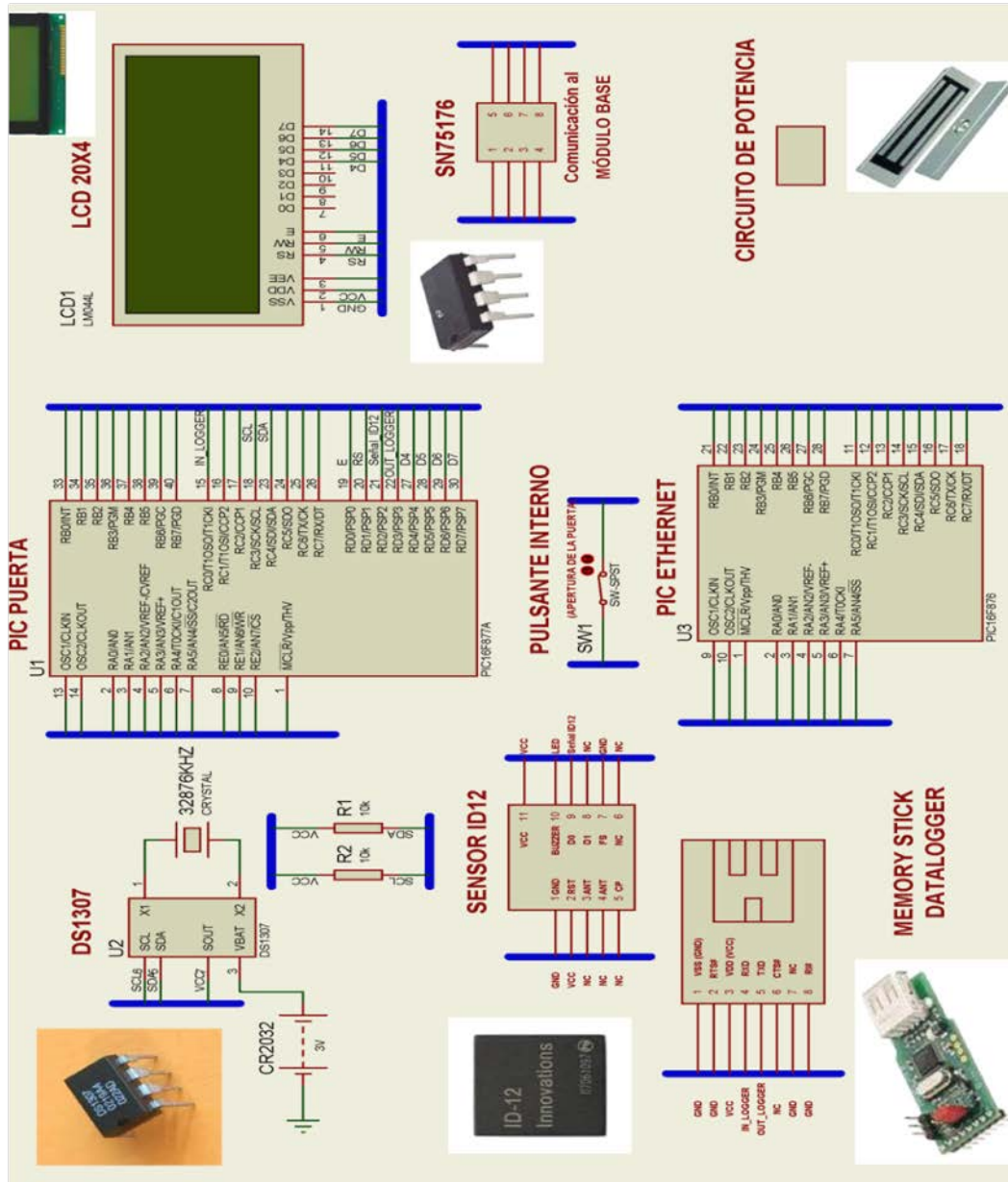


Figura 61 Esquema General del Módulo Registro de Datos / Año 2012

Fuente: Autores

3.4.3 Estructura del “Módulo Base”

En este módulo se marcan dos componentes claves como son el PIC16F876 y el convertidor serial (RS232 – Ethernet). El PIC se encarga de controlar el flujo de datos de la comunicación entre el PC y los módulos registro de datos, el convertidor serial se encarga de enviar y recibir datos para la red Ethernet. Esto se muestra en el diagrama de bloques adjunto en la Figura 62.

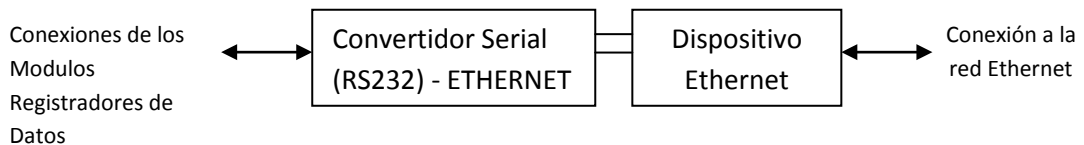


Figura 62 Diagrama de bloques del Módulo Base
Fuente: Autores

3.4.3.1 Conexiones Convertidor Serial (RS232 – Ethernet)

Aquí se utilizan el convertidor serial – Ethernet de la empresa Wiznet, el modelo utilizado es el WIZ110SR, cuya transmisión serial es a 9600 baudios, con 8 bits de datos, 1 bit de stop, sin control de flujo, este posee la dirección IP 192.168.1.105, la cual es configurable por medio del Wiznet Tools.

La alimentación de este módulo es +5 VDC, y se encuentra relacionado con el router DLINK, la cual se encarga de proveer el estándar Ethernet.

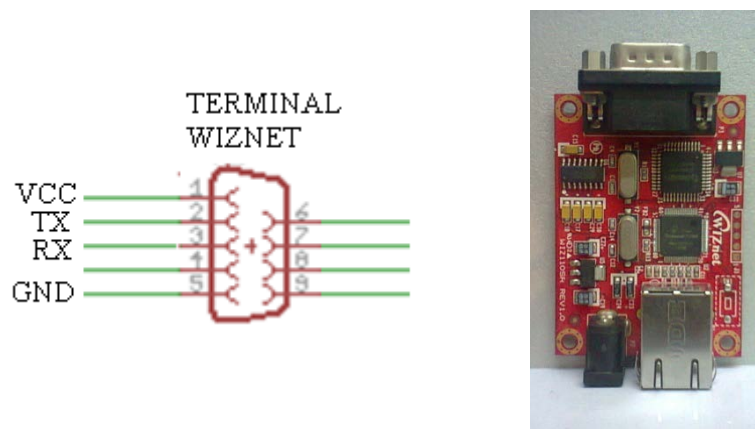


Figura 63 Conexiones Wiznet
Fuente: Autores

3.4.3.2 Conexiones PIC Módulo BASE

Denominado como PIC Módulo BASE, como su nombre lo indica es un PIC16F876 capaz de controlar todo el flujo de información dentro de la red con ayuda del WIZNET (Convertidor Serial – Ethernet), su programación fue realizada en un lenguaje de alto nivel PICBASIC y se detallo en capítulos anteriores.

Aquí se anotan conexiones importantes del PIC en mención:

Pines X1 – X2 (Cristal de cuarzo) Pines 13 (OSC1), 14 (OSC2)

Pines de comunicación para SN75176
(Red RS485)

Pines 15 (C0), 16 (C1), 17 (C2)

Pines de comunicación para WIZNET

Pines 33 (B0), 16 (C1)

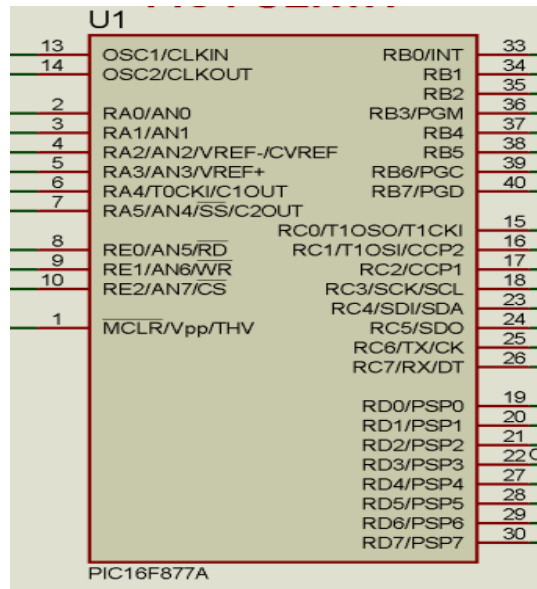


Figura 64 Conexiones PIC Módulo Base

Fuente: Autores

3.4.3.3 Conexiones SN75176

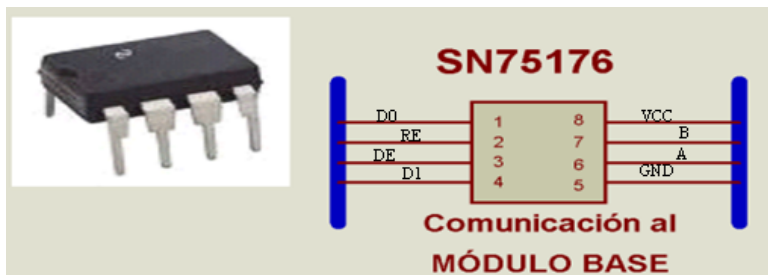


Figura 65 Estructura SN75176

Fuente: Autores

El integrado capaz de enlazar la red entre los módulos registro de datos y el módulo base es el SN75176.

Este se encarga del hardware de la red RS485, por lo tanto con la ayuda de este integrado todos los módulos registro de datos y el módulo base están conectados en paralelo. Cuando se envía información para que lea un módulo registro de datos, es enviado en sí para todos los módulos, pero sólo responde uno, ya que cada módulo cuenta con direcciones diferentes entre ellos.

3.4.4 Esquema General del Módulo Base

Se realiza un esquema general de los componentes del “Módulo Base” para tener una idea más clara de los elementos usados, esto se puede observar en la Figura 66.

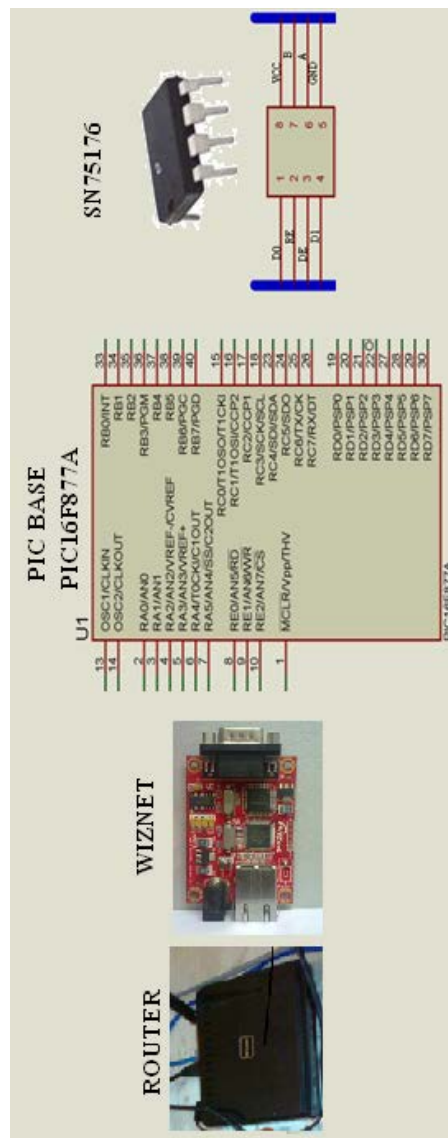


Figura 66 Esquema General del Módulo Base
Fuente: Autores

3.5 Pruebas en Protoboard

Diferentes gráficos presentan el desarrollo del proyecto, desde las fases experimentales en Protoboard hasta llegar a las placas electrónicas en la sección siguiente.

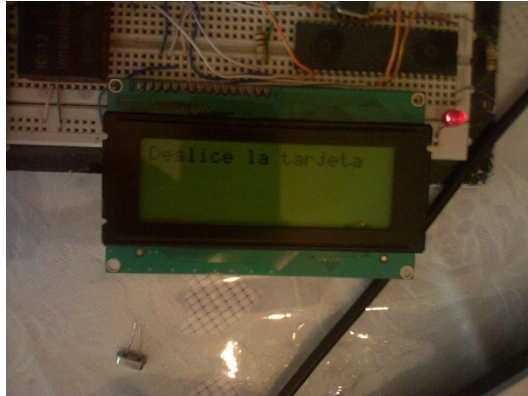


Figura 67 Pruebas en Protoboard (LCD 20 X4)
Fuente: Autores

En la Figura 67, se observa el Display LCD 20 X 4 que presenta el mensaje “Deslice la tarjeta”, indicando que está lista para recibir datos de la tarjeta RFID y seguir procesando estos datos como indica el programa interno.

Los nuevos elementos en pleno funcionamiento, después de realizar las configuraciones explicadas en la sección 3.1 y 3.2 del sensor ID12 y el memory stick datalogger respectivamente.

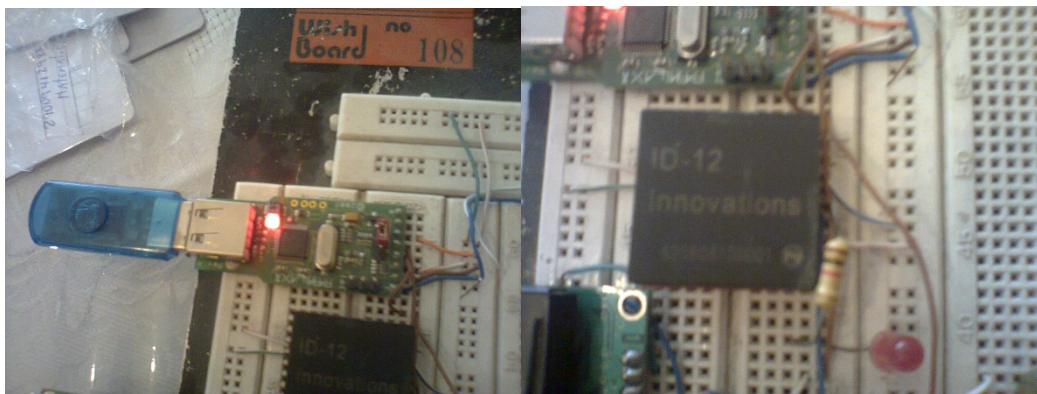


Figura 68 Pruebas en Protoboard (Memory Stick y Sensor ID12)
Fuente: Autores

Aquí se observa un módulo registro de datos, en las cuales se hacen pruebas de comunicación con la laptop, con sus respectivos cables, para luego de esto hacer pruebas inalámbricas con ayuda del router.

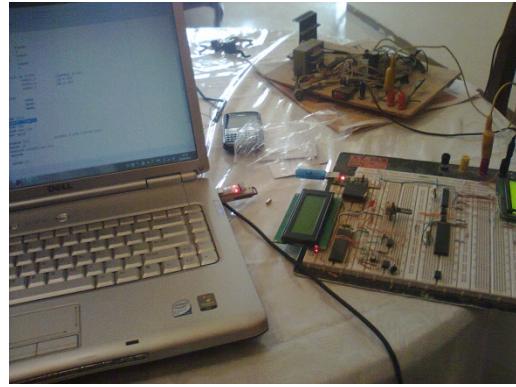


Figura 69 Pruebas en Protoboard (Comunicación con laptop)

Fuente: Autores

Se armaron varios módulos para realizar distintas pruebas en tiempo real, y observar el desarrollo del proyecto hasta que no haya errores en la transmisión de datos.

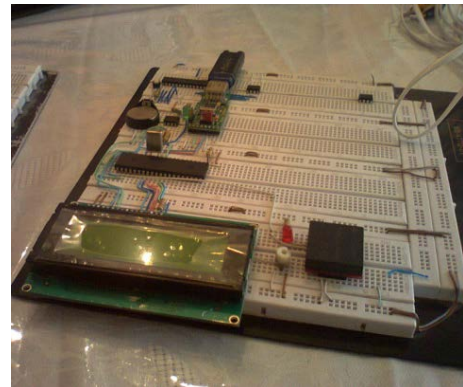


Figura 70 Pruebas en Protoboard (Módulos Registro de Datos)

Fuente: Autores

3.6 Diseño de placas electrónicas.

En electrónica, un **circuito impreso** o **PCB** (del inglés *printed circuit board*), es un medio para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de *rut*as o *pistas* de material conductor, grabados en hojas de cobre laminado sobre un sustrato no conductor, comúnmente baquelita o fibra de vidrio.

“Los circuitos impresos son baratos, y habitualmente de una fiabilidad elevada aunque de vez en cuando pueda tener fallos técnicos. Requieren de un esfuerzo mayor para el posicionamiento de los componentes, y tienen un costo inicial más alto que otras alternativas de montaje, como el montaje *punto a punto* (o *wire-wrap*), pero son mucho más baratos, rápidos y consistentes en producción en volúmenes.”⁶

Existen diferentes tipos de circuitos impresos que se pueden desarrollar, que son:

- ✓ Multicapa: Es lo más habitual en productos comerciales. Suele tener entre 8 y 10 capas, de las cuales algunas están enterradas en el sustrato.
- ✓ 2-sided plated holes: Es un diseño muy complicado de bajo coste con taladros metalizados que nos permite hacer pasos de cara.
- ✓ Single-sided non-plated holes: Es un PCB con agujeros sin metalizar. Se usa en diseños de bajo coste y sencillos.
- ✓ 2-sided non-plated holes: Diseño sencillo con taladros sin metalizar. Sustrato de fibras de vidrio y resina. Hay que soldar por los dos lados para que haya continuidad

Para el desarrollo de este proyecto se ha escogido el software EAGLE el cual es un acrónimo **E**asily **A**pplicable **G**raphical **L**ayout **E**ditor que literalmente significa editor de diseño gráfico fácilmente aplicable.

El programa consta de tres módulos, un Diagramador, un Editor de esquemas y un Autorouter que están integrados por lo que no hay necesidad de convertir los Netlist

6 http://es.wikipedia.org/wiki/Circuito_impreso / **Autor:** Wikimedia Project / **Año:** 2012

entre esquemas y diseños. Es una potente aplicación con la que diseñar circuitos impresos y realizar esquemas electrónicos se la realiza de una manera amigable y fácil.

“Gracias a este editor se logra diseñar esquemas y placas de circuito impreso con auto router, es decir con la función que automatiza el dibujo de pistas en la placa de circuitos impresos, y todo esto en un entorno ergonómico.”⁷

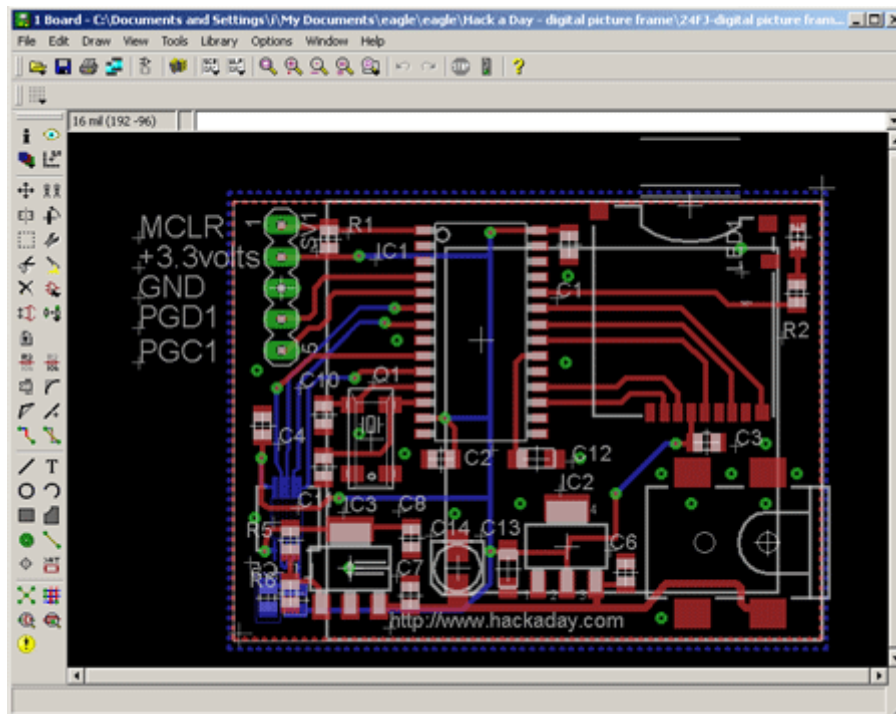


Figura 71 Ventana PCB del software EAGLE
Fuente: <http://www.electronics-lab.com/blog/?tag=eagle&paged=2>

7 <http://www.taringa.net/posts/info/11037233/Software-de-Electronica.html/>
Autor: IntelligentColective / **Año:** 2012

3.6.1 Diseño esquemático. Módulo Registro de Datos

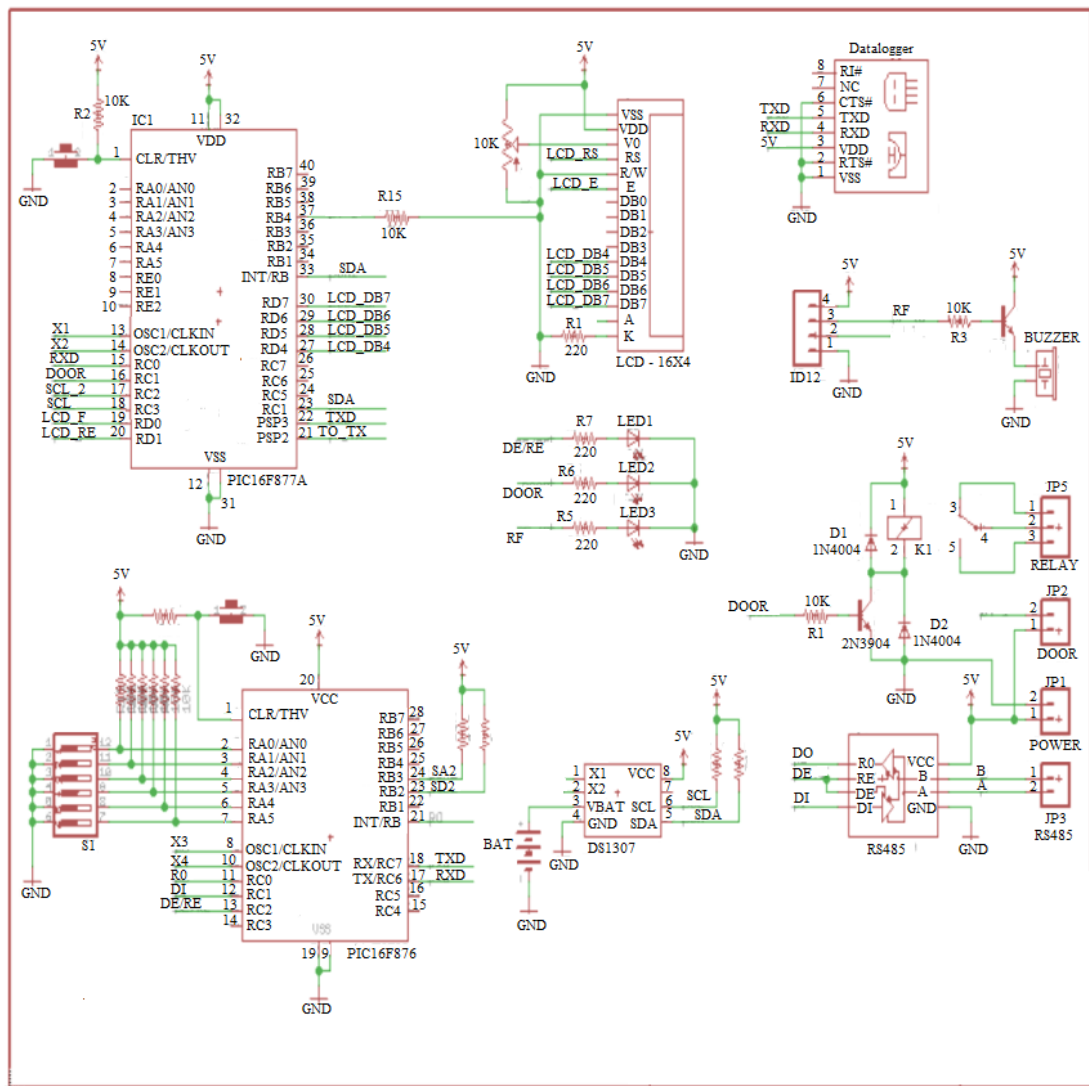


Figura 72 Diseño Esquemático. Módulo Registro de Datos / Año 2012

Fuente: Autores

3.6.2 Diseño elaborado en Eagle. Módulo Registro de Datos

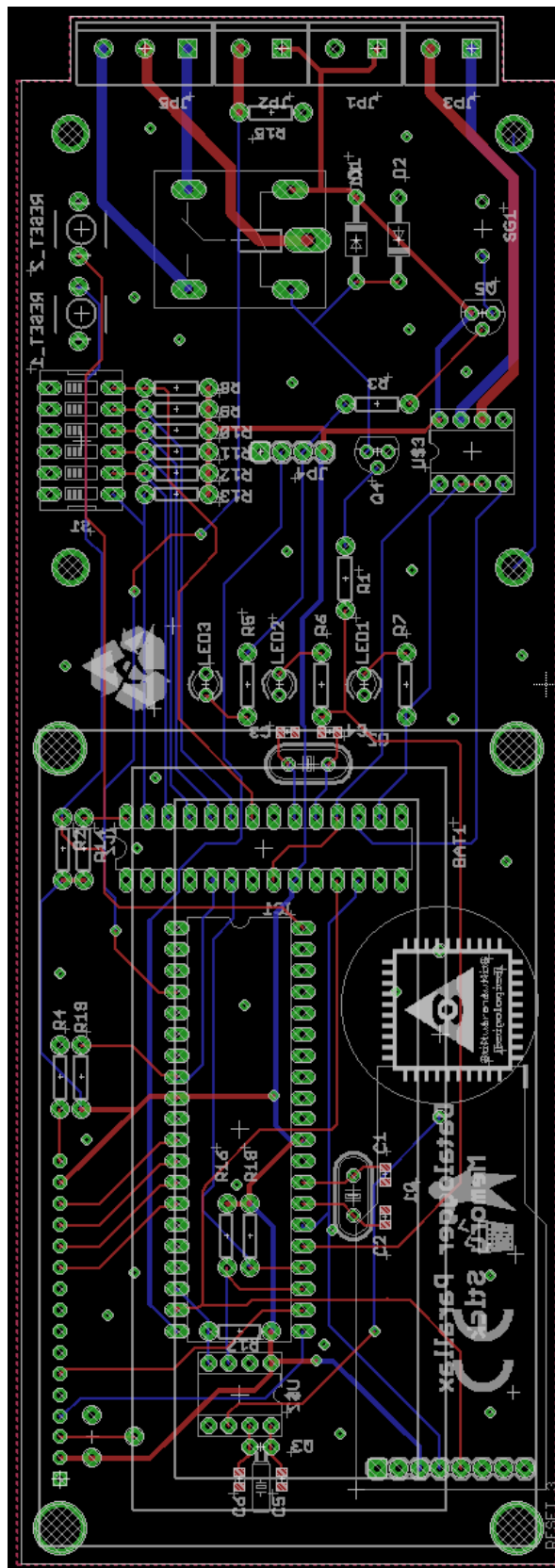


Figura 73 Diseño elaborado en Eagle. Módulo Registro de Datos / Año 2012

Fuente: Autores

3.6.3 Diseños PCB de Módulo Registro de Datos

Nuestro proyecto fue diseñado en el software EAGLE, el cual posee sus librerías propias, y se encuentra fabricada en fibra de vidrio para mejorar la durabilidad de la tarjeta (problemas de contactos, humedad, etc.).

La placa del módulo registro de datos debe alojar todos los elementos tales como: LCD 20X4, Sensor ID12, Memory Stick Datalogger, PIC16F877A, PIC16F876, SN75176, donde vemos en la cara frontal en la Figura 74, que se puede apreciar el espacio para el LCD 20X4 y el memory stick datalogger, es aquí que se determina que la distancia entre el LCD y la placa base debe ser de 2cm. aprox. Para no intervenir con el espacio del memory stick y poder insertar de una manera más fácil el pendrive.

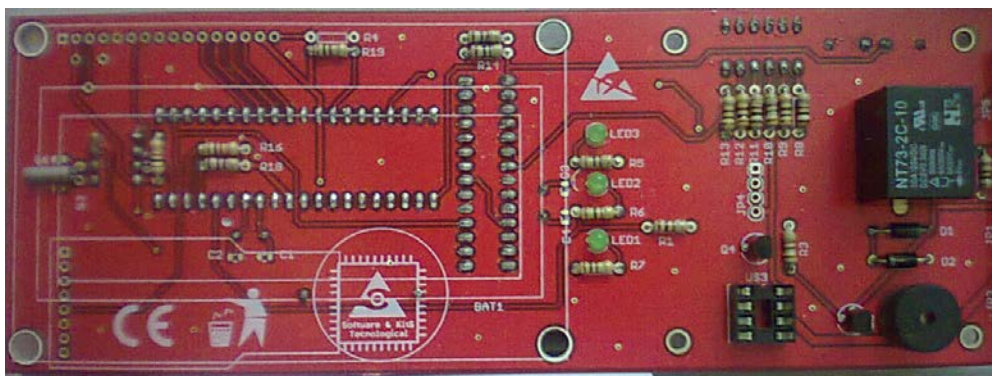


Figura 74 Diseño PCB Módulo Registro de Datos (1)

Fuente: Autores

En la Figura 75, se muestra el espacio requerido, cabe indicar que los conductores para el LCD 20X4 son sólo patillas de diodos rectificadores ya que son muy estables y un poco difíciles de deformar y así evitar malos contactos con otros elementos.

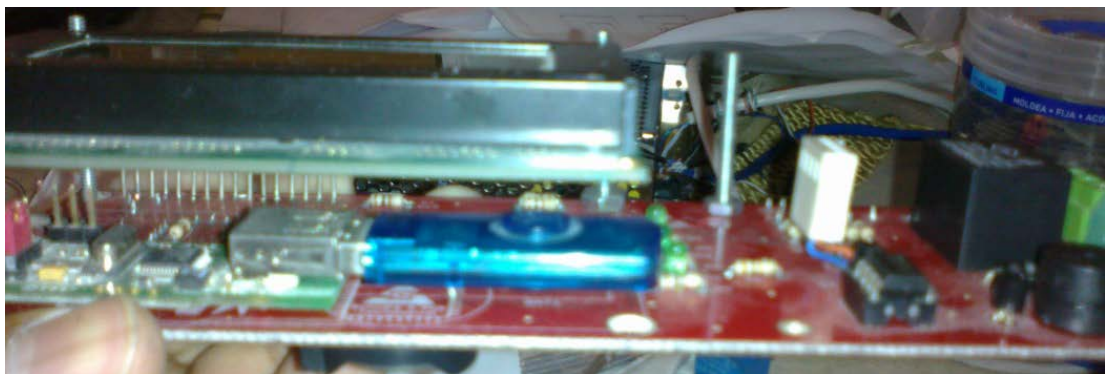


Figura 75 Diseño PCB Módulo Registro de Datos (2)

Fuente: Autores

Aquí se visualiza la vista frontal y posterior de la tarjeta, con algunos elementos soldados, se puede notar LCD 20X4, la etapa de potencia (relé), conectores de entrada y salida de datos ubicados en la parte derecha de la tarjeta electrónica. Según se muestra en la Figura 76 y Figura 77.

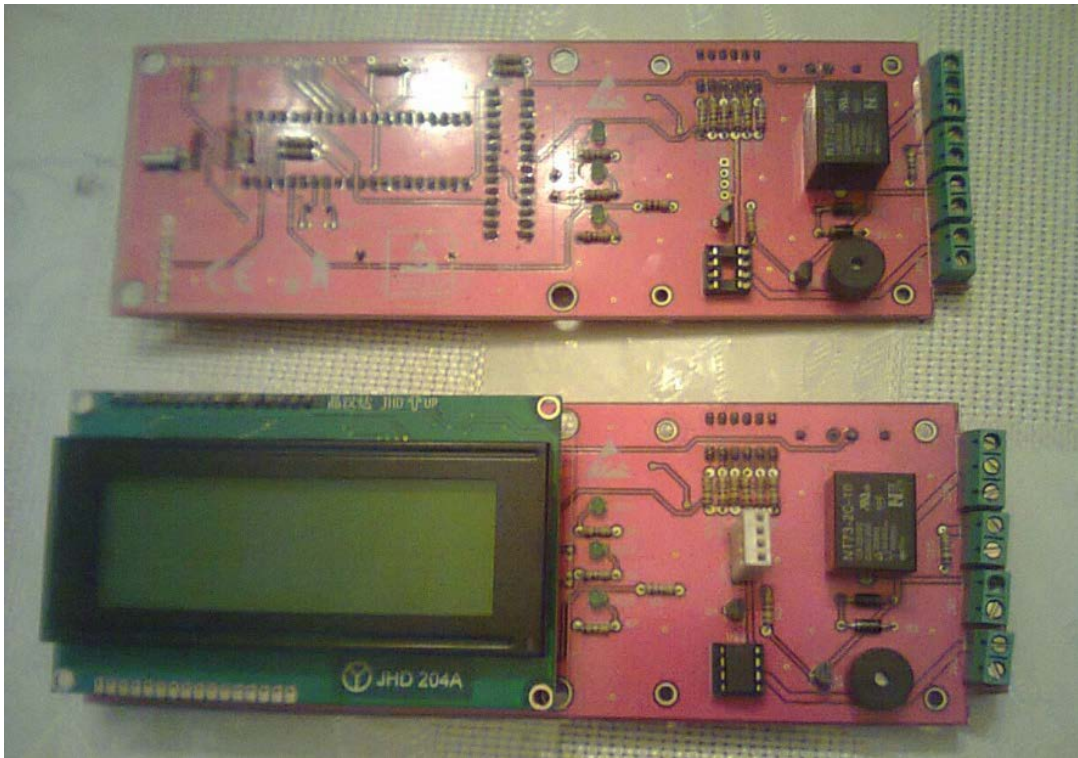


Figura 76 Diseño PCB Modulo Registro de Datos (3)
Fuente: Autores

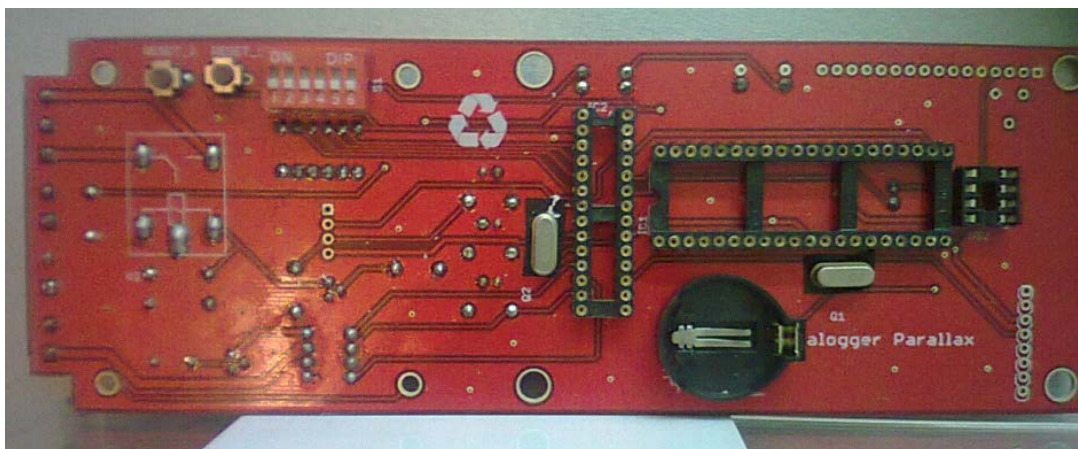


Figura 77 Diseño PCB Módulo Registro de Datos (4)
Fuente: Autores

3.6.4 Diseño esquemático. Módulo Base

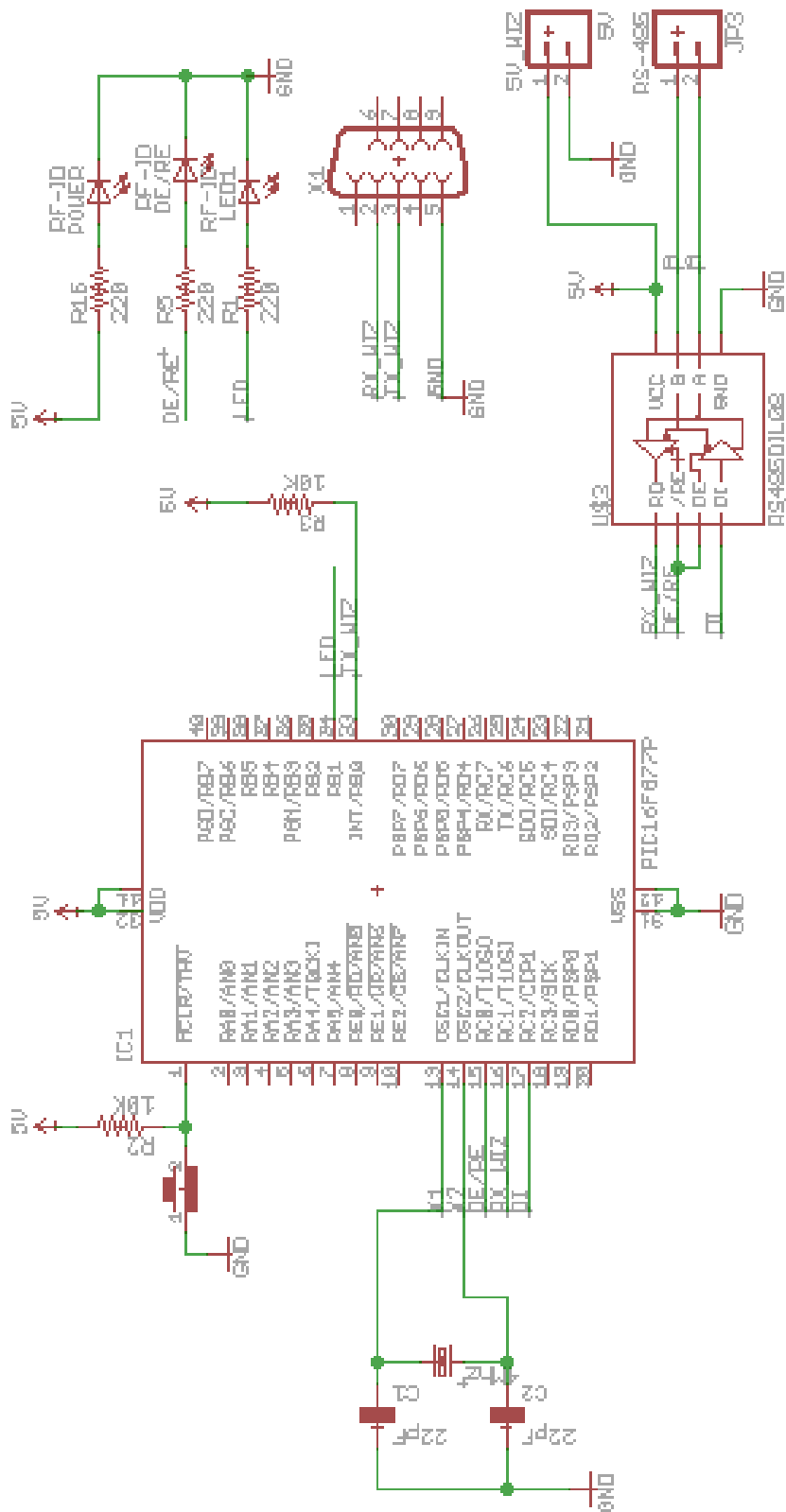


Figura 78 Diseño esquemático. Módulo Base. / Año 2012

Fuente: Autores

3.6.5 Diseño elaborado en Eagle. Módulo Base

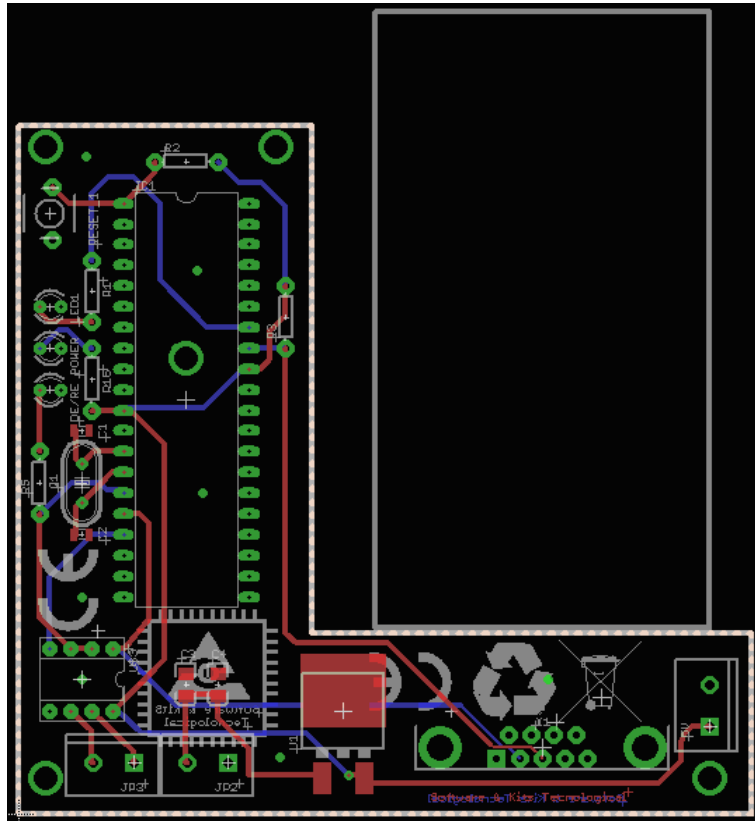


Figura 79 Diseño elaborado en Eagle. Módulo Base
Fuente: Autores

3.6.6 Diseños PCB de Módulo Base

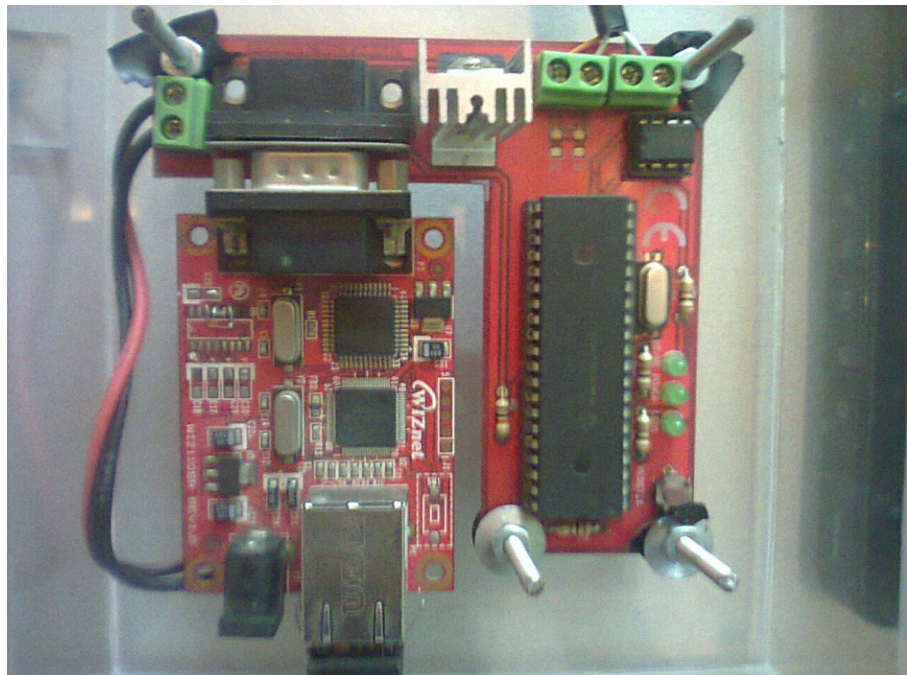


Figura 80 Diseño PCB Módulo Base
Fuente: Autores

3.6.7 Diseño esquemático. Módulo RFID

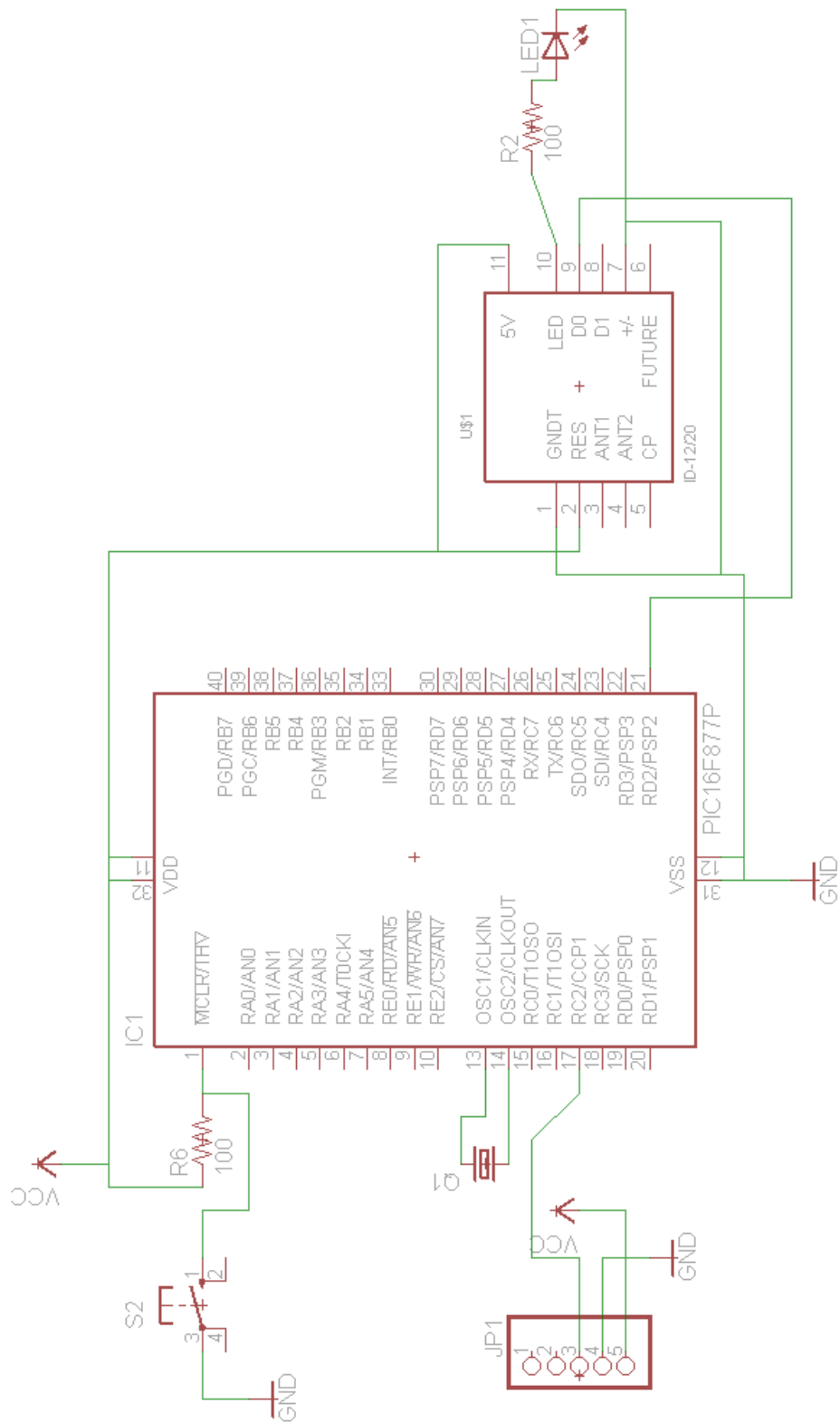


Figura 81 Diseño esquemático. Módulo RFID. / Año 2012
Fuente: Autores

3.6.8 Diseño elaborado en Eagle. Módulo RFID

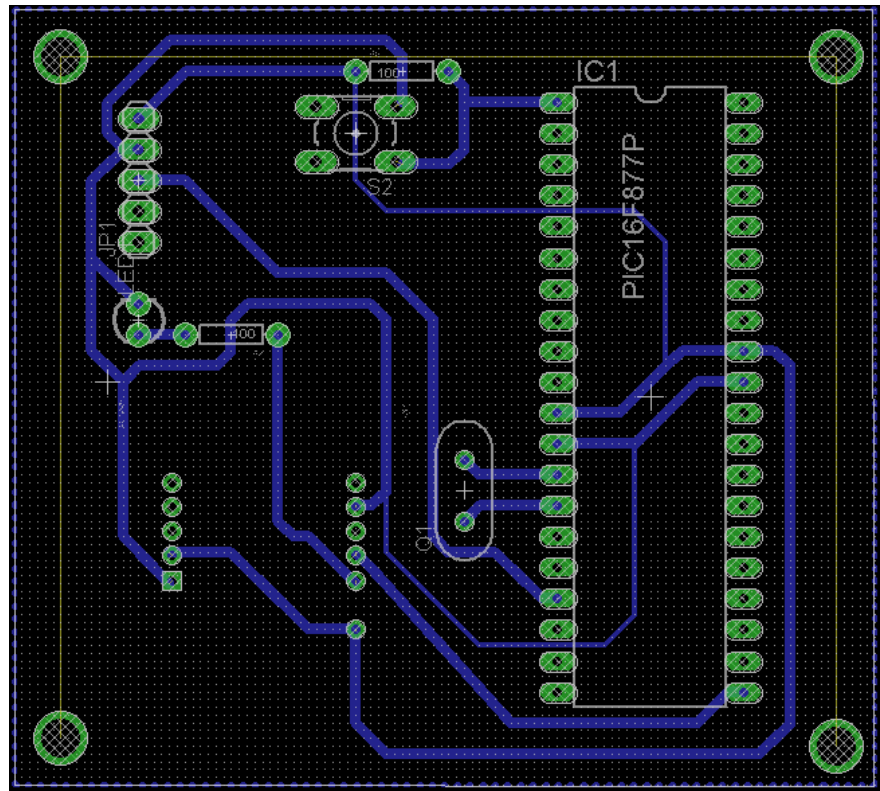


Figura 82 Diseño elaborado en Eagle. Módulo RFID

Fuente: Autores

3.6.9 Diseños PCB de Módulo RFID



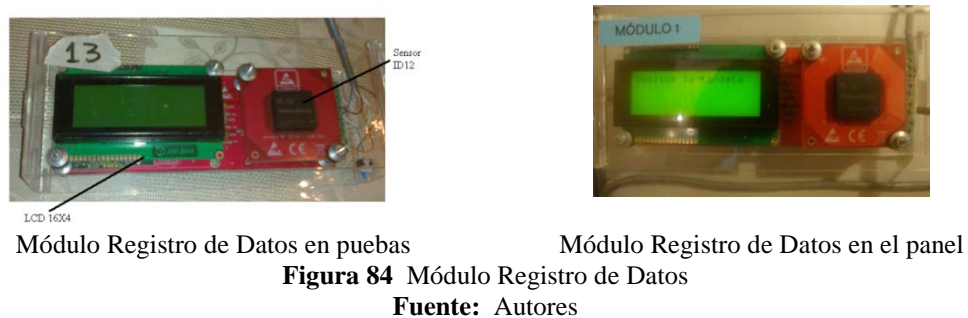
Figura 83 Diseño PCB Módulo RFID

Fuente: Autores

3.7 Funcionamiento y pruebas generales del proyecto

Las pruebas en primera instancia fueron realizadas en protoboard y se realizó con dos módulos, a medida que se fue desarrollando el proyecto, se obtuvo los resultados de las tarjetas impresas las cuales con todos sus elementos se muestran a continuación.

Se presenta uno de los módulos registro de datos, que se usó para realizar pruebas. Figura 84.



Dentro de cada módulo registro de datos hay una bornera que sirve de interfaz para comunicarse entre ellas y con el módulo base. Observar Figura 85.

- ✓ Recordar que todos los módulos registro de datos se conectan en una red RS485, para esto se usa los pines de señal de transmisión y recepción de datos.
- ✓ La entrada de alimentación para hacer funcionar el módulo.
- ✓ Los pines que van al pulsante interno (para la apertura de la puerta desde el interior).
- ✓ Los pines para habilitar o deshabilitar la chapa magnética.

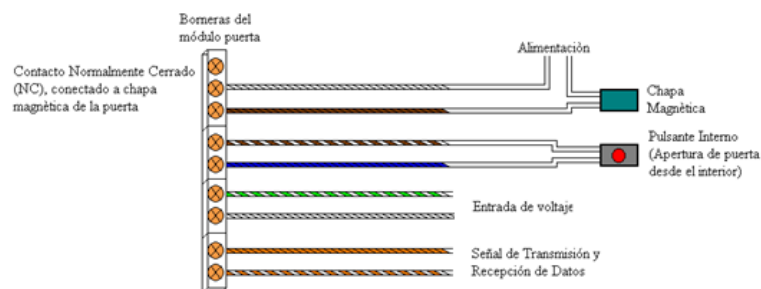
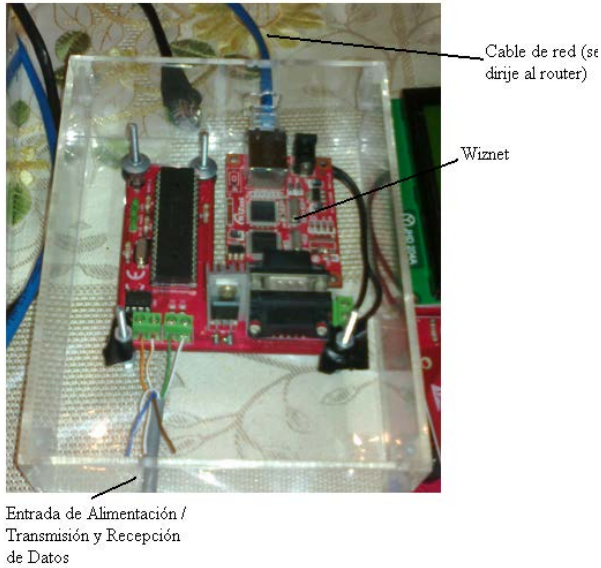


Figura 85 Diagrama de conexiones del Módulo Registro de Datos
Fuente: Autores

En la figura 86, se presenta el módulo base el cual relaciona todos los módulos registro de datos y establece comunicación con el software Cardget; esto por medio del router, en la Figura 87.



Módulo Base en pruebas

Módulo Base en el panel

Figura 86 Módulo Base
Fuente: Autores



Router en pruebas

Router en el panel

Figura 87 Router
Fuente: Autores

Las fuentes de alimentación son de 120 V de entrada a 2 Amperios con 5 Voltios de salida. Las fotos adjuntas demuestran este dispositivo.



Fuente de alimentación en pruebas



Fuente de alimentación en el panel

Figura 88 Fuente de alimentación

Fuente: Autores

Los pulsantes internos que activarán la chapa magnética desde el interior del lugar a controlar, se presentan en la Figura 89.



Figura 89 Pulsantes

Fuente: Autores

En la Figura 90, se observa el módulo RFID el cual sirve para ingresar nuevo personal que tendrá acceso a los diferentes puntos a controlar con el sistema.



Figura 90 Módulo RFID

Fuente: Autores

Aquí en la Figura 91, se presenta el conjunto de todos los dispositivos que se usan en el proyecto, en el cual podemos acotar:

- ✓ Módulos registro de datos(4 u.)
- ✓ Módulo base (1 u.)
- ✓ Router (1 u.)
- ✓ Fuente de alimentación (1 u.)
- ✓ Laptop (1 u.)



Figura 91 Elementos de Control de Acceso
Fuente: Autores

A continuación se muestran fotos del proyecto terminado, con sus respectivas puertas a escala, todos los módulos, están ubicados en cada una de las puertas, determinando como indica el tema un prototipo de control de acceso.

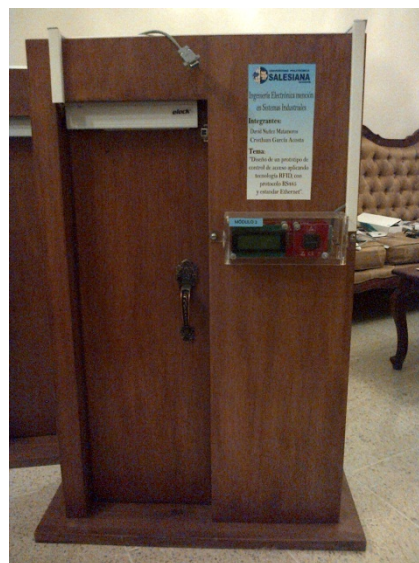


Figura 92 Prototipo de Control de Acceso (1)
Fuente: Autores

En una maqueta se muestran las fuentes de poder, el router y el módulo base, necesarios para completar el proyecto del prototipo de control de acceso.



Figura 93 Prototipo de Control de Acceso (2)
Fuente: Autores

Finalmente se presenta las cuatro puertas con sus respectivos acabados.

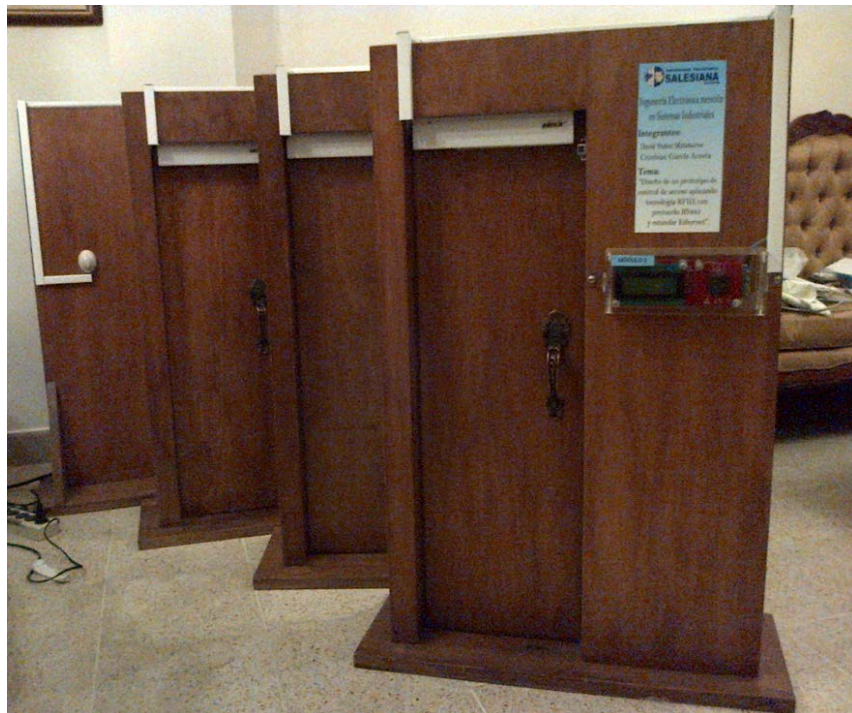


Figura 94 Prototipo de Control de Acceso (3)
Fuente: Autores

CAPÍTULO 4

PROGRAMACIÓN Y SOFTWARE DEL SISTEMA

En este capítulo se explicarán las líneas de programación elaboradas en lenguaje de alto nivel PICBASIC así como también un manual de usuario del software Cardget, para relacionarlo con todo el proyecto.

4.1 Programación del “Módulo Registro de Datos”

En esta sección se hace referencia al programa insertado en el PIC16F877A del Módulo Registro de Datos, como se indicó en capítulos anteriores aquí se encuentra el conjunto de pasos necesarios para desarrollar el proyecto.

```
*****
'* Name : MOD_REGISTRO.BAS *
'* Author : DAVID NUNEZ Y CRISTHIAN GARCIA *
'* Notice : CONTROL DE ACCESO Y MARCACIÓN DE ENTRADA PARA *
'* : PERSONAL CON TARJETA RFID Y COMUNICACIÓN *
'* : INALÁMBRICA *
'* Notes : ESTE PROGRAMA SE ENCUENTRA UBICADO EN EL PIC *
'* : PRINCIPAL DEL MODULO REGISTRO DE DATOS *
*****
```

: Definición del Ucontrolador

```
@DEVICEXT_OSC
```

```
INCLUDE "MODEDEFS.BAS"; INCLUYE LOS MODOS DE COMUNICACION
Define osc 4
```

: DEFINICION DE I2C

```
Define I2C_SCLOUT 1
```

: DEFINICION DE LCD

```
Define LCD_DREG PORTD
Define LCD_DBIT 4
Define LCD_RSREG PORTD
Define LCD_RSBIT 1
Define LCD_EREG PORTD
Define LCD_EBIT 0
DEFINE LCD_BITS 4
DEFINE LCD_LINES 4
DEFINE LCD_COMMANDUS 2000
DEFINE LCD_DATAUS 50
```

: Variables del id12

```
C1 VAR BYTE
C2 VAR BYTE
```

C3	VAR	BYTE
C4	VAR	BYTE
C5	VAR	BYTE
C6	VAR	BYTE
C7	VAR	BYTE
C8	VAR	BYTE
C9	VAR	BYTE
C10	VAR	BYTE
C11	VAR	BYTE
C12	VAR	BYTE

: Pines de comunicación del id12

SINT	var	PORTD.2
------	-----	---------

: Pines del RTC (DS1307)

CPIN	VAR	PORTC.3; pin de señal de reloj I2C PIN 18
DPIN	VAR	PORTC.4; pin de señal de datos I2C PIN 23
SEGU	VAR	BYTE
MINU	VAR	BYTE
HORA	VAR	BYTE
DIASEM	VAR	BYTE
DIAMES	VAR	BYTE
MES	VAR	BYTE
ANIO	VAR	BYTE
D1	VAR	BYTE
D2	VAR	BYTE
ACTUALIZADO	VAR	BIT
POS	VAR	BYTE
RLJ	VAR	BYTE

: Pines de chapa magnética

cm	VAR	PORTC.1; pin de conexión de relé de chapa magnética
PULS	VAR	PORTB.4; pin desconexión de cm, desde adentro aula
SREAD	VAR	BYTE

: Pin de Ethernet

ETH	VAR	PORTC.2
PIN_DES	VAR	PORTB.0

: Pines de comunicación de datalogger

LOGINT	VAR	PORTD.3
LOGOUT	VAR	PORTC.0
P1	VAR	BYTE
P2	VAR	BYTE
P3	VAR	BYTE
P4	VAR	BYTE
P5	VAR	BYTE
P6	VAR	BYTE
PH	VAR	BYTE
PM	VAR	BYTE

PUNT_DIA	VAR	WORD
PUNT_HORA	VAR	WORD
PUNT_HORA1	VAR	WORD
PUNT_MINU	VAR	WORD
PUNT_TARJ	VAR	WORD
PUNT_NOM	VAR	WORD
PUNT_LTR	VAR	WORD
PUNT_MAT	VAR	WORD
PUNT_MANT	VAR	WORD
PUNT_N_MANT	VAR	WORD
T1	VAR	BYTE
T2	VAR	BYTE
T3	VAR	BYTE
T4	VAR	BYTE
T5	VAR	BYTE
T6	VAR	BYTE
T7	VAR	BYTE
T8	VAR	BYTE
T9	VAR	BYTE
T10	VAR	BYTE
T11	VAR	BYTE
T12	VAR	BYTE
T13	VAR	BYTE
T14	VAR	BYTE
T15	VAR	BYTE
T16	VAR	BYTE
T17	VAR	BYTE
T18	VAR	BYTE
T19	VAR	BYTE
T20	VAR	BYTE
S1	VAR	BYTE
S2	VAR	BYTE
S3	VAR	BYTE
S4	VAR	BYTE
S5	VAR	BYTE
S6	VAR	BYTE
S7	VAR	BYTE
S8	VAR	BYTE
S9	VAR	BYTE
S10	VAR	BYTE
S11	VAR	BYTE
S12	VAR	BYTE
S13	VAR	BYTE
S14	VAR	BYTE
S15	VAR	BYTE
S16	VAR	BYTE
S17	VAR	BYTE
S18	VAR	BYTE
S19	VAR	BYTE
S20	VAR	BYTE

```

S21          VAR    BYTE
M1           VAR    BYTE
M2           VAR    BYTE
M3           VAR    BYTE
M4           VAR    BYTE

```

; Declaración de puertos

```

PAUSE 6000    ; antes 6000
TRISB = %00000000

```

```

P3 = 0

```

```

P4 = 0

```

```

HIGH CM

```

```

LCDOUT $FE, 1

```

; Cuerpo del programa (Cabecera)

```

Main:

```

; Variables encerradas

```

C1 = 0:C2 =0:C3 =0:C4 =0:C5 =0:C6 =0:

```

```

C7 =0:C8 =0:C9 =0:C10 =0:C11 =0:C12 =0:

```

```

SEGU=0:MINU =0:HORA=0:DIASEM =0:DIAMES =0:MES =0:ANIO

```

```

=0:D1 =0:D2 =0:ACTUALIZADO=0:POS =0:RLJ =0

```

```

P1 =0:P2=0:P3=0:P4=0:P5=0:P6=0:PH=0:PM=0

```

```

T1=0:T2=0:T3=0:T4=0:T5=0:T6=0:T7=0:T8=0:T9=0:T10=0:T11=0:T12=0:

```

```

T13=0:T14=0:T15=0:T16=0:T17=0:T18=0:T19=0:T20=0:

```

```

M1=0:M2=0:M3=0:M4=0

```

```

PULS =0:PORTC.1 =0:D1=0:D2=0:P1=0:P2=0

```

```

SREAD = 0

```

```

PUNT_DIA = 0

```

```

PUNT_HORA = 0

```

```

PUNT_NOM = 0

```

```

PUNT_MINU = 0

```

```

PUNT_TARJ = 0

```

```

PUNT_MAT = 0

```

```

PUNT_LTR = 0

```

```

PUNT_MANT = 0

```

```

PUNT_N_MANT = 0

```

```

POS = 0

```

```

PIN_DES=0

```

; Mensaje principal en el LCD 20X4

```

LCDOUT $FE,$80,"Deslice la tarjeta "

```

```

lcdout $FE,$0C0,"          "

```

```

lcdout $FE,$94,"          "

```

```

lcdout $FE,$D4,"          "

```

; Sub programas dentro del main (programa principal; aquí cada subprograma ejecuta funciones específicas, y no regresa al programa principal hasta terminar con su proceso individual.

```
GOSUB PULSANTE
GOSUB DESCONEC
GOSUB LECT_TARJ
GOSUB DS1307
GOSUB MANTEN
GOSUB BUSC_DIA
GOSUB HORA_INI
GOSUB MINU_INI
GOSUB BUSC_TARJ
```

```
;GOSUB GRAB_LOG
  GOSUB ENV_COMP
GOSUB BUSC_LTR
  PUNT_MAT = PUNT_LTR + 3
  GOSUB LECT_MAT
  PUNT_NOM = PUNT_LTR + 23
  GOSUB LECT_PROF
  GOSUB GRAB_LOG
GOSUB HAB_PUERTA
```

GOTO MAIN **; Sintaxis que indica regreso a la cabecera principal main**

; Ciclo MANTEN, en el cual las líneas de programación busca en el pendrive si la tarjeta leída pertenece a un código de mantenimiento.

```
MANTEN: LCDOUT $FE,$D4," BUSCANDO..."
SEROUT2 LOGOUT,84,[$0D]
  PAUSE 20
  SEROUT2 LOGOUT,84,["IPA", $0D]
  PAUSE 20
  SEROUT2 LOGOUT,84,["OPR MANTEN.TXT", $0D]
  PAUSE 20
  SEROUT2 LOGOUT,84,["SEK ",DEC PUNT_MANT, $0D]
  SEROUT2 LOGOUT,84,["RDF 12", $0D]
  SERIN2 LOGINT,84,[T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12]
  PAUSE 20
  SEROUT2 LOGOUT,84,["CLF MANTEN.TXT", $0D]

  IF
(C1==T1)AND(C2==T2)AND(C3==T3)AND(C4==T4)AND(C5==T5)AND(C6==
T6)AND(C7==T7)AND(C8==T8)AND(C9==T9)AND(C10==T10)AND(C11==T1
1)AND(C12==T12)THEN
  GOSUB GRAB_LOG
  PUNT_N_MANT = PUNT_MANT + 13
  LCDOUT $FE,1
  LCDOUT $FE,$80," BIENVENIDO"
```



```
LCDOUT $FE,$D4
GOSUB NOM_MANTEN
GOSUB HAB_PUERTA
GOTO MAIN
ELSE
```

```
if (T1 < 174) then
    PUNT_MANT = PUNT_MANT + 28:PAUSE 50:GOTO MANTEN
else
    RETURN
endif
endif
```

; Ciclo NOM MANTEN, en el cual las líneas de programación busca en el pendrive el nombre a quién pertenece la tarjeta de mantenimiento.

```
NOM_MANTEN: SEROUT2 LOGOUT,84,[$0D]
    PAUSE 20
    SEROUT2 LOGOUT,84,["IPA", $0D]
    PAUSE 20
    SEROUT2 LOGOUT,84,["OPR MANTEN.TXT", $0D]
    PAUSE 20
    SEROUT2 LOGOUT,84,["SEK ", DEC PUNT_N_MANT, $0D]
    SEROUT2 LOGOUT,84,["RDF 13", $0D]
    SERIN2 LOGINT,84,[T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13]
    PAUSE 20
    SEROUT2 LOGOUT,84,["CLF MANTEN.TXT", $0D]

    LCDOUT T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13
RETURN
```

; Ciclo DESCONEC, en el cual las líneas de programación Actualiza Datos en el módulo

```
DESCONEC: POS = 0
    IF PIN_DES == 1 THEN
LCDOUT $FE,$1," ACTUALIZANDO DATOS";:PAUSE 1000
PIN_DES = 0

        FOR POS = 0 TO 6
serin2 ETH,84,600,DESCONEC,[RLJ]
        GOSUB GRABAR_RTC
NEXT
    ELSE
    RETURN
ENDIF
```

; Ciclo PULSANTE, en el cual las líneas de programación Habilita chapa mangnetica.

```

PULSANTE: IF PULS == 1 THEN
    GOSUB HAB_PUERTA
    ELSE
RETURN
    ENDIF

```

: Ciclo LECT TARJ, líneas de programación que recogen el código de las tarjetas RFID.

```

LECT_TARJ:
SERIN2
SINT,84,100,MAIN,[WAIT($02),C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12]
RETURN

```

```

GRAB_LOG: SEROUT2 LOGOUT,84,["OPW ID12.XLS", $0D]
    PAUSE 20
    SEROUT2 LOGOUT,84,["WRF 59", $0D] ;ANTES 34
PAUSE 20
; SEROUT2
LOGOUT,84,[C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12, ".", $09]
;GRABACION DEL CODIGO DE LA TARJETA
SEROUT2
LOGOUT,84,[S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15,S16,S17,S18,S19
,S20,S21, $09] ;GRABACION DEL NOMBRE DEL PROFESOR
SEROUT2
LOGOUT,84,[T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15,T16,T17,T1
8,T19,T20, $09] ;GRABACION DEL MATERIA
SEROUT2 LOGOUT,84,[HEX2 DIAMES, "/", HEX2 MES, "/20", HEX2 ANIO, $09]
;GRABACION DE LA FECHA
SEROUT2 LOGOUT,84,[HEX2 HORA, ":", HEX2 MINU, $0D]
;GRABACION DE LA HORA

```

```

PAUSE 20
SEROUT2 LOGOUT,84,["CLF ID12.XLS", $0D]
PAUSE 20
RETURN

```

: Ciclo ENV COMP, envío de código RFID, fuera del módulo.

```

ENV_COMP: SEROUT2 ETH,84,[c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12]
    PAUSE 20
RETURN

```

```

BUSC_DIA: PUNT_DIA = (D2)*(290)
RETURN

```

: Ciclo HORA INI, busca la hora.

```

HORA_INI: IF HORA <= $9 THEN
PUNT_HORA = 4+(HORA-$7)*$13
ELSE

```

```

IF (HORA >= $A AND HORA <= $19) THEN
PUNT_HORA = $4+(HORA-$D)*$13
ELSE
  IF (HORA >= $20) THEN
    PUNT_HORA = $4+(HORA-$13)*$13
  ENDIF
ENDIF
ENDIF
ENDIF

```

```

PUNT_HORA1 = PUNT_HORA + PUNT_DIA
PUNT_MINU = PUNT_HORA1 + 2
PUNT_TARJ = PUNT_MINU + 2

```

```

RETURN

```

: Ciclo MINU INI, busca los minutos.

```

MINU_INI: SEROUT2 LOGOUT,84,["OPR CLASE.TXT", $0D]
PAUSE 200
SEROUT2 LOGOUT,84,["SEK ", DEC PUNT_MINU, $0D]
SEROUT2 LOGOUT,84,["RDF 2", $0D]
SERIN2 LOGINT,84,[P5,P6]
PAUSE 200
SEROUT2 LOGOUT,84,["CLF CLASE.TXT", $0D]
PAUSE 200

  IF (P5 == $30) THEN
    PM = (P6-48)
  ELSE
if (P5 == $31) THEN
  PM = (P5-33)+(P6-48)
  ELSE
  IF (P5 == $32) THEN
    PM = (P5-18)+(P6-48)
  ELSE
  IF (P5 == $33) THEN
    PM = (P5-3)+(P6-48)
  ELSE
  IF (P5 == $34) THEN
    PM = (P5+12)+(P6-48)
  ELSE
  IF (P5 == $35) THEN
    PM = (P5+27)+(P6-48)
  ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

```

```

        IF (MINU >= PM) THEN
        RETURN
        ELSE
LCDOUT $FE,$D4," FUERA DE HORARIO": PAUSE 1000
ENDIF
;GOTO MAIN

BUSC_TARJ: SEROUT2 LOGOUT,84,["OPR CLASE.TXT",$0D]
        PAUSE 50
        SEROUT2 LOGOUT,84,["SEK ",DEC PUNT_TARJ, $0D]
        SEROUT2 LOGOUT,84,["RDF 14",$0D]
        SERIN2 LOGINT,84,[T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,M1,M2]
        PAUSE 50
        SEROUT2 LOGOUT,84,["CLF CLASE.TXT",$0D]
        PAUSE 50

        IF
(C1==T1)AND(C2==T2)AND(C3==T3)AND(C4==T4)AND(C5==T5)AND(C6==
T6)AND(C7==T7)AND(C8==T8)AND(C9==T9)AND(C10==T10)AND(C11==T1
1)AND(C12==T12)THEN
        LCDOUT $FE,$1:
        RETURN
        ELSE
        LCDOUT $FE,$D4," MARC. RECHAZADA":PAUSE 2000:GOTO MAIN
        ENDIF

BUSC_LTR: SEROUT2 LOGOUT,84,["OPR PROFESOR.TXT",$0D]
        PAUSE 20
        SEROUT2 LOGOUT,84,["SEK ",DEC PUNT_LTR, $0D]
        PAUSE 20
        SEROUT2 LOGOUT,84,["RDF 2",$0D]
        SERIN2 LOGINT,84,[M3,M4]
        PAUSE 20
        SEROUT2 LOGOUT,84,["CLF PROFESOR.TXT",$0D]
        PAUSE 20

        IF (M1 == M3) THEN
        IF (M2 == M4) THEN
        RETURN
        ELSE
        PUNT_LTR = PUNT_LTR + 46
GOTO BUSC_LTR
        ENDIF
        ENDIF

```

; Ciclo LECT MANT, busca la materia dentro del pendrive..

```
LECT_MAT: SEROUT2 LOGOUT,84,[$0D]
```

```

SEROUT2 LOGOUT,84,["OPR PROFESOR.TXT",\$0D]
PAUSE 20
SEROUT2 LOGOUT,84,["SEK ",DEC PUNT_MAT, \$0D]
PAUSE 20
SEROUT2 LOGOUT,84,["RDF 20",\$0D]
SERIN2
LOGINT,84,[T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15,T16,T17,T1
8,T19,T20]
;SERIN2 LOGINT,84,[M1]
PAUSE 20
SEROUT2 LOGOUT,84,["CLF PROFESOR.TXT",\$0D]
PAUSE 20
LCDOUT \$FE,\$80," MARC. ACEPTADA"
LCDOUT
\$FE,\$94,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15,T16,T17,T18,T1
9,T20
LCDOUT \$FE,\$d4
RETURN

```

; Ciclo LECT PROF, busca el profesor dentro del pendrive. .

```

LECT_PROF: SEROUT2 LOGOUT,84,[\$0D]
SEROUT2 LOGOUT,84,["OPR PROFESOR.TXT",\$0D]
PAUSE 20
SEROUT2 LOGOUT,84,["SEK ",DEC PUNT_NOM, \$0D]
PAUSE 20
SEROUT2 LOGOUT,84,["RDF 21",\$0D]
SERIN2
LOGINT,84,[S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15,S16,S17,S18,S
19,S20,S21]
PAUSE 20
SEROUT2 LOGOUT,84,["CLF PROFESOR.TXT",\$0D]
PAUSE 20
LCDOUT
S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15,S16,S17,S18,S19,S20,S21
RETURN

```

; Ciclo HAB PUERTA, activa la etapa de potencia cuando el control de acceso ya ha verificado el personal y éste es correcto.

```

HAB_PUERTA: high CM      ;anteslow cm
PAUSE 5000
low CM      ;antes high cm
GOTO MAIN

```

; Ciclo DS1307, se encarga los valores de fecha y hora almacenados en el RTC DS1307

```

DS1307:
I2CREAD DPIN, CPIN, %11010000,0,[SEGU]

```

```

I2CREAD DPIN, CPIN, %11010000,1,[MINU]
I2CREAD DPIN, CPIN, %11010000,2,[HORA]
I2CREAD DPIN, CPIN, %11010000,3,[DIASEM]
I2CREAD DPIN, CPIN, %11010000,4,[DIAMES]
I2CREAD DPIN, CPIN, %11010000,5,[MES]
I2CREAD DPIN, CPIN, %11010000,6,[ANIO]
LCDOUT $FE,1,HEX2 HORA,":",HEX2 MINU,":",HEX2 SEGU
LCDOUT $FE,$C0

```

```

IF DIASEM = $1 THEN
LCDOUT "DOM.":D1 = "D":D2 = 0:ENDIF
IF DIASEM = $2 THEN
LCDOUT "LUN.":D1 = "L":D2 = 1:ENDIF

```

```

IF DIASEM = $3 THEN
LCDOUT "MAR.":D1 = "M":D2 = 2:ENDIF
IF DIASEM = $4 THEN
LCDOUT "MIE.":D1 = "W":D2 = 3:ENDIF
IF DIASEM = $5 THEN
LCDOUT "JUE.":D1 = "J":D2 = 4:ENDIF
IF DIASEM = $6 THEN
LCDOUT "VIE.":D1 = "V":D2 = 5:ENDIF
IF DIASEM = $7 THEN
LCDOUT "SAB.":D1 = "S":D2 = 6:ENDIF

```

```

LCDOUT $FE,$C5,HEX2 DIAMES,"/"
LCDOUT $FE,$CB,"/20",HEX2 ANIO

```

```

LCDOUT $FE,$C8
IF MES = $1 THEN LCDOUT "ENE"
IF MES = $2 THEN LCDOUT "FEB"
IF MES = $3 THEN LCDOUT "MAR"
IF MES = $4 THEN LCDOUT "ABR"
IF MES = $5 THEN LCDOUT "MAY"
IF MES = $6 THEN LCDOUT "JUN"
IF MES = $7 THEN LCDOUT "JUL"
IF MES = $8 THEN LCDOUT "AGO"
IF MES = $9 THEN LCDOUT "SEP"
IF MES = $10 THEN LCDOUT "OCT"
IF MES = $11 THEN LCDOUT "NOV"
IF MES = $12 THEN LCDOUT "DIC"

```

```

;PAUSE 2000
RETURN

```

```

GRABAR_RTC:
I2CWRITE DPIN,CPIN,%11010000,POS,[RLJ] ;SEGUNDOS
RETURN

```

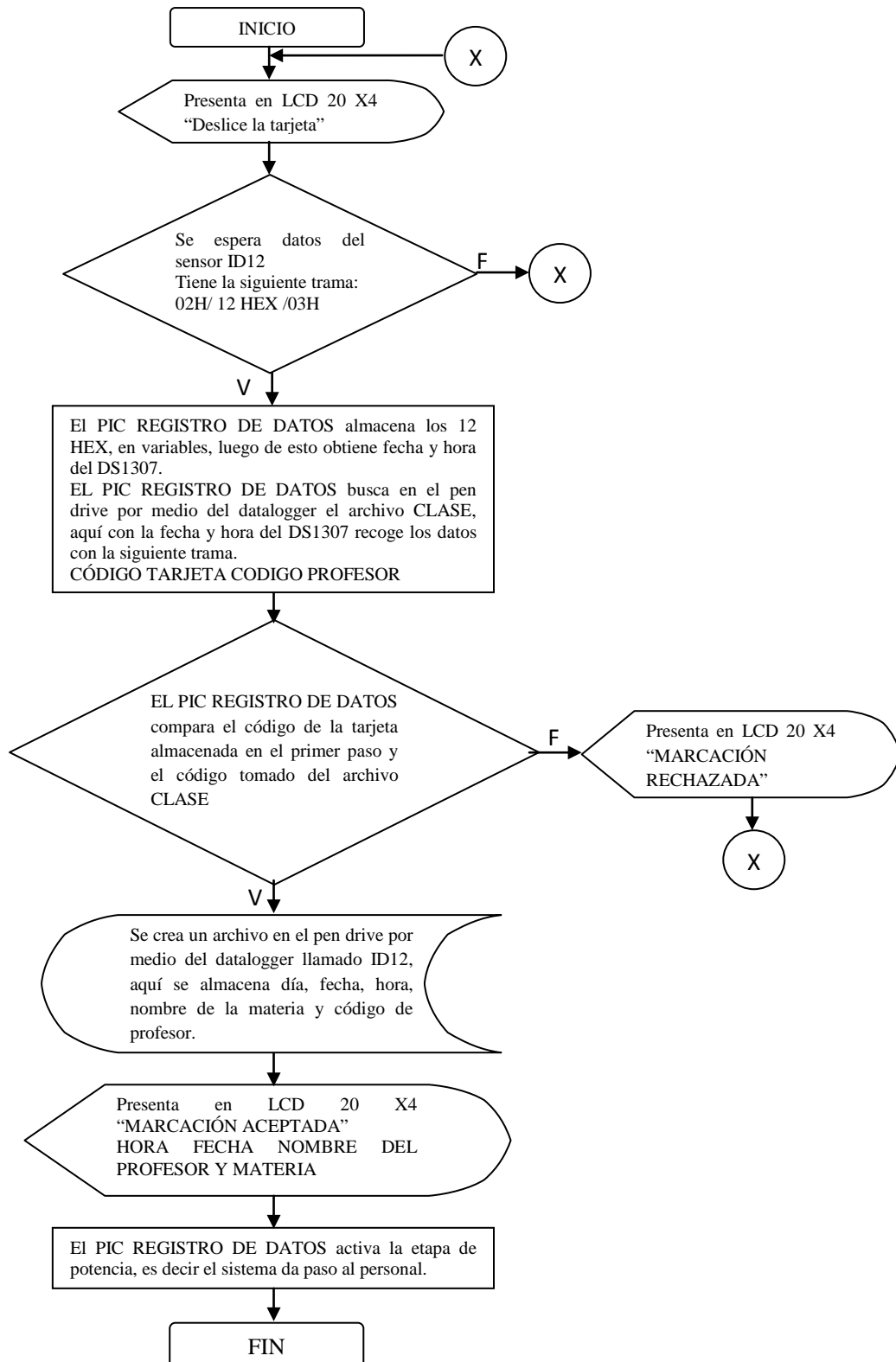
```

END

```

4.2 Diagrama de flujo “Módulo Registro de Datos”

Otra forma de entender líneas de programación es por medio de un diagrama de flujo, a continuación se muestra este tipo de diagramas representando la secuencia que ejecuta el Módulo Registro de Datos.



4.3 Programación del “PIC16F876 del MÓDULO REGISTRO DE DATOS”

Otro microcontrolador presente en el módulo Registro de Datos es el PIC16F876, en las líneas siguientes se muestra la programación de dicho integrado.

```
*****
'* Name : MOD_ETHERNET.BAS *
'* Author : DAVID NUNEZ Y CRISTHIAN GARCIA *
'* Notice : CONTROL DE ACCESO Y MARCACIÓN DE ENTRADA PARA *
'* : PERSONAL CON TARJETA RFID Y COMUNICACIÓN *
'* : INALÁMBRICA *
'* Notes : ESTE PROGRAMA SE ENCUENTRA UBICADO EN EL MODULO *
'* EN MENCION, EN EL PIC 16F876 EL CUAL SIRVE PARA *
'* UBICAR DIRECCIONES DEL MODULO *
*****
```

: Incluye los módulos de comunicación

```
INCLUDE "MODEDEFS.BAS"; INCLUYE LOS MODOS DE COMUNICACION
define osc 4
```

: Bus comunicación I2C

```
define I2C_SCLOUT 1
```

```
trisB=%00000001
OPTION_REG = %10000000
```

: Definición del microcontrolador

```
DEFINE HSER_RCSTA 90h
DEFINE HSER_TXSTA 24h
DEFINE HSER_BAUD 9600
DEFINE HSER_SPBRG 25
```

:Definición de LCD

```
define LCD_DREG PORTB
define LCD_DBIT 4
define LCD_RSREG PORTB
define LCD_RSBIT 1
define LCD_EREG PORTC
define LCD_EBIT 5
```

: Pines de comunicación de datalogger

```
D0 VAR BYTE
D1 VAR BYTE
D7 VAR BYTE
DIR VAR BYTE
```

: Pines de comunicación de RS485

```
LOGINT VAR PORTC.0
```



```
LOGOUT  VAR    PORTC.1
CONTROL VAR    PORTC.2
```

: Variables de DS1307

```
RLJ     VAR    BYTE
RLJ0    VAR    BYTE
RLJ1    VAR    BYTE
POS     VAR    BYTE
CPIN    VAR    PORTC.3
DPIN    VAR    PORTC.4
```

: Variables generales

```
ACK     VAR    BYTE
ETX     VAR    PORTB.2
AVI_COM VAR    PORTB.3
C1      VAR    BYTE
C2      VAR    BYTE
C3      VAR    BYTE
C4      VAR    BYTE
C5      VAR    BYTE
C6      VAR    BYTE
C7      VAR    BYTE
C8      VAR    BYTE
C9      VAR    BYTE
C10     VAR    BYTE
C11     VAR    BYTE
C12     VAR    BYTE
```

```
DIREC_PUER VAR    BYTE
```

```
ADCON1 = %00000110
```

```
TRISA = %11111111
```

```
OnInterruptGoto ING
```

```
INTCON = %10010000
```

: Variables encero

```
PAUSE 3000
```

```
MAIN:  LOW AVI_COM
```

```
        LOW CONTROL
```

```
LCDOUT $FE,$1
```

```
        DIREC_PUER = PORTA
```

```
        LCDOUT $FE,$80,DEC PORTA
```

```
        RLJ0 = 0:RLJ1 = 0:RLJ = 0:POS = 0
```

```
        D0 = 0:D1 = 0
```

```
        GOSUB PUERTA
```

```
        GOSUB ENV_COMP
```

GOTO MAIN

PUERTA: SERIN2
ETX,84,100,MAIN,[C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12]
LCDOUT \$FE,\$80,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12
PAUSE 50
RETURN

ENV_COMP: HIGH control

IF DIREC_PUER <= \$9 THEN
SEROUT2 LOGOUT,84,["0",DEC
DIREC_PUER,":",C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12] ;ESPERA A
QUE LLEGUE DATO
ELSE
SEROUT2 LOGOUT,84,[DEC
DIREC_PUER,":",C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12] ;ESPERA A
QUE LLEGUE DATO
ENDIF

LOW CONTROL
RETURN

DISABLE

ING: PAUSE 15

low control

SERIN2 LOGINT,84,600,SAL_INT,[D0] ; *Espere a que llegue el dato*
SERIN2 LOGINT,84,600,SAL_INT,[D1]
SERIN2 LOGINT,84,600,SAL_INT,[D7]

HIGH CONTROL

IF D0 == \$30 THEN
DIR = (D1 - 48)
ELSE
IF D0 == \$31 THEN
DIR = (D0 - 33) + (D1 - 54)
ENDIF
ENDIF

IF DIREC_PUER == DIR THEN
GOTO PRO
ELSE

LOW CONTROL:DIR = 0
INTCON = %10010000

resume MAIN

```
GOTO MAIN
ENDIF
```

```
PRO:    HIGH AVI_COM
IF D7 == 51 THEN      ;LLEGA DATO 3,ACTUALIZA FECHA Y HORA
GOTO RELOJ
ELSE
    IF D7 == 52 THEN      ; BUSCAR MODULOS
GOTO PING
ELSE
ENDIF
ENDIF
```

```
RELOJ:  HIGH CONTROL
SEROUT2 LOGOUT,84,["D"]
```

```
low control
FOR POS = 0 TO 6
;low control
SERIN2 LOGINT,84,[RLJ0,RLJ1]; Espera que llegue el dato
;HIGH CONTROL
```

```
RLJ0 = RLJ0 << 4
RLJ1 = RLJ1 - 48
RLJ = RLJ0 + RLJ1
SEROUT2 ETX,84,[RLJ]
PAUSE 10
NEXT
HIGH CONTROL
```

```
;HIGH CONTROL
SEROUT2 LOGOUT,84,["A"]
LOW CONTROL
```

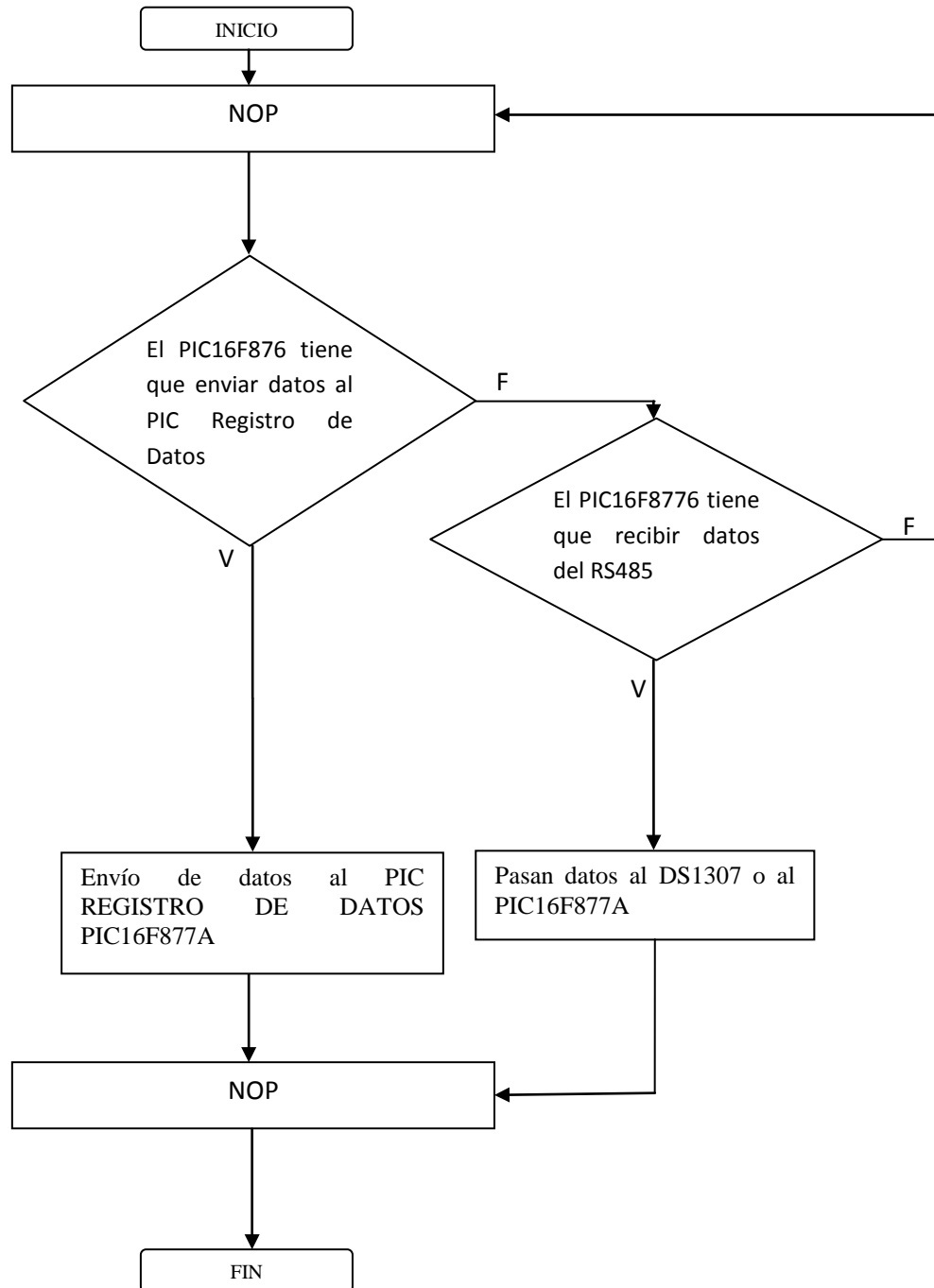
```
INTCON = %10010000
resume MAIN
```

```
PING:  HIGH CONTROL
SEROUT2 LOGOUT,84,["X"]
LOW CONTROL
```

```
SAL_INT: INTCON = %10010000
resume MAIN
ENABLE
END
```

4.4 Diagrama de flujo “PIC16F876 del Módulo Registro de Datos”

Se explica mediante un diagrama de flujo la secuencia en ciclo cerrado que realiza el PIC16F876, para trabajar conjuntamente con la red RS485.



4.5 Programación del “Módulo Base”

Este programa detalla la forma de comunicación entre los módulos de registro de datos y el computador principal.

```
*****
'* Name : MOD_BASE.BAS *
'* Author : DAVID NUNEZ Y CRISTHIAN GARCIA *
'* Notice : CONTROL DE ACCESO Y MARCACIÓN DE ENTRADA PARA *
'* : PERSONAL CON TARJETA RFID Y COMUNICACIÓN *
'* : INALÁMBRICA *
'* Notes : ESTE PROGRAMA SE ENCUENTRA UBICADO EN EL MODULO *
'* EN MENCIÓN, EL CUAL SIRVE DE INTERFAZ ENTRE LOS *
'* MODULOS REGISTRO DE DATOS Y LA COMPUTADORA *
'* *
*****
```

; Definición del oscilador

```
@DEVICEXT_OSC
DEFINE OSC 4
DEFINE SER2_BITS 8 ' Set Serin2 and Serout2 data bits to 8
```

; Incluye los módulos de comunicación

```
INCLUDE "MODEDEFS.BAS"
trisB=%00000001
;OPTION_REG = %10000000
```

;Pines de Comunicación del PC

```
DEFINE HSER_RCSTA 90h ;RC7 (PIN 26) ES RX
DEFINE HSER_TXSTA 24h ;RC6 (PIN 25) ES TX
DEFINE HSER_BAUD 9600
DEFINE HSER_SPBRG 25
```

;Pines de comunicación de RS485

```
CONTROL VAR PORTC.0 ;CONTROL A 485
LOGOUT VAR PORTC.2 ;TX A 485
LOGINT VAR PORTC.1 ;RX A 485
```

;Pin de comunicación a Comp.

```
COMP VAR PORTB.0 ;RX A COMPUTADORA
LED VAR PORTB.1 ;
```

;Variables de número de tarjeta.

```
DIREC_PUER VAR BYTE
C1 VAR BYTE
C2 VAR BYTE
C3 VAR BYTE
C4 VAR BYTE
```

```

C5    VAR    BYTE
C6    VAR    BYTE
C7    VAR    BYTE
C8    VAR    BYTE
C9    VAR    BYTE
C10   VAR    BYTE
C11   VAR    BYTE
C12   VAR    BYTE

```

; Variables Generales.

```

D0    VAR    BYTE
D1    VAR    BYTE
D7    VAR    BYTE
D22   VAR    BYTE
DIREC_MOD VAR    BYTE
DAT_RELOJ VAR    BYTE
RLJ1  VAR    BYTE
RLJ0  VAR    BYTE
PIN   VAR    BYTE
PAUSE 3000

```

; Cabecera del programa.

```

MAIN:
    LOW LED
    LOGINT=0:LOGOUT=0:COMP=0:LED=0
D0=0:D1=0:D7=0:DIREC_PUER=0:DIREC_MOD=0
    C1=0:C2=0:C3=0:C4=0:C5=0:C6=0:
    C7=0:C8=0:C9=0:C10=0:C11=0:C12=0
DAT_RELOJ=0:RLJ0=0:RLJ1=0:PIN=0

```

```

clear
    GOSUB M_PUERTAS
    GOSUB REC_COMP
    GOTO MAIN

```

; Ciclo M PUERTAS, recoge los datos de los módulos puerta.

```

M_PUERTAS: LOW CONTROL
    SERIN2
LOGINT,84,100,REC_COMP,[DIREC_PUER,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,
C11,C12]
PAUSE 10
    DIREC_MOD = DIREC_PUER + 48
    HIGH CONTROL
    HSEROUT [DIREC_PUER,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12]
    RETURN

```

```

REC_COMP:
    HIGH LED
    IF COMP == 1 THEN

```

```

GOTO A
ELSE
GOTO MAIN
ENDIF
A:   SERIN2 COMP,84,600,MAIN,[D22]
      SERIN2 COMP,84,600,MAIN,[D0]
      SERIN2 COMP,84,600,MAIN,[D1]
      SERIN2 COMP,84,600,MAIN,[D7]
      HIGH CONTROL
      serout2 logout,84,[D22]:PAUSE 250:serout2 logout,84,[D0]:PAUSE
250:serout2 logout,84,[D1]:PAUSE 250:serout2 logout,84,[D7]
      LOW CONTROL
;pause 13

if D7 == 51 then      ;ENVIAR ACTUALIZACION RELOJ
goTO ENV_RELOJ
      ELSE
if D7 == 52 then      ;ENVIAR PING
goTO PING
      ENDIF
      ENDIF
      GOTO MAIN

ENV_RELOJ: LOW CONTROL
      PAUSE 100

      DAT_RELOJ = 0
      HIGH control
      FOR DAT_RELOJ = 1 TO 7
      SERIN2 COMP,84,[RLJ0,RLJ1]
      PAUSE 10
;HIGH control
      SEROUT2 LOGOUT,84,[RLJ0,RLJ1]
;LOW control
      NEXT
      LOW control
goto MAIN

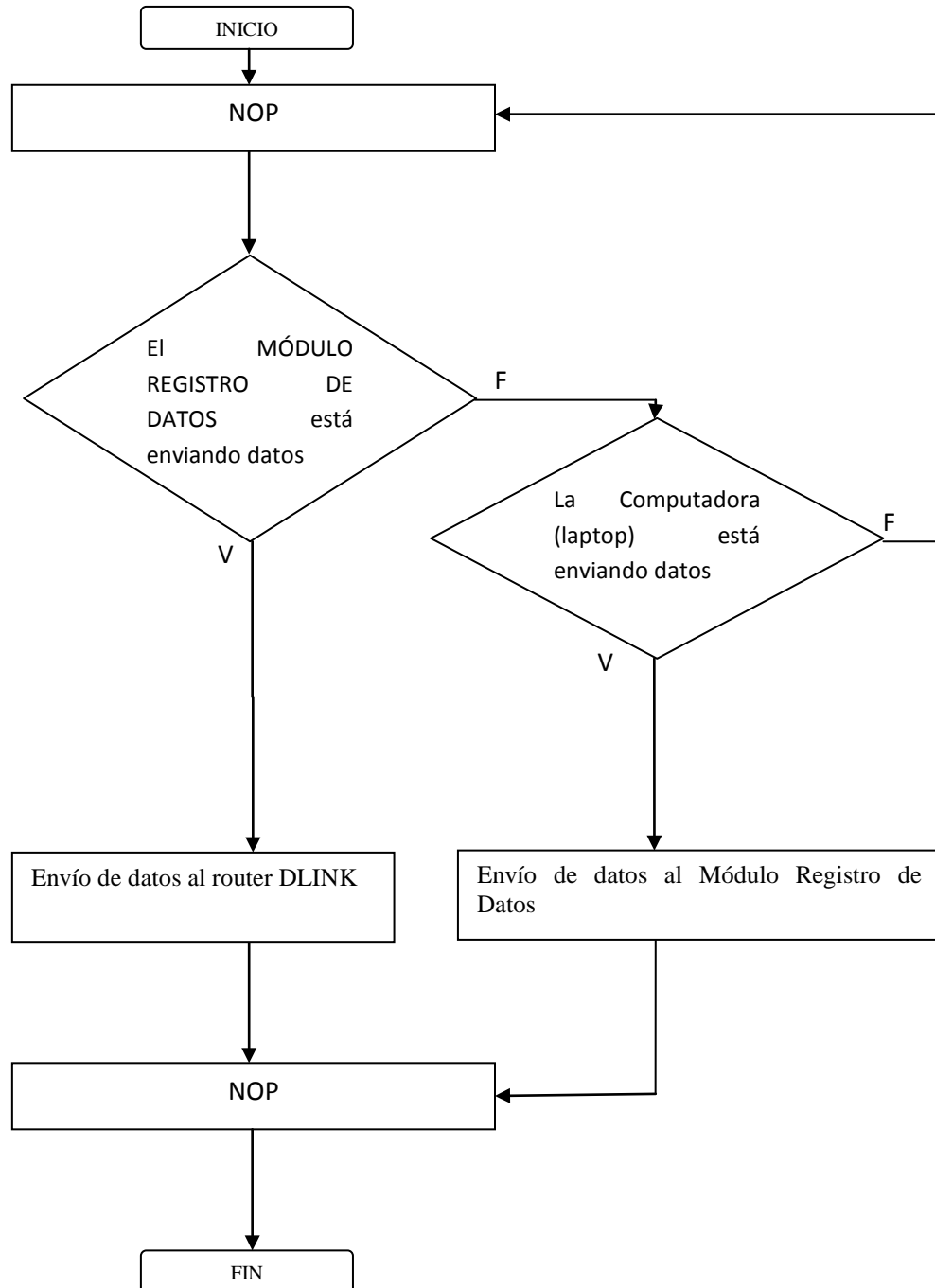
PING:   LOW control
GOTO MAIN

END

```

4.6 Diagrama de flujo “Módulo Base”

El diagrama de flujo presenta una idea más clara del funcionamiento del módulo base, siendo éste una parte importante del proyecto, ya que relaciona los datos del sistema en general.



4.7 Programación del “Módulo ID12”

Este módulo sirve para ingresar nuevos usuarios al sistema, básicamente lee el código de una tarjeta y con ayuda del software Cardget se ingresa información personal (nombres, número de cédula, teléfonos, etc). Con esto los datos se almacenan en el computador. A continuación presentamos su programación.

```
'*****  
'* Name : MOD_ID12.BAS *  
'* Author : DAVID NUNEZ Y CRISTHIAN GARCIA *  
'* Notice : CONTROL DE ACCESO Y MARCACIÓN DE ENTRADA PARA *  
'* : PERSONAL CON TARJETA RFID Y COMUNICACIÓN *  
'* Notes : ESTE PROGRAMA SE ENCUENTRA UBICADO EN EL MODULO id12*  
'* CAPAZ DE AGREGAR TARJETAS RFID A LA BASE DEL SISTEMA *  
'*****
```

; Definición del microcontrolador

```
@DEVICEXT_OSC
```

```
INCLUDE "MODEDEFS.BAS"; INCLUYE LOS MODOS DE COMUNICACION  
define osc 4
```

; Variables del id12

```
C1 VAR BYTE  
C2 VAR BYTE  
C3 VAR BYTE  
C4 VAR BYTE  
C5 VAR BYTE  
C6 VAR BYTE  
C7 VAR BYTE  
C8 VAR BYTE  
C9 VAR BYTE  
C10 VAR BYTE  
C11 VAR BYTE  
C12 VAR BYTE
```

; Pines de comunicación del id12

```
SINT var PORTD.2  
ETH VAR PORTC.2
```

; Cabecera del programa

```
MAIN: gosub LECT_TARJ  
GOSUB ENV_COMP  
GOTO MAIN
```

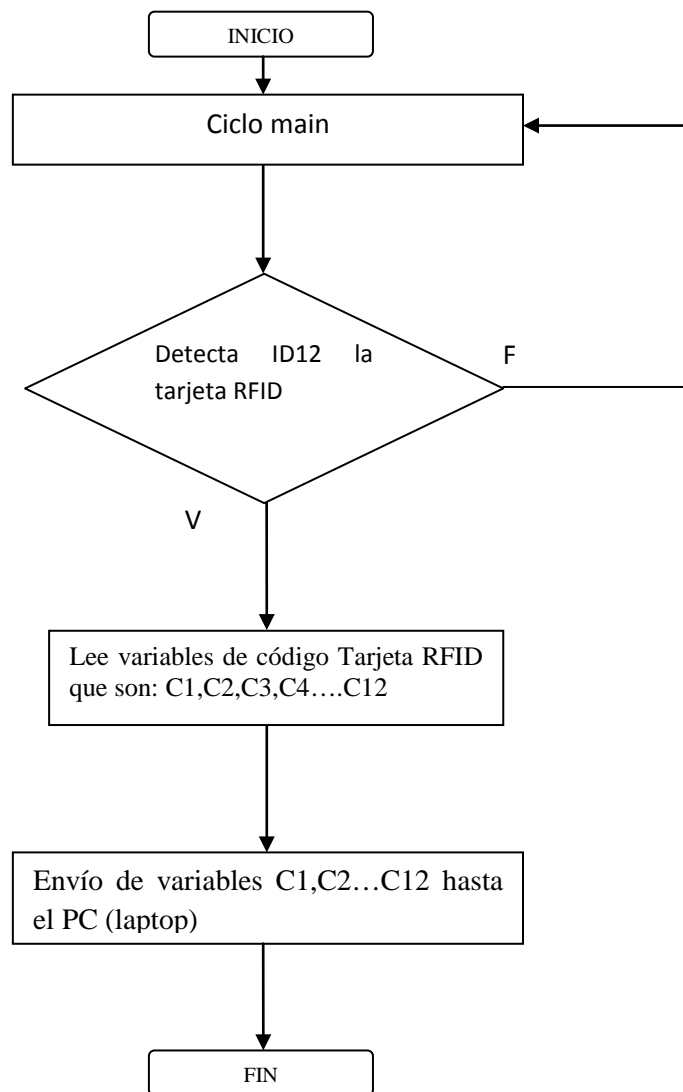
```
LECT_TARJ:
```

```
SERIN2
SINT,84,100,MAIN,[WAIT($02),C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12]
RETURN
```

```
ENV_COMP: SEROUT2 ETH,84,[c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12]
PAUSE 20
RETURN
END
```

4.8 Diagrama de flujo “Módulo ID12”

Las mismas características con la sección anterior, pero usando diagrama de flujo para una mejor comprensión del proceso.



4.9 Software (Cardget)

El icono para iniciar el programa es el indicado en la Figura 95.

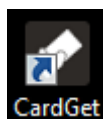


Figura 95 Ícono del software CardGet.
Fuente: Autores

Al ejecutar el software, se presenta la siguiente pantalla (Figura 96), en la que nos pide el usuario y contraseña. El usuario por default es **admin** y la contraseña es **malcom**, pero también se puede crear más usuarios con sus propias claves, dependiendo de cuántas personas tienen acceso al programa. Cabe indicar que existen tres intentos para ingresar al software, en el caso de fallar esta pantalla se cierra.



Figura 96 Ventana de contraseña
Fuente: Autores

Después de ingresar los datos correctos se presenta la ventana para seleccionar el Gateway (Figura 97). En la cual nos indica la dirección IP en la cual se encuentra la red de controles de acceso.

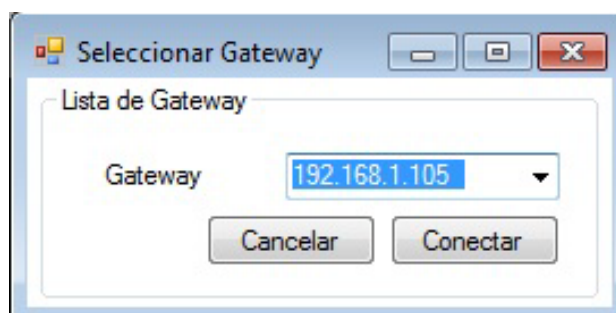


Figura 97 Selección Gateway
Fuente: Autores

En caso de que no se pueda conectar a la IP seleccionada, aparece un mensaje de error como la siguiente pantalla.

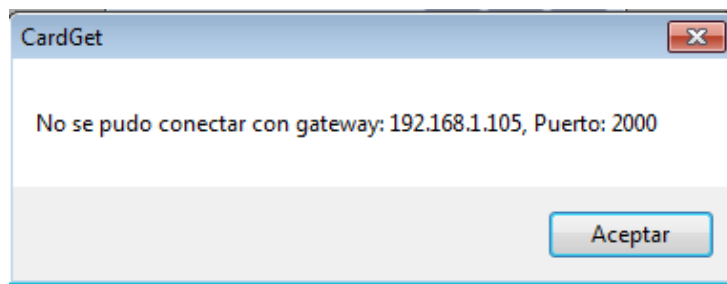


Figura 98 CardGet. Mensaje de error
Fuente: Autores

Después de conectarnos con la dirección IP, se presenta la pantalla principal del programa, el cual consta de tres menús principales, cuyos submenús se muestran a continuación.

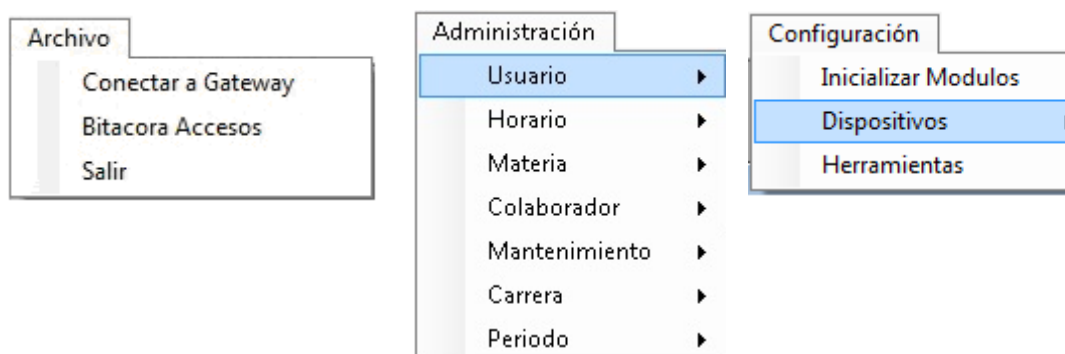


Figura 99 CardGet. Menú contextuales
Fuente: Autores

Dentro del menú **Archivo** tenemos las opciones: Conectar a Gateway, Bitácora Accesos, Salir. Como indica la Figura 99. Donde la opción **Conectar a Gateway**, nos permite conectarnos en caso de que no se haya conectado a la dirección IP anteriormente.

La opción **Bitácora de Accesos**, nos permite ver los accesos que se han registrado en línea. La Figura 100, explica algunos datos ingresados en línea cuando se activan las salidas de cada módulo. Los datos anteriormente almacenados como: fecha de acceso, cédula, número de tarjeta, nombre, apellido, carrera, materia del personal que hace uso del sistema son mostrados en esta ventana. Además da la posibilidad de

exportar el archivo a Excel y puede realizar una actualización manual (limpiar pantalla) por medio de un clic en el botón **Actualizar**.

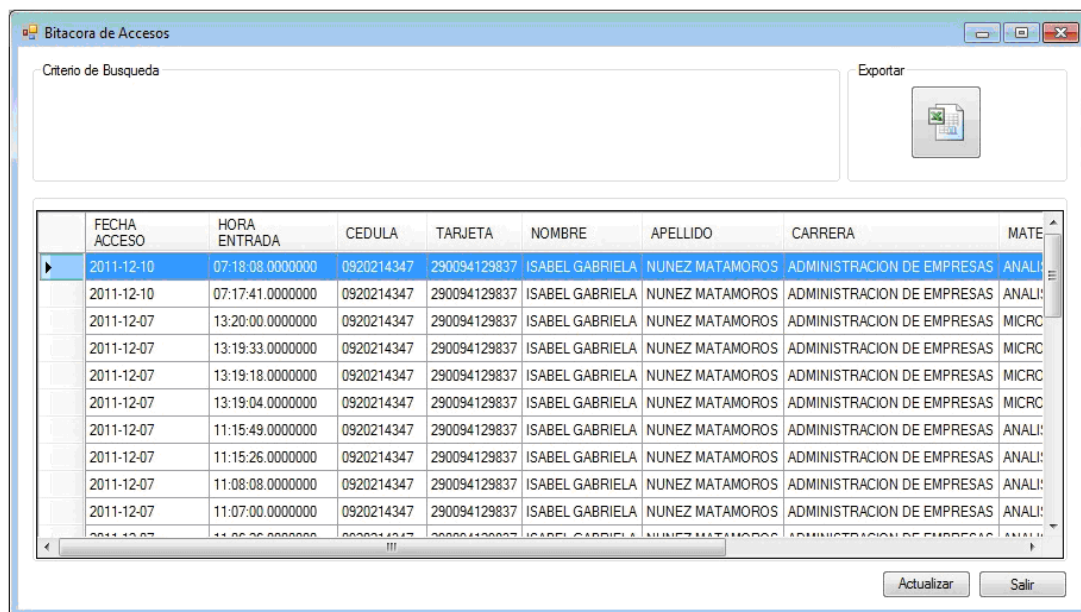


Figura 100 CardGet. Bitácora de Accesos
Fuente: Autores

La última opción dentro del menú Archivo es *Salir*, el cual por consiguiente se encarga de cerrar el programa.

En el menú **Administración** se presenta diferentes opciones como: *Usuario*, *Horario*, *materia*, *colaborador*, *mantenimiento*, *carrera*, *período*.

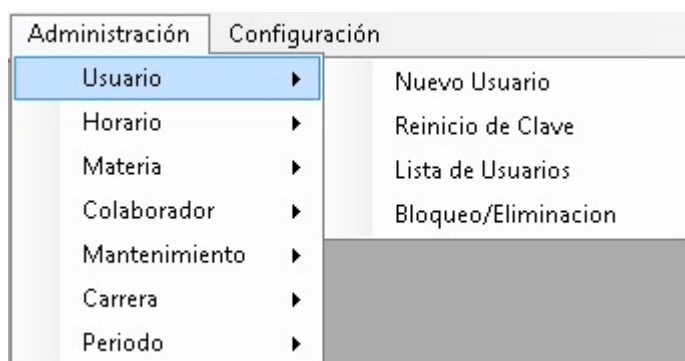


Figura 101 CardGet. Administración. Usuario
Fuente: Autores

Dentro de usuario tenemos lo siguiente:

La opción Nuevo Usuario, nos presenta una ventana (Figura 102) permite dar claves de accesos a diferentes personas para que puedan tener acceso al programa, aquí se ingresan los datos tales como: cédula, usuario, nombre del usuario a ingresar, apellidos del usuario a ingresar, correo electrónico, clave, confirmación de la clave.

También nos da la opción que al primer ingreso de un nuevo usuario, este pueda cambiar la clave, habilitando la opción **Solicitar cambio de clave**.

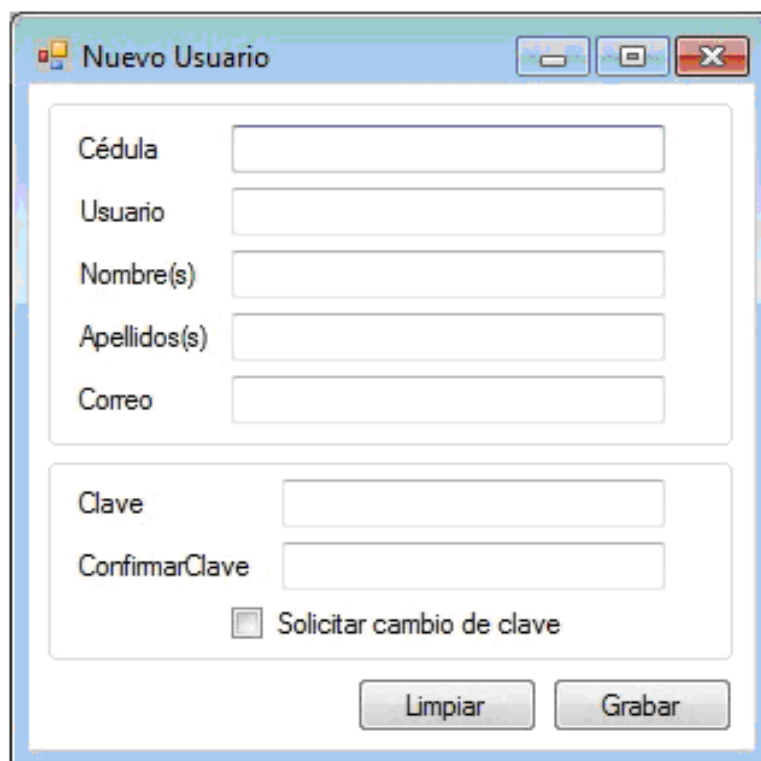


Figura 102 CardGet. Administración. Usuario. Nuevo Usuario
Fuente: Autores

La opción Reinicio de Clave, nos permite como su nombre los indica cambiar la clave de usuario. Como indica la Figura 103.

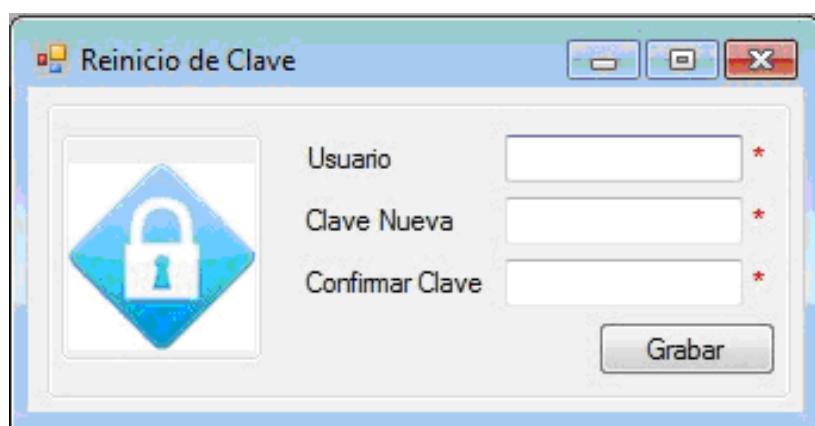



Figura 103 CardGet. Administración. Usuario. Reinicio de clave
Fuente: Autores

La opción Lista de usuarios, nos permite ver los usuarios que se han adherido al software.

	CEDULA	USUARIO	NOMBRE	APELLIDO	EMAIL	STATUS	FECHA DE CREACION	ADMIN
▶	099999999	ADMIN	ADMIN	ADMIN	ADMIN@SALESIANA.COM	0	2011-01-01	ADMIN
	0920214343	CNUNEZ	CARLOS GUSTAVO	NUNEZ MATAMOROS	CNUNEZ@SALESIANA.COM	1	2011-08-06	ADMIN
	0920214349	MAVALOS	MONICA	AVALOS	MAVALOS@GRUPOWONG.COM	0	2011-08-16	ADMIN

Figura 104 CardGet. Administración. Usuario. Lista de usuarios
Fuente: Autores

Con ayuda del ícono  se exporta el archivo a un documento en Excel y poder tener un registro a nivel de usuario. Así lo describe la Figura 105.

	CEDULA	USUARIO	NOMBRE	APELLIDO	EMAIL	STATUS	FECHA DE CREACION	ADMINISTRADOR
5	099999999	ADMIN	ADMIN	ADMIN	ADMIN@SALESIANA.COM	0	2011-01-01	ADMIN
6	0920214343	CNUNEZ	CARLOS GUSTAVO	NUNEZ MATAMOROS	CNUNEZ@SALESIANA.COM	1	2011-08-06	ADMIN
7	0920214349	MAVALOS	MONICA	AVALOS	MAVALOS@GRUPOWONG.COM	0	2011-08-16	ADMIN

Figura 105 CardGet. Archivo Excel generado para Lista de usuarios
Fuente: Autores

La última opción es de Bloqueo/Eliminación, que nos permite eliminar usuarios del programa. En la cual la información requerida para encontrar un usuario es la cédula.

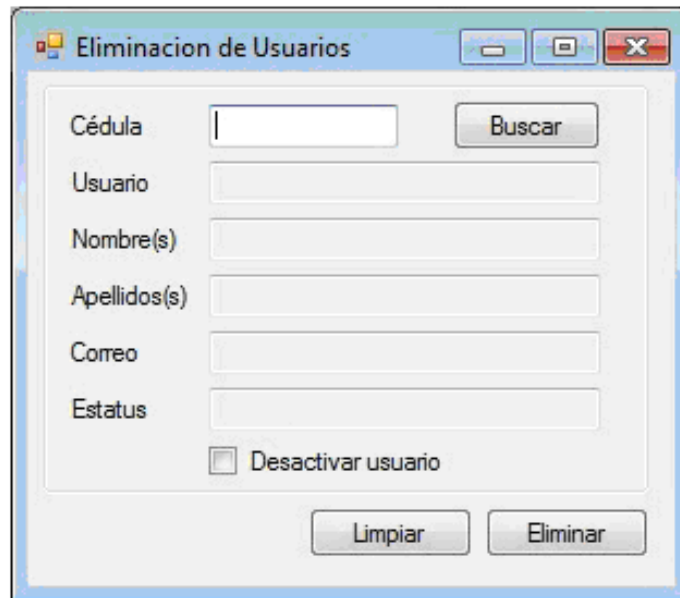


Figura 106 CardGet. Administración. Usuario. Eliminación de usuarios
Fuente: Autores

Dentro de Horario tenemos la opción Nuevo Horario, como nos presenta la Figura 107.

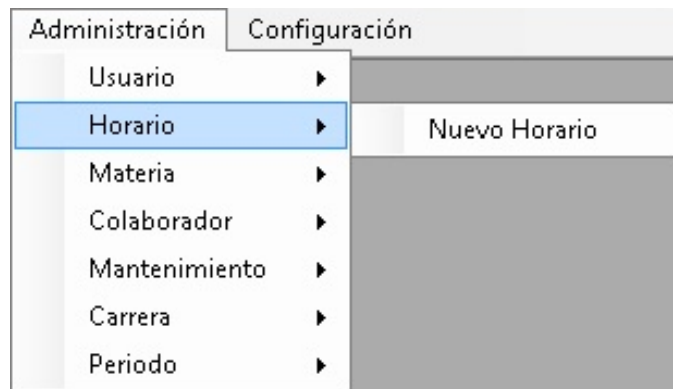


Figura 107 CardGet. Administración. Horario. Nuevo Horario
Fuente: Autores

Al escoger esta opción se presenta una ventana, en donde se ingresará un nuevo horario. Donde nos permite escoger el Periodo, Carrera, Salón, Ubicación y además de exportar a Excel el menú creado, según como indica la Figura 108.

NO	DOMINGO	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
07:00							
08:00							
09:00							
10:00							
11:00							
12:00							
13:00							
14:00							
15:00							
16:00							
17:00							
18:00							
19:00							
20:00							
21:00							

Figura 108 CardGet. Administración. Horario. Nuevo Horario
Fuente: Autores

Siguiendo la ruta Administración/Materia/Nueva materia se ingresará nuevas materias. Por medio de la ventana que se presenta en la Figura 109.

Datos Materia

Materia *

Descripcion *

Limpiar Grabar Salir

Figura 109 CardGet. Administración. Materia. Nueva materia
Fuente: Autores

Dentro de Materia, está la opción Lista de Materias, que nos permite ver un listado de las materias ingresadas con anterioridad.

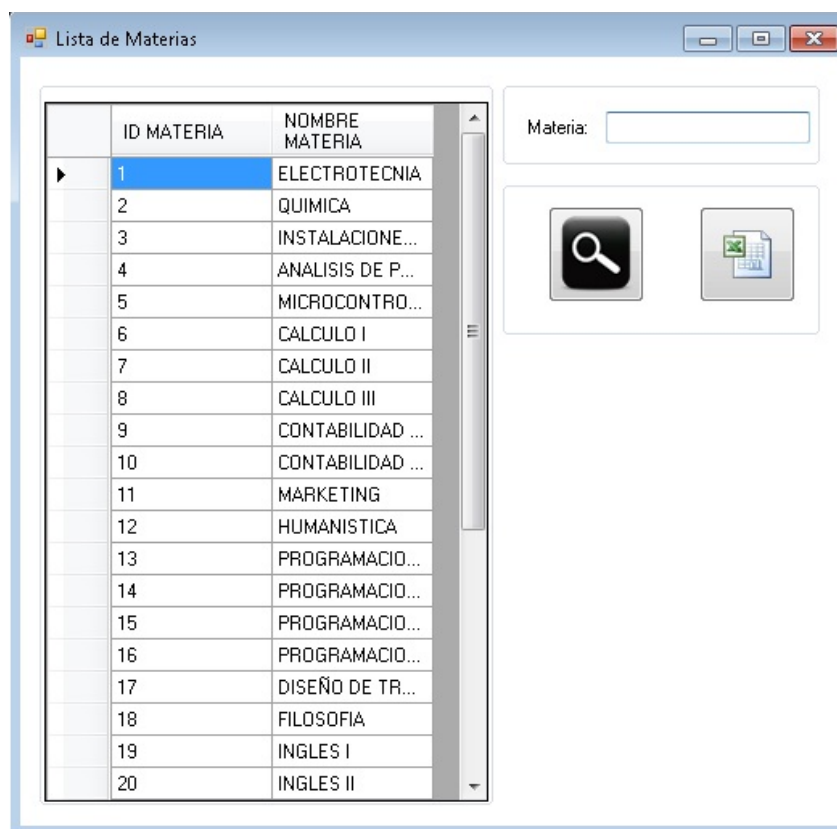


Figura 110 CardGet. Administración. Materia. Lista de Materia
Fuente: Autores

Por consiguiente también se presenta la opción de Eliminar Materias, que busca la materia según el ID Materia, para poder eliminarla.

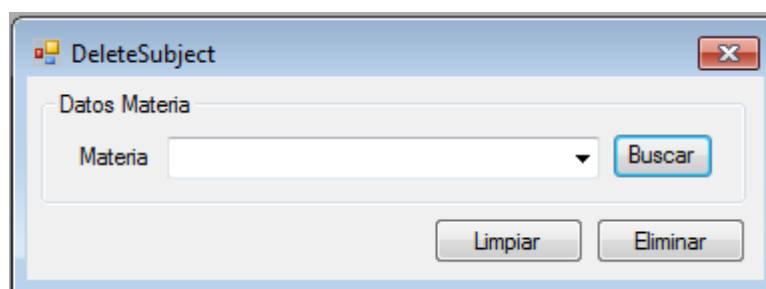


Figura 111 CardGet. Administración. Materia. Eliminar Materias
Fuente: Autores

La opción Nuevo Colaborador permite ingresar nuevo personal como profesores o personas de mantenimiento. Al darle a esta opción, se nos presenta la siguiente

pantalla, Figura 112, la cual nos permite escoger el puerto en el cual se conectará el módulo RFID que nos permitirá leer las tarjetas y asignarlas a una persona.

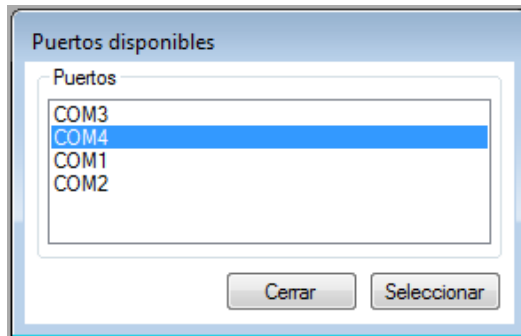


Figura 112 CardGet.Administración.Colaborador. Puertos disponibles
Fuente: Autores

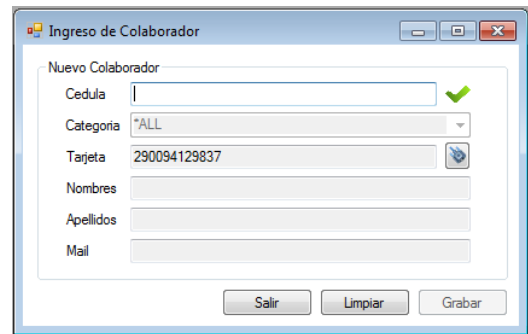


Figura 113 CardGet. Administración. Colaborador. Nuevo colaborador
Fuente: Autores

Al escoger el puerto disponible, aparece la pantalla de la Figura 113, en la cual se ingresan los datos necesarios y al darle en grabar, se ha ingresado una nueva tarjeta

En colaboradores, también se nos muestra la opción Lista de Colaboradores, que nos muestra el personal ingresado (Figura 114), además que nos permite generar un archivo Excel.

	CEDULA	CATEGORIA	TARJETA	NOMBRE	APELLIDO	EMAIL
▶	0920214345	MANTENIMIENTO	290094118C20	DAVID JAVIER	NUNEZ MATAMOROS	DNUNEZ@SALESIANA.COM
	0920214346	MANTENIMIENTO	2500AC133AA0	DIANA GABRIELA	AVILES ZAVALA	DAVILES@SALESIANA.COM
	0920214347	EDUCATIVA	290094129837	ISABEL GABRIELA	NUNEZ MATAMOROS	INUNEZ@SALESIANA.COM
	0920214348	EDUCATIVA	01089592B4BA	JOHNNY MICHAEL	CHONG-QUI SALAZAR	JCHONQUI@SALESIANA.COM
	0920214350	MANTENIMIENTO	01089592B4BA	DANIEL	AMBROSSI TUTIVEN	DAMBROSSI@GRUPOWONG.COM

Figura 114 CardGet. Administración. Colaborador. Lista de Colaboradores
Fuente: Autores

Por consiguiente tenemos la opción de Eliminar Colaborador, que nos permite suprimir un colaborador, como indica la Figura 115.

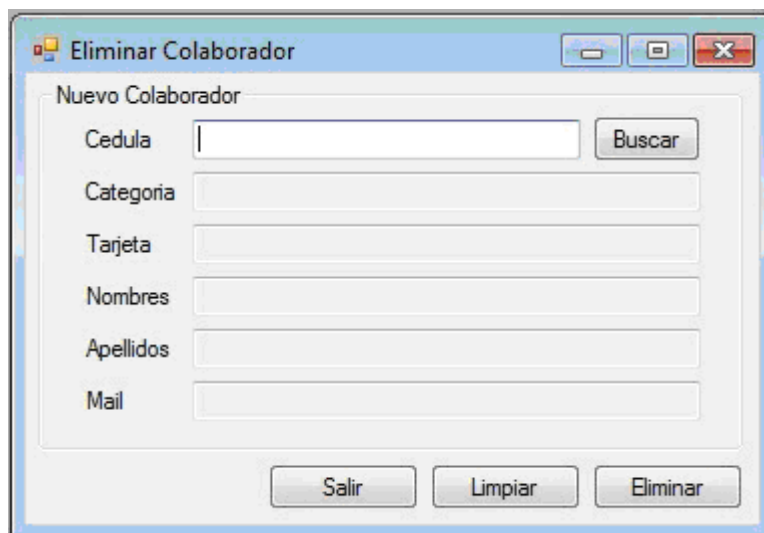


Figura 115 CardGet. Administración. Colaborador. Eliminar Colaborador
Fuente: Autores

Siguiendo la ruta Administración/Mantenimiento, tenemos la opción de Sincronizar, la cual nos muestra la siguiente pantalla Figura 116, para poder crear el archivo de mantenimiento.

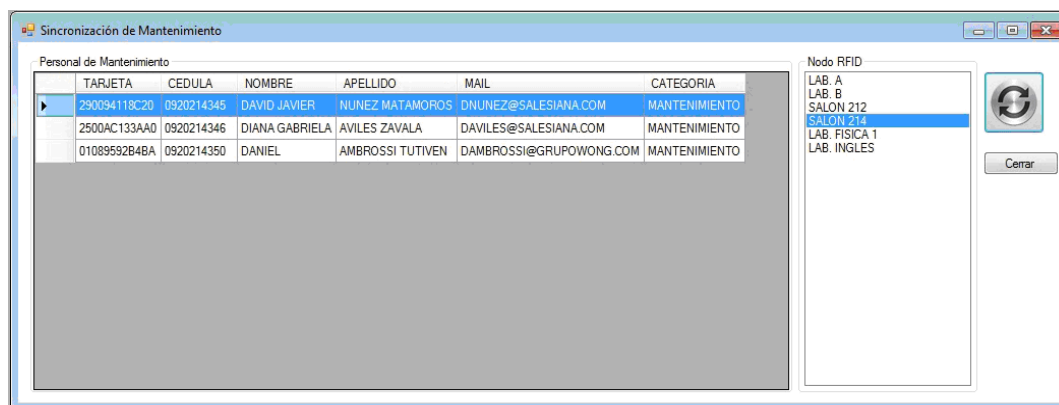


Figura 116 CardGet. Administración. Mantenimiento. Sincronización de mantenimiento
Fuente: Autores

Siguiendo la ruta Administración/Carrera, tenemos diferentes opciones como:
Nueva carrera, Lista de carrera, Eliminar carrera.

La primera ventana Nueva Carrera (Figura 117), permite ingresar el nombre de las carreras presentes en la Universidad.

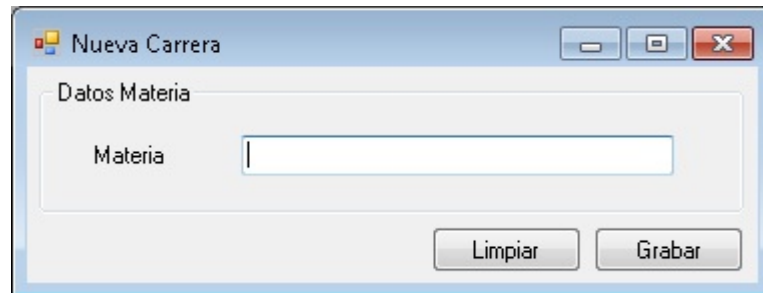


Figura 117 CardGet. Administración. Carrera. Nueva Carrera
Fuente: Autores

La ventana de Lista de Carreras (Figura 118) nos permitirá ver las diferentes carreras ingresadas con anterioridad; además de esto como en opciones anteriores permite generar un archivo de Excel con estos datos.

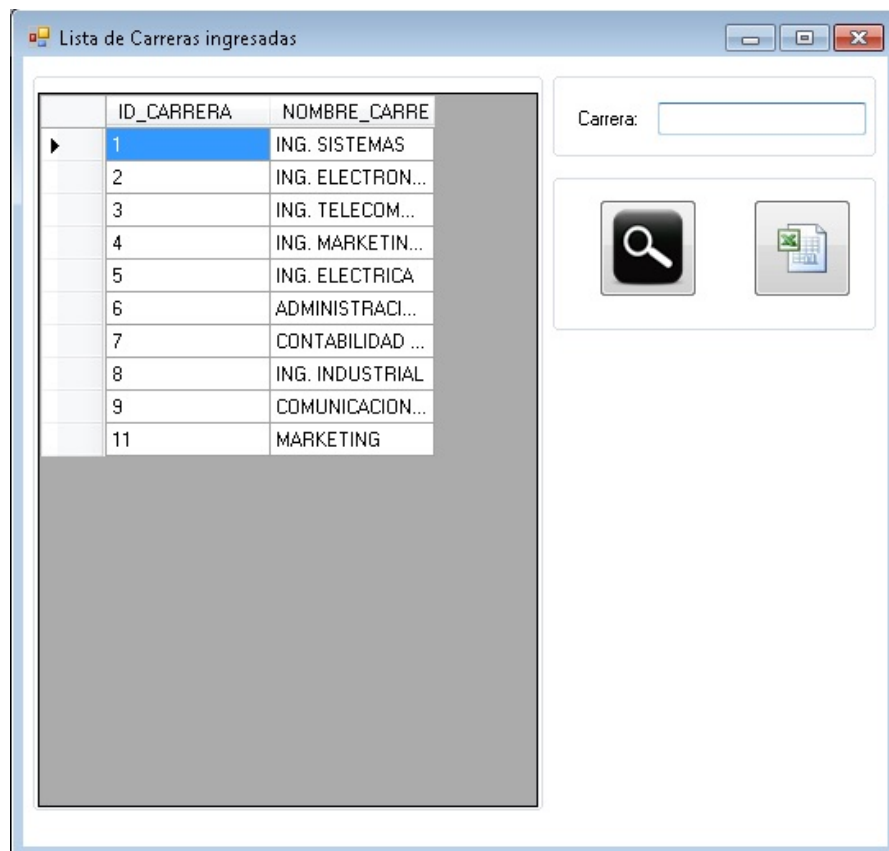


Figura 118 CardGet. Administración. Carrera. Lista de carreras ingresadas
Fuente: Autores

Otra opción es la ventana de Eliminar de Carrera (Figura 119), la cual nos permite eliminar una carrera.

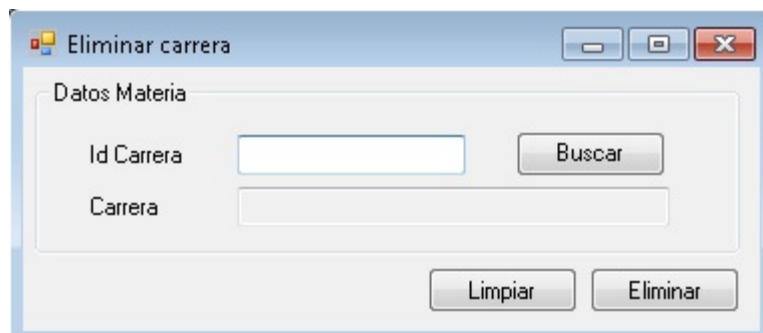


Figura 119 CardGet. Administración. Carrera. Eliminar carrera
Fuente: Autores

Con la ruta Administración/Periodo, se presenta otra ventana denominada Nuevo Período el cual nos permite crear un nuevo periodo. (Figura120)

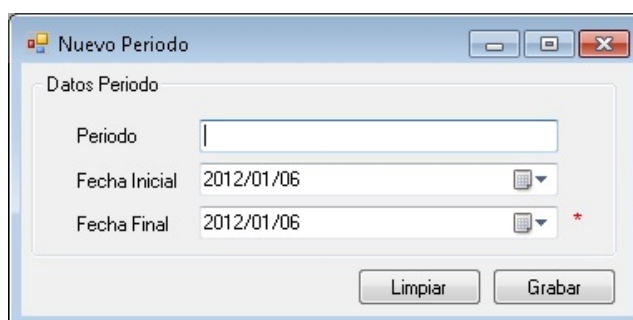


Figura 120 CardGet. Administración. Período. Nuevo período
Fuente: Autores

Dentro de esta misma ruta, se encuentra la opción Eliminar Período, como su nombre lo indica elimina un período.

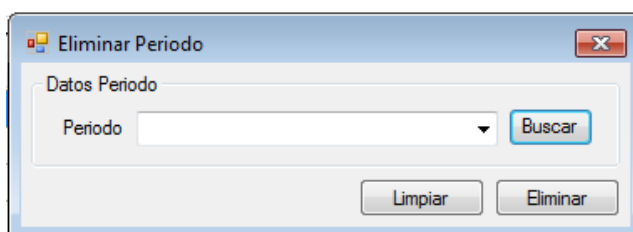


Figura 121 CardGet. Administración. Período. Eliminar período
Fuente: Autores

En el menú de Herramientas dentro de la opción Configuración tenemos las siguientes opciones: Inicializa módulos, Nuevo dispositivos, Herramientas.

A continuación se explica de manera detallada cada uno de ellos.

La opción Inicializa módulos, envía una trama de datos a los módulos para poder habilitarlos en la red.

La opción de Dispositivos, la cual nos da las siguientes submenús:

La pantalla Alta de Equipos, nos permite ingresar nuevos módulos a la red, para poder leerlos, crear horarios, etc.

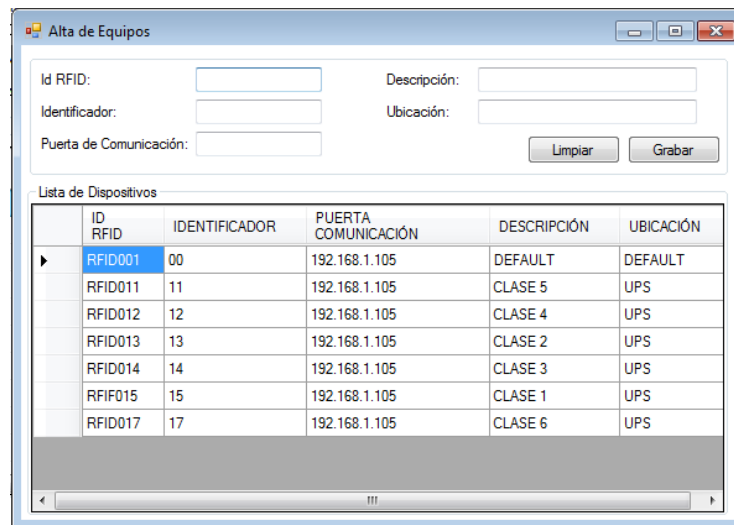


Figura 122 CardGet. Configuración. Dispositivo. Nuevo dispositivo
Fuente: Autores

La pantalla de Listado RFID, nos permite ver los módulos que hemos creado y además poder exportar el archivo a Excel.

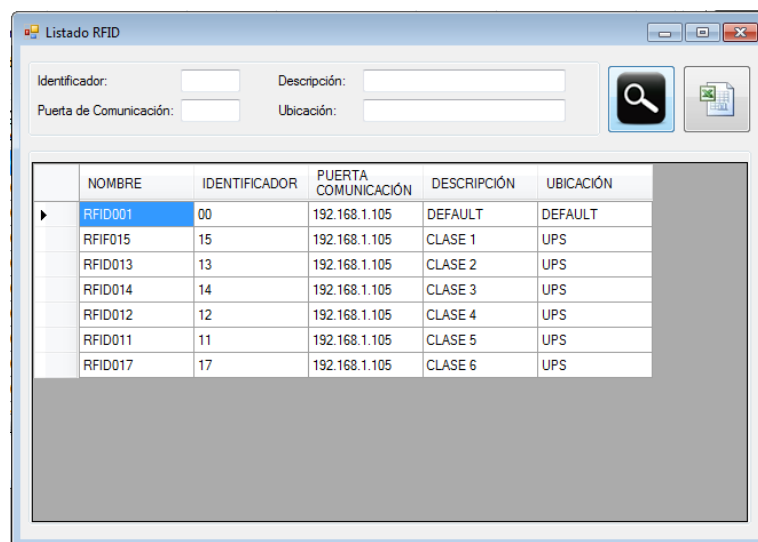


Figura 123 CardGet. Configuración. Dispositivo. Listado RFID
Fuente: Autores

La pantalla Eliminar Dispositivo RFID(Figura 124) nos permite borrar módulos que ya hemos creado anteriormente.



Figura 124 CardGet. Configuración. Dispositivo. Eliminar dispositivo
Fuente: Autores

Dentro de la ruta Configuración/Herramientas se genera una ventana en el cual se interactúa con los módulos.

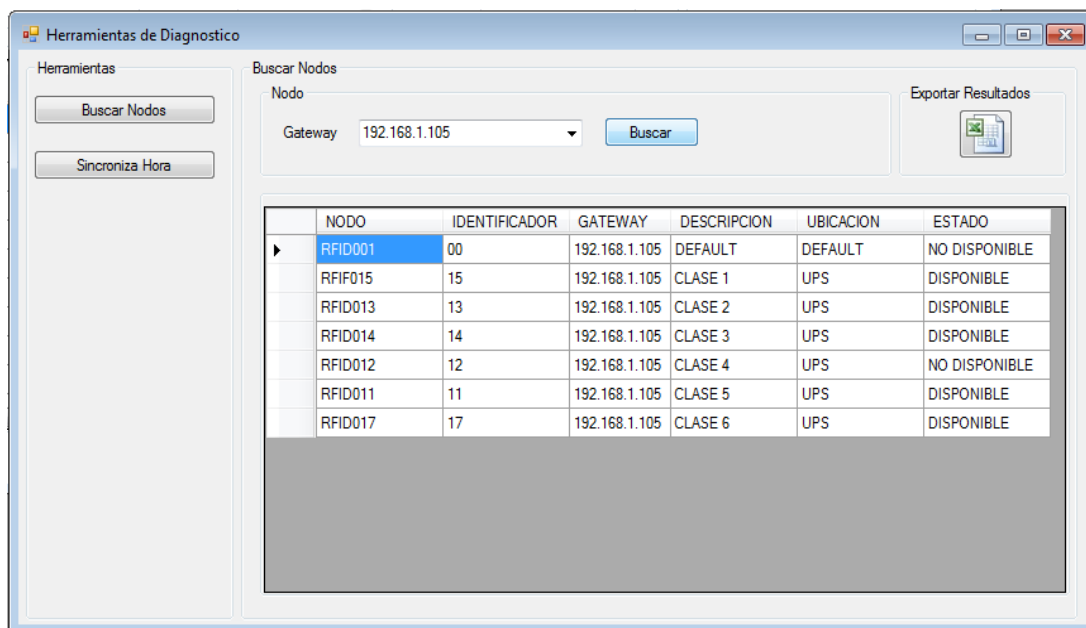
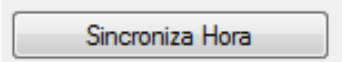


Figura 125 CardGet. Configuración. Herramientas de Diagnóstico
Fuente: Autores

El ícono  permite sincronizar la hora a cada módulo de manera individual.

CAPÍTULO 5

PRÁCTICAS DE LABORATORIO

En este capítulo se detallan las prácticas destinadas para el laboratorio, en las cuales se toman apuntes acerca de la comunicación en general.

5.1 Desarrollo de los módulos para prácticas

Como un trabajo complementario se realizaron dos (2) tarjetas electrónicas para que el estudiante interactúe con los diferentes dispositivos usados en nuestro proyecto.

Entre los diferentes elementos que contiene podemos anotar:

- PIC16F877A, es el principal, ya que en él se guardará la programación de cada práctica y relacionará todos los demás elementos.
- LCD 20X4, presentará los mensajes de acuerdo a lo indicado.
- Memory Stick Datalogger, nuevo elemento capaz de crear archivos en formato XLS usando pen drive como unidad de almacenamiento.
- LM35, sensor de temperatura muy versátil que se incluye en esta tarjeta siendo el receptor de una variable física para trabajar con valores reales.
- SN75176, integrado capaz de usar como protocolo RS485.
- MAX232, chip que genera señales en protocolo RS232.
- Módulo Xbee, capaz de incorporar tecnología inalámbrica en nuestras prácticas.

Todo lo mencionado se puede apreciar en la Figura 126.

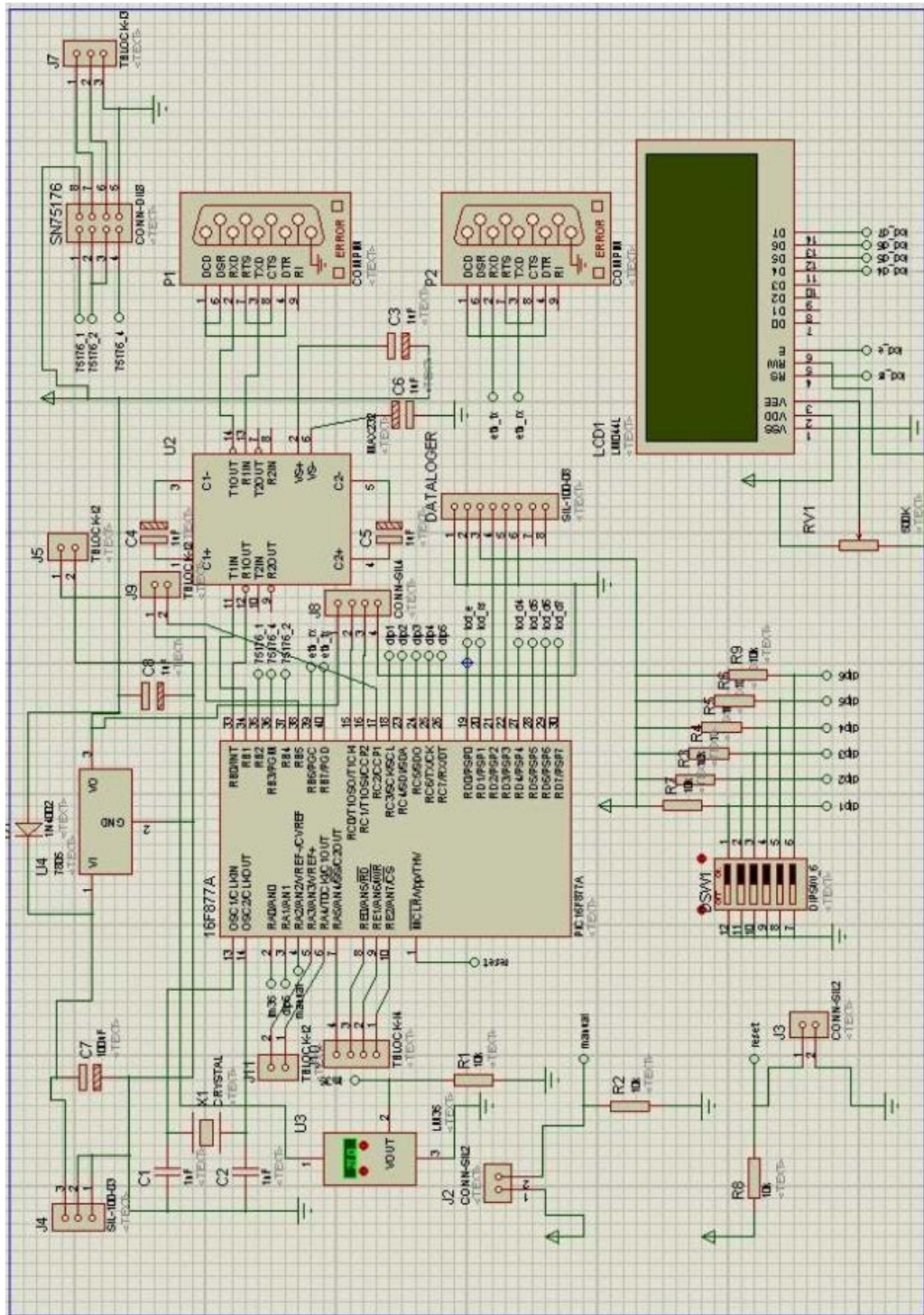


Figura 126 Diseño en PROTEUS. Tarjeta para prácticas
Fuente: Autores

Siguiendo con el desarrollo de los módulos, se presenta un diagrama 3D, en donde se aprecia como quedará el trabajo final de la tarjeta electrónica.

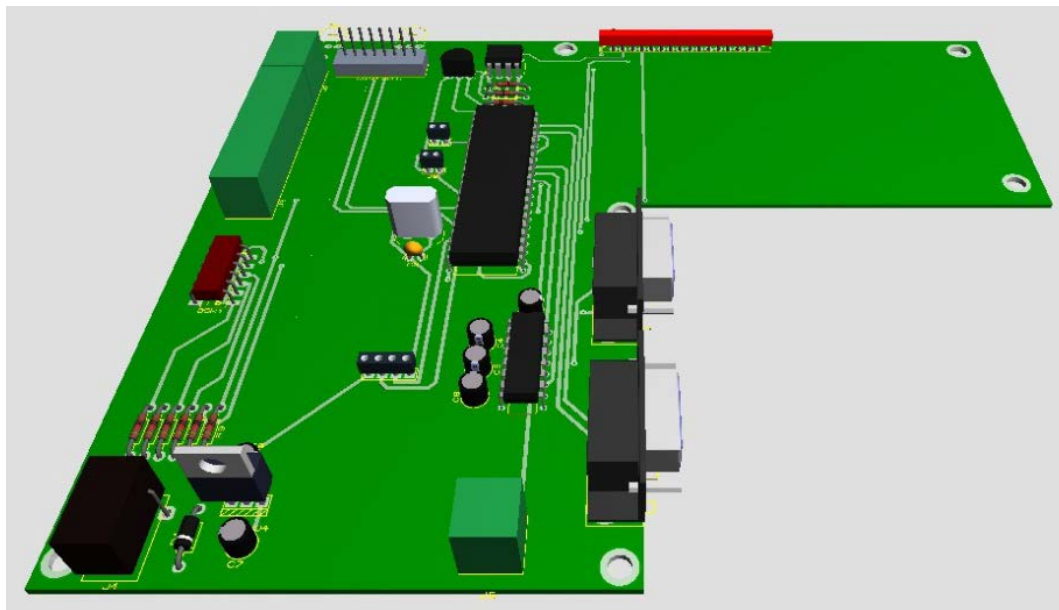
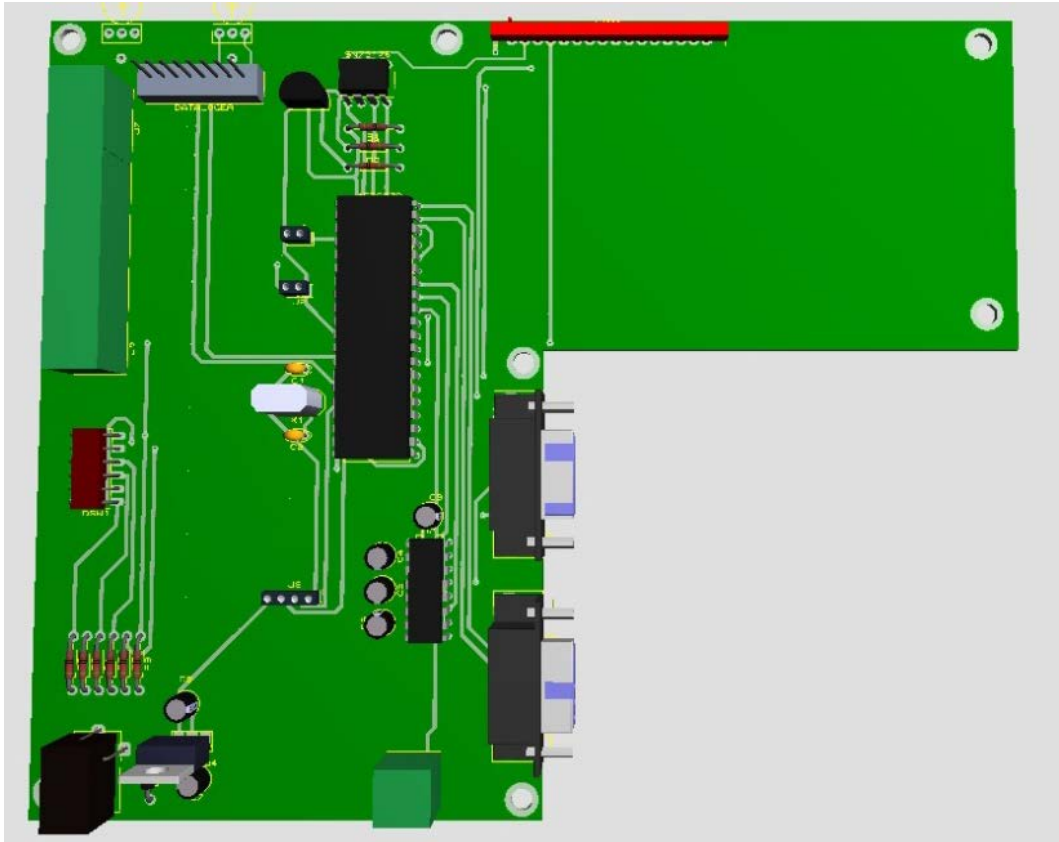


Figura 127 Diseño en PROTEUS. Tarjeta para prácticas Vista 3D
Fuente: Autores

Finalmente se aprecia la foto real de la tarjeta destinada para prácticas de comunicaciones, una vez soldados todos los elementos que intervienen en la tarjeta se dispone a realizar las prácticas propuestas. Siendo éstas un complemento para entender de una manera mas práctica lo que se enseña en el aulas de clase.

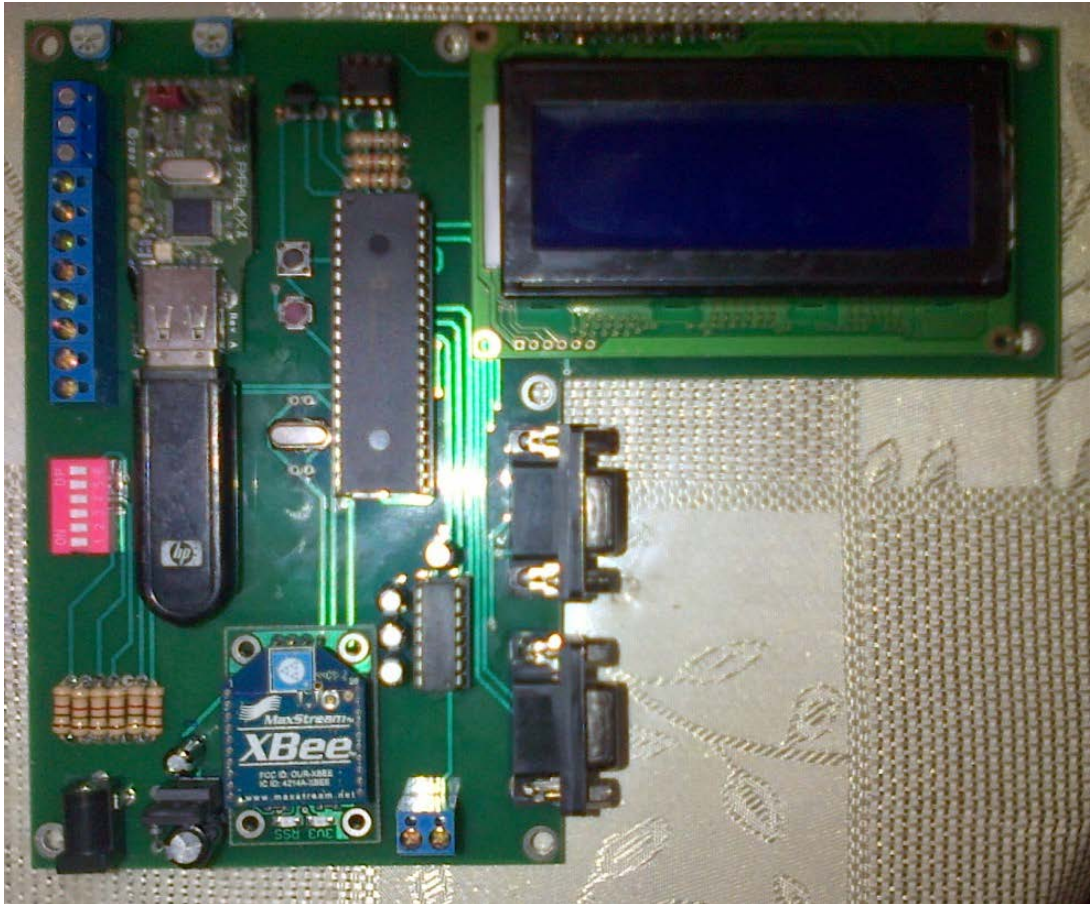



Figura 128 Módulo de tarjeta para prácticas
Fuente: Autores

5.2 Comunicación Serial Asíncrona RS232

	UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL
Formato para Prácticas de Laboratorio.	

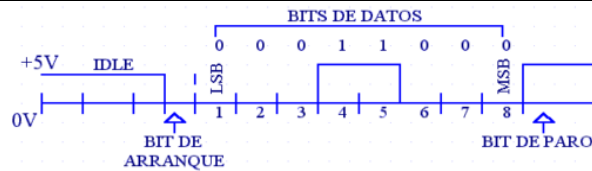
Carrera:	Nombre de la Asignatura	Profesor:
Ingeniería en Electrónica con Mención en Sistemas Industriales	Digitales	Ing. Luis Córdova

Práctica No.	Nombre de la Práctica	Duración (Horas por semana)
1	Comunicación Serial Asíncrona RS232	2

1	INTRODUCCIÓN
En esta práctica el alumno comprenderá la comunicación asíncrona por medio de la norma RS232. Así como también familiarizarse con el software. Además pondrá en aplicación el lenguaje de alto nivel PICBASIC.	

2	OBJETIVOS (COMPETENCIAS)
<p>2.1 Introducir al estudiante al análisis y comprensión de la comunicación asíncrona por medio de la norma RS232.</p> <p>2.2 Mostrar las características operativas de las tarjetas diseñadas para realizar prácticas de comunicación.</p> <p>2.3 Relacionar en conjunto el hardware (tarjetas para prácticas) y software (Hyperterminal).</p>	

3	FUNDAMENTO
<p><i>Comunicación Asíncrona:</i></p> <p>En este tipo de comunicación no son necesarios los pulsos de reloj.</p> <p>La duración de cada bit está determinada por la velocidad con la cual se realiza la transferencia de datos.</p> <p>La siguiente figura muestra la estructura de un carácter que se transmite en forma serial asíncrono.</p>	



Comúnmente la línea del transmisor se encuentra en estado alto. Para iniciar la transmisión, el transmisor coloca esta línea en bajo durante un tiempo, (bit de arranque) a continuación empieza la transmisión con un intervalo de tiempo los bits correspondientes al dato, (empezando por el bit menos significativo).

Si el receptor no está sincronizado con el transmisor, este desconoce cuando se van a recibir los datos; por lo tanto ambos deben tener los mismos parámetros de velocidad, paridad, número de bits del dato y el bit de parada.

Protocolo RS232

Cuando existen distancias cortas se pueden trabajar con niveles lógicos TTL (0V-5V), pero cuando aumentan las distancias, estas señales tienden a distorsionarse debido al efecto capacitivo de los conductores y su resistencia eléctrica.

Con esto se presenta el gran inconveniente de pérdida de información en el flujo de datos.

Ante esta desventaja, surgió la necesidad de crear un acuerdo que permitiera a los equipos de varios fabricantes comunicarse entre sí. La EIA (Electronics Industry Association) elaboró la norma RS232, en la cual se define la interface mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial.

Todas las normas RS232 cumplen con los siguientes niveles de voltaje:

- Un "1" lógico es un voltaje comprendido entre -5v y -15v en el transmisor y entre -3v y -25v en el receptor.
- Un "0" lógico es un voltaje comprendido entre +5v y +15v en el transmisor y entre +3v y +25v en el receptor.

La importancia de conocer esta norma, radica en los niveles de voltaje que maneja el

puerto serial del ordenador, ya que son diferentes a los que se utiliza en los microcontroladores y los demás circuitos integrados. Por lo tanto se necesita de una interfaz que haga posible la conversión a los estándares del TTL.

Este interface es útil para la comunicación punto a punto y velocidad lenta. Por ejemplo, el puerto COM1 de PC para un ratón, el puerto COM2 para un módem, etc. Esto es un ejemplo de la comunicación punto a punto: un puerto y un aparato.

Por lo tanto, la longitud de cable es limitada de 30 a 60 m (el mayor problema es la intervención o resistencia). Brevemente, RS232 es diseñado para la comunicación del aparato de distancia cercana y soporta un transmisor y un receptor.

Circuito MAX232

El circuito integrado soluciona los problemas de niveles de voltaje cuando se requiere enviar unas señales digitales sobre una línea RS232. Este chip se utiliza en aquellas aplicaciones donde no se dispone de fuentes dobles de +12v y -12v. El CI MAX232 necesita solamente una fuente de +5v para su operación, internamente tiene un elevador de voltaje que convierte el voltaje de +5v al de doble polaridad de +12v y -12v, gracias a la ubicación de capacitores externos al circuito.

Software Hyperterminal

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.

Aunque utilizar HyperTerminal con un servicio de boletín electrónico para tener acceso a información de equipos remotos es una práctica que está dejando de ser habitual gracias al World Wide Web, HyperTerminal sigue siendo un medio útil para configurar y probar el módem o examinar la conexión con otros sitios.

HyperTerminal graba los mensajes enviados o recibidos por servicios o equipos situados al otro extremo de la conexión. Por esta razón, puede actuar como una valiosa herramienta para solucionar problemas de configuración y uso del módem.

Para confirmar que el módem está bien conectado o ver su configuración, puede enviar comandos a través de HyperTerminal y ver los resultados. HyperTerminal ofrece la funcionalidad de desplazamiento, que le permite revisar el texto recibido que sobrepase el espacio de la pantalla.

HyperTerminal sirve también para transferir archivos grandes de un equipo a un equipo portátil a través del puerto serie, en lugar de realizar la configuración del portátil en una red.

Puede utilizar HyperTerminal para ayudar a depurar el código fuente desde un terminal remoto. También puede utilizar HyperTerminal para comunicarse con los equipos antiguos basados en caracteres.

HyperTerminal está diseñado para ser una herramienta fácil de utilizar y no viene a sustituir a otras herramientas principales disponibles en el mercado. HyperTerminal puede utilizarse para realizar las tareas específicas descritas.

Norma RS232 con PIC BASIC

Para poder utilizar la norma RS232 se debe incluir una librería **INCLUDE “modedef.bas”** (modos de comunicación), aquí se encuentran algunos de los parámetros para las comunicaciones, por ejemplo en nuestro caso las velocidades de transmisión que son para dato invertido: N300, N1200, N2400, N9600, y para dato verdadero: T300, T1200, T2400, T9600.

Los datos invertidos por ejemplo el N2400, quiere decir que un 1 lógico vale 0V y un 0 Lógico vale 5V, en cambio para dato verdadero por ejemplo el T2400 el 1 lógico vale 5V y el 0 lógico vale 0V.

LA DECLARACIÓN SEROUT.

```
SEROUT portb.1, N2400,["D"] ;
```

Esta declaración sirve para enviar datos seriales en un formato estándar asincrónico

usando 8 bits de dato, sin paridad y 1 stop bit, (8N1), para nuestro ejemplo indica el envío del carácter ASCII “D” por el puerto B1 a 24008 N1, en dato invertido.

LA DECLARACIÓN SERIN.

SERIN portb.0,N2400,letra

Esta declaración sirve para recibir datos seriales en un formato estándar asincrónico usando 8 bits de dato, sin paridad y 1 stop bit, (8N1), para nuestro ejemplo la sintaxis descrita espera un dato serial y lo guarda en la variable previamente denominada “letra”.

Cabe recalcar que esta declaración detiene el programa esperando el bit de inicio, y solamente cuando haya recibido un dato, continúa con la siguiente línea de programa.

Para hacer que sólo espere un determinado tiempo y luego continúe con la siguiente línea debe utilizar el comando Timeout, o utilizar una interrupción externa para atender el dato a recibir.

4	PROCEDIMIENTO (DESCRIPCIÓN)	
4.1	EQUIPO NECESARIO	MATERIAL DE APOYO
	<ul style="list-style-type: none"> ✓ Convertidor RS232 a USB Módulos de práctica en la cual consta de: <ul style="list-style-type: none"> ✓ 1 Microcontrolador PIC16F877A ✓ 1 LCD 16X4 ✓ 1 Conector DB9 ✓ 1 Memory Stick Datalogger ✓ 1 MAX232 ✓ 6 Entradas (Dipswitch) ✓ 6 Salidas LED ✓ LM35 (Sensor de Temperatura) 	Proyector Pizarra Líquida

4.2 DESARROLLO DE LA PRÁCTICA

Actualmente muy pocos ordenadores cuentan con puerto serial RS232, pero para poder entender este tipo de protocolo usaremos un convertidor RS232 a USB.

4.2.1 Realizar un programa en PIC BASIC capaz de enviar datos de temperatura con ayuda del LM35 al software Hyperterminal, estos datos serán enviados de manera automática.

;Definición del Ucontrolador

```
@DEVICEXT_OSC      ;Oscilador de Cristal de Cuarzo
define osc 4        ;Cristal de cuarzo de 4Mhz
```

;Pines de comunicación del LCD

```
define      LCD_DREG  PORTD
define      LCD_DBIT  4
define      LCD_RSREG PORTD
define      LCD_RSBIT 1
define      LCD_EREG  PORTD
define      LCD_EBIT  0
DEFINE     LCD_BITS 4
DEFINE     LCD_LINES 4
DEFINE     LCD_COMMANDUS 2000
DEFINE     LCD_DATAUS 50
```

;Entradas/Salidas de Comunicación Serial Max232

```
Rx_232  var  portb.0
Tx_232  var  portb.1
```

;Variable de Temperatura

```
Temp_Byte  var  byte
Temperatura  var  word
Decimal    var  byte
```

;Declaración de puertos

```
trisa = %00000111
trisb = $00000001
trisd = %00000000      ;Puerto del LCD
adcon1 = %00001110     ;Configuración de Pines analógicos
temp_Byte = 0
Temperatura = 0
```

```
LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.1"
```

```
main:
```

```
Auto:  ADCIN 0, Temp_Byte      ;LEO Temperatura
        Temperatura = (19*Temp_Byte)/10
        Decimal = Temperatura dig 0
gotoCal_Temp
```

;Muestra Temperatura en LCD

Cal_Temp: if Temperatura <= 9 then

gotoUnidad

else

If (Temperatura >= 10 and Temperatura <=99)then

gotodecenas

else

If (Temperatura >= 100)then

gotoCentenas

goto main

endif

endif

endif

Unidad:

LCDOUT \$FE,\$0C0,"Temp. = ",dec1 Temperatura," ",dec decimal," C "

gosub env_unidad_232

PAUSE 1000

goto main

Decenas:

LCDOUT \$FE,\$0C0,"Temp. = ",dec2 Temperatura," ",dec decimal," C "

gosub env_decenas_232

PAUSE 1000

gotomain

Centenas:

LCDOUT \$FE,\$0C0,"Temp. = ",dec3 Temperatura," ",dec decimal," C "

gosub env_centenas_232

PAUSE 1000

gotomain

;Envío Temperatura por RS232

Env_Unidad_232: serout2 TX_232,84,["Temp. = ",dec1 Temperatura," ",dec decimal," C"]

Return

Env_Decenas_232: serout2 TX_232,84,["Temp. = ",dec2 Temperatura," ",dec decimal,"

C"]

Return

Env_Centenas_232: serout2 TX_232,84,["Temp. = ",dec3 Temperatura," ",dec decimal,"

C"]

Return

End

4.2.2 Con ayuda del software Hyperterminal se enviará la Letra “Y” hacia el modulo de practica. El programa que se desarrollará en PIC BASIC será capaz de recibir este dato, y así mismo enviar el valor de la temperatura del LM35 hacia el software Hyperterminal.

;Definición del Ucontrolador

```
@DEVICEEXT_OSC      ;Oscilador de Cristal de Cuarzo
define osc 4         ;Cristal de cuarzo de 4Mhz
```

;Pines de comunicación del LCD

```
define      LCD_DREG  PORTD
define      LCD_DBIT  4
define      LCD_RSREG PORTD
define      LCD_RSBIT 1
define      LCD_EREG  PORTD
define      LCD_EBIT  0
DEFINE     LCD_BITS 4
DEFINE     LCD_LINES 4
DEFINE     LCD_COMMANDUS 2000
DEFINE     LCD_DATAUS 50
```

;Entradas/Salidas de Comunicación Serial Max232

```
Rx_232 var portb.0
Tx_232 var portb.1
```

;Variable de Temperatura

```
Temp_Byte var byte
Temperatura var word
Decimal var byte
```

;Variables Generales

```
Letra var byte
```

;Declaración de puertos

```
trisa = %00000111
trisb = $00000001
trisd = %00000000      ;Puerto del LCD
adcon1 = %00001110    ;Configuración de Pines analógicos
```

```
temp_Byte = 0
Temperatura = 0
```

```
LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.2"
```

```
main:
```

```
Y:      SERIN2 Rx_232,84,[Letra]
```

```

if Letra == "Y" then
goto Auto
    else
gotomain
endif
Auto:    ADCIN 0, Temp_Byte; Se lee Temperatura
        Temperatura = (19*Temp_Byte)/10
        Decimal = Temperatura dig 0
gotoCal_Temp

;Muestro Temperatura en LCD
Cal_Temp: if Temperatura <= 9 then
gotoUnidad
    else
        If (Temperatura >= 10 and Temperatura <=99)then
gotodecenas
    else
        If (temperatura>= 100)then
gotoCentenas
goto main
endif
endif
endif

Unidad:  LCDOUT $FE,$0C0,"Temp. = ",dec1 Temperatura," ",dec decimal," C "
gosub env_unidad_232
        PAUSE 1000
gotomain

Decenas:  LCDOUT $FE,$0C0,"Temp. = ",dec2 Temperatura," ",dec decimal," C "
gosub env_decenas_232
PAUSE 1000
gotomain

Centenas:  LCDOUT $FE,$0C0,"Temp. = ",dec3 Temperatura," ",dec decimal," C "
gosub env_centenas_232
        PAUSE 1000
gotomain

;Envío Temperatura por RS232

Env_Unidad_232:  serout2 TX_232,84,["Temp. = ",dec1 Temperatura," ",dec decimal," C"]
return
Env_Decenas_232:  serout2 TX_232,84,["Temp. = ",dec2 Temperatura," ",dec decimal,"
C"]
return
Env_Centenas_232:  serout2 TX_232,84,["Temp. = ",dec3 Temperatura," ",dec decimal,"
C"]
return
END

```

4.2.3 En base a los conocimientos adquiridos realizar un programa en PICBASIC, capaz de interactuar las dos tarjetas de prácticas de la siguiente manera:

* Enviar el valor real de temperatura (LM35) desde la tarjeta de práctica #1 y presentarlos en el LCD de la tarjeta de práctica #2. De esta manera la tarjeta de práctica #1 actúa como transmisor y la tarjeta de práctica # 2 como receptor.

Programación para Tarjeta de Práctica # 1 (TRANSMISOR)

;Definición del Ucontrolador

```
@DEVICEXT_OSC      ;Oscilador de Cristal de Cuarzo
define osc 4        ;Cristal de cuarzo de 4Mhz
```

;Pines de comunicación del LCD

```
define LCD_DREG PORTD
define LCD_DBIT 4
define LCD_RSREG PORTD
define LCD_RSBIT 1
define LCD_EREG PORTD
define LCD_EBIT 0
DEFINE LCD_BITS 4
DEFINE LCD_LINES 4
DEFINE LCD_COMMANDUS 2000
DEFINE LCD_DATAUS 50
```

;Entradas/Salidas de Comunicación Serial Max232

```
Rx_232 var portb.0
Tx_232 var portb.1
```

;Variable de Temperatura

```
Temp_Byte var byte
Temperatura var word
Decimal var byte
```

;Declaración de puertos

```
trisa = %00000111
trisb = $00000001
trisd = %00000000      ;Puerto del LCD
adcon1 = %00001110    ;Configuración de Pines analógicos
```

```
temp_Byte = 0
Temperatura = 0
```

```
LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.3"
```

```
main:
```

```
Auto:   ADCIN 0, Temp_Byte; Se lee dato de temperatura
        Temperatura = (19*Temp_Byte)/10
        Decimal = Temperatura dig 0
gotoCal_Temp
```

;Muestro Temperatura en LCD

```
Cal_Temp: if temperatura<= 9 then
gotoUnidad
```

```

else
  If (temperatura>= 10 and Temperatura <=99)then
gotodecenas
  else
  If (temperatura>= 100)then
gotoCentenas
goto main
endif
endif
endif

Unidad: LCDOUT $FE,$0C0,"Temp. = ",dec1 Temperatura,",",dec decimal," C "
gosub env_unidad_232
  PAUSE 1000
gotomain

Decenas: LCDOUT $FE,$0C0,"Temp. = ",dec2 Temperatura,",",dec decimal," C "
gosub env_decenas_232
PAUSE 1000
gotomain

Centenas: LCDOUT $FE,$0C0,"Temp. = ",dec3 Temperatura,",",dec decimal," C "
gosub env_centenas_232
  PAUSE 1000
gotomain

;Envío Temperatura por RS232

Env_Unidad_232: serout2 TX_232,84,["Temp. = ",dec1 Temperatura,",",dec decimal," C"]
return

Env_Decenas_232: serout2 TX_232,84,["Temp. = ",dec2 Temperatura,",",dec decimal,"
C"]
return

Env_Centenas_232: serout2 TX_232,84,["Temp. = ",dec3 Temperatura,",",dec decimal,"
C"]
return

END

Programación para Tarjeta de Práctica # 2 (RECEPTOR)

;Definición del Ucontrolador
@DEVICEXT_OSC ;Oscilador de Cristal de Cuarzo
define osc 4 ;Cristal de cuarzo de 4Mhz

;Pines de comunicación del LCD
define LCD_DREG PORTD
define LCD_DBIT 4

```

```

define      LCD_RSREG  PORTD
define      LCD_RSBIT  1
define      LCD_EREG   PORTD
define      LCD_EBIT   0
DEFINE     LCD_BITS 4
DEFINE     LCD_LINES 4
DEFINE     LCD_COMMANDUS 2000
DEFINE     LCD_DATAUS 50

;Entradas/Salidas de Comunicación Serial Max232
Rx_232  var  portb.0
Tx_232  var  portb.1

;Variables de Lectura de Temperatura
Lee_Tempe  var  word
Lee_Decimal  var  byte

;Declaración de puertos
trisa = $00000001
trisd = %00000000

LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.3"

main:
        SERIN2 Rx_232,84,2000,main,[Lee_Tempe,Lee_decimal]
Lee_tempe = Lee_Tempe - 48
lee_decimal = lee_decimal - 48
LCDOUT $FE,$0C0,"Temp. = ",dec lee_tempe,",",dec lee_Decimal," C"
        pause 1000


goto main
END

```

5	REFERENCIAS
---	-------------

http://technet.microsoft.com/es-es/library/cc736511%28v=ws.10%29.aspx

5.3 Uso del Memory Stick Datalogger

	UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL
Formato para Prácticas de Laboratorios	

Carrera:	Nombre de la Asignatura	Profesor:
Ingeniería en Electrónica con Mención en Sistemas Industriales	Digitales	Ing. Luis Córdova

Práctica No.	Nombre de la Práctica	Duración (Horas por semana)
2	Uso del Memory Stick Datalogger	2

1	INTRODUCCIÓN
<p>En esta práctica el alumno comprenderá la comunicación asíncrona por medio de un nuevo dispositivo electrónico capaz de almacenar información conocido como Memory Stick Datalogger, aplicando el lenguaje de alto nivel PICBASIC.</p>	

2	OBJETIVOS (COMPETENCIAS)
<p>2.1 Comprobar experimentalmente la comunicación asíncrona, por medio de comandos elaborados en PICBASIC.</p> <p>2.2 Reconocer y comprender el funcionamiento de un Memory Stick Datalogger, como un nuevo dispositivo electrónico que se puede aplicar en diferentes campos.</p>	

3	FUNDAMENTO
<p>El Memory Stick Datalogger es un puente de host USB que permite conectar un dispositivo de almacenamiento masivo USB, como una unidad flash, a tu microcontrolador. El IC Vinculum / Firmware en el módulo de registro de datos maneja el sistema de archivos de la Memory Stick para que pueda compartir los archivos con su PC. Todo este control lo hace mediante simples comandos de serie.</p> <p><u>Características:</u></p> <ul style="list-style-type: none"> ✓ Simple interfaz serie o SPI para microcontroladores. 	

- ✓ Mando Extendido o Short Form-Set / Respuestas.
- ✓ Bajo consumo de energía de operación (25mA Run/2mA espera).
- ✓ Fácil actualización del firmware.

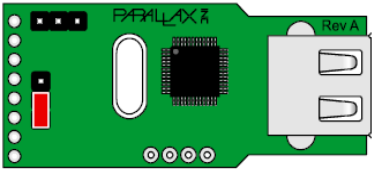
Especificaciones:

Símbolo	Descripción	Mínimo	Típico	Máximo	Unidades
Vdd	Voltaje de alimentación	4.75	5	5.25	V
	Temperatura de almacenamiento	-65 0		150 0	C
	Temperatura de operación	0 0		70 0	C
Ivdd	Corriente de operación		25		mA
Ivdd	Corriente de standby	1	2	2	mA

Pines de Conexión

A continuación se presenta los pines para usarlo con modo UART

Pines	Nombre	Descripción
1	Vss	Conectar a tierra del sistema
2	RTS #	RequestToSend
3	Vdd	Conectar a 5 V
4	RXD	Recibir datos
5	TXD	Transmitir datos
6	CTS#	Clear ToSend
7	NC	No Conectar
8	RI#	Ring Indicator



En el capítulo 2 en la Tabla 2.4 se describen los comandos necesarios para usar el Memory Stick Datalogger.

Memory Stick Datalogger con PIC BASIC

Entre las especificaciones del Memory Stick indica que podemos usarlo en modo serial, por lo tanto se trabajará con la UART del microcontrolador.

Tomando como referencia algunas comandos citados en capítulos anteriores se describe lo siguiente:

<i>Comando</i>	<i>Descripción</i>
'OPR' <sp><name><cr>	Abre un archivo para leerlo, 'OPR' Comando, <sp>Espacio, <name>Nombre del archivo, <cr>Enter, Ejemplo: OPR PRUEBA.TXT

A continuación se escribe la línea de programación en PIC BASIC.

```
SEROUT2 PORTD.3,84,["OPR PRUEBA.TXT", $0D]
```

Esto me indica que el microcontrolador envía mediante vía serial al Memory Stick Datalogger la sintaxis del comando empleado para abrir un archivo de nombre PRUEBA.TXT.

<i>Comando</i>	<i>Descripción</i>
'CLF' <sp><name><cr>	Abre un archivo para leerlo, 'CLF' Comando, <sp>Espacio, <name>Nombre del archivo, <cr>Enter, Ejemplo: CLF PRUEBA.TXT

A continuación se escribe la línea de programación en PIC BASIC.

```
SEROUT2 LOGOUT,84,["CLF PRUEBA.TXT", $0D]
```

Esto me indica que el microcontrolador envía mediante vía serial al Memory Stick Datalogger la sintaxis del comando empleado para cerrar un archivo de nombre PRUEBA.TXT.

4	PROCEDIMIENTO (DESCRIPCIÓN)																
4.1	EQUIPO NECESARIO	MATERIAL DE APOYO															
	Módulos de práctica en la cual consta de: 1 Microcontrolador PIC16F877A 1 LCD 16 X 4 1 Conector DB9 1 Memory Stick Datalogger 1 MAX232 6 Entradas (Dipswitch) 6 Salidas LED LM35 (Sensor de Temperatura)	Proyector Pizarra Líquida															
4.2	DESARROLLO DE LA PRÁCTICA																
<p>4.2.1 Desarrollar una lógica de programación capaz de transmitir la variación de temperatura del LM35 serialmente (RS232), con agentes externos se variará la temperatura alrededor del LM35 y se tomarán valores de temperatura cada 10 segundos y estos valores serán almacenados en el memory stick datalogger en una hoja de cálculo (Excel) de la siguiente manera:</p> <table border="1" data-bbox="662 1265 1045 1556"> <tr> <td>Muestra</td> <td>35</td> <td>C</td> </tr> <tr> <td>Muestra</td> <td>45</td> <td>C</td> </tr> <tr> <td>Muestra</td> <td>50</td> <td>C</td> </tr> <tr> <td>Muestra</td> <td>55</td> <td>C</td> </tr> <tr> <td>Muestra</td> <td>60</td> <td>C</td> </tr> </table> <p>Cabe recalcar que se debe realizar un programa en PICBASIC para interactuar con el memory stick datalogger.</p> <p>;Definición del Ucontrolador @DEVICEXT_OSC ;Oscilador de Cristal de Cuarzo define osc 4 ;Cristal de cuarzo de 4Mhz</p> <p>;Pines de comunicación del LCD define LCD_DREG PORTD define LCD_DBIT 4 define LCD_RSREG PORTD define LCD_RSBIT 1</p>			Muestra	35	C	Muestra	45	C	Muestra	50	C	Muestra	55	C	Muestra	60	C
Muestra	35	C															
Muestra	45	C															
Muestra	50	C															
Muestra	55	C															
Muestra	60	C															

```

define      LCD_EREG  PORTD
define      LCD_EBIT  0
DEFINE     LCD_BITS  4
DEFINE     LCD_LINES 4
DEFINE     LCD_COMMANDUS 2000
DEFINE     LCD_DATAUS 50

;Entradas/Salidas de Comunicación Serial Max232
Rx_232    var    portb.0
Tx_232    var    portb.1

;Variables para Manejo de Datalogger
Cant_Datos var    byte
Data_Temp  var    Portd.2

;Variable de Temperatura
Temp_Byte  var    byte
Temperatura var    word
Decimal    var    byte

;Variables Generales
tiempo     var    byte
Num_Muestra var    byte

;Declaración de puertos
trisa = %00000111
trisb = $00000001
trisd = %00000000      ;Puerto del LCD
adcon1 = %00001110    ;Configuración de Pines analógicos

temp_Byte = 0
Temperatura = 0
Num_Muestra = 0

pause 1000

SEROUT2 Data_Temp,84,[$0D]; EnceroDatallogger

LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.1"

main:

Auto:    ADCIN 0, Temp_Byte; Se lee valor de Temperatura
         Temperatura = (19*Temp_Byte)/10
         Decimal = Temperatura dig 0
         Num_Muestra = Num_Muestra + 1
         gotoCal_Temp

;Muestro Temperatura en LCD
Cal_Temp: if temperatura <= 9 then
gotoUnidad
         else
         If (temperatura>= 10 and Temperatura <=99)then

```

```

gotodecenas
    else
        If (temperatura>= 100)then
gotocentenas
goto main
endif
endif
endif

Unidad:  LCDOUT $FE,$0C0,"Temp. = ",dec1 Temperatura," ",dec decimal," C "
gosub env_unidad_232
gosubGrab_Unidad

for tiempo = 1 to 10
    pause 1000
next

goto main

Decenas:  LCDOUT $FE,$0C0,"Temp. = ",dec2 Temperatura," ",dec decimal," C "
gosub env_decenas_232
gosubGrab_Decena

for tiempo = 1 to 10
    pause 1000
next

goto main

Centenas:  LCDOUT $FE,$0C0,"Temp. = ",dec3 Temperatura," ",dec decimal," C "
gosub env_centenas_232
gosubGrab_Centena

for tiempo = 1 to 10
    pause 1000
next

goto main

;Envio Temperatura por RS232

Env_Unidad_232:  serout2 TX_232,84,["Temp. = ",dec1 Temperatura," ",dec decimal," C"]
return

Env_Decenas_232:  serout2 TX_232,84,["Temp. = ",dec2 Temperatura," ",dec decimal,"
C"]
return

Env_Centenas_232:  serout2 TX_232,84,["Temp. = ",dec3 Temperatura," ",dec decimal,"
C"]
return

;Grabación de Temperatura de Datalogger

Grab_Unidad:SEROUT2 Data_Temp,84,["OPW Env_Temperatura.xls", $0D]

```

```

PAUSE 20
    SEROUT2 Data_Temp,84,["WRF 16", $0D]
PAUSE 20
    SEROUT2 Data_Temp,84,["Muestra", " = ", Dec1 Temperatura, "", DEC Decimal, "
C"] ;Grabación de la temperatura
SEROUT2 Data_temp,84,["CLF Env_Temperatura.xls", $0D]
    PAUSE 20
    RETURN

Grab_Decena:SEROUT2 Data_Temp,84,[$0D]
    PAUSE 100
    SEROUT2 Data_Temp,84,["IPA", $0D]
    PAUSE 100
    SEROUT2 Data_Temp,84,["OPW Tempera.xls", $0D]
    PAUSE 100
    SEROUT2 Data_Temp,84,["WRF 17", $0D]
PAUSE 20
    SEROUT2 Data_Temp,84,["Muestra", " = ", Dec2 Temperatura, "", DEC Decimal, "
C", $0D] ;Grabación de la temperatura
PAUSE 20

    SEROUT2 Data_temp,84,["CLF Tempera.xls", $0D]
    PAUSE 20
    RETURN


Grab_Centena:SEROUT2 Data_Temp,84,["OPW Env_Temperatura.xls", $0D]
    PAUSE 20
    SEROUT2 Data_Temp,84,["WRF 18", $0D]
PAUSE 20
    SEROUT2 Data_Temp,84,["Muestra", " = ", Dec3 Temperatura, "", DEC Decimal, "
C"] ;Grabación de la temperatura
SEROUT2 Data_temp,84,["CLF Env_Temperatura.xls", $0D]
    PAUSE 20
    RETURN

END

```

5	REFERENCIAS
http://www.robotshop.com/content/PDF/memorystickdataloggerv1.1-27937.pdf	

5.4 Comunicación serial RS485 (SN75176)

	UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL
Formato para Prácticas de Laboratorio.	

Carrera:	Nombre de la Asignatura	Profesor:
Ingeniería en Electrónica con Mención en Sistemas Industriales	Digitales	Ing. Luis Córdova

Práctica No.	Nombre de la Práctica	Duración (Horas por semana)
3	Comunicación serial RS485 (SN75176)	2

1	INTRODUCCIÓN
En esta práctica el alumno comprenderá la comunicación por medio de la norma RS485. Diferencias entre RS232 y RS485. Además usará el Memory Stick datalogger, aplicando el lenguaje de alto nivel PICBASIC.	

2	OBJETIVOS (COMPETENCIAS)
2.1 Mostrar al estudiante las diferencias entre la comunicación serial RS232 y RS485.	
2.2 Mostrar las características operativas de las tarjetas diseñadas para realizar prácticas de comunicación serial.	
2.3 Comprobar experimentalmente la longitud de enlace del CI SN75176, capaz de usar la norma RS485.	
2.4 Reconocer y comprender el funcionamiento de un Memory Stick Datalogger.	

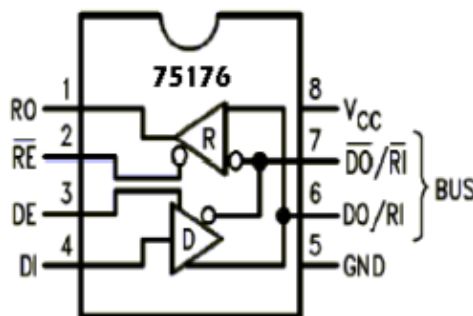
3	FUNDAMENTO
<u>Norma RS485:</u> Cuando se necesita transmitir a largas distancias o con más altas velocidades que RS232, se utiliza RS485; esta interfaz tiene muchas ventajas con respecto a RS232, entre las cuales tenemos: ✓ Bajo costo: Los circuitos integrados para transmitir y recibir requieren una	

fuentes de +5v para poder generar una diferencia mínima de 1.5v entre las salidas diferenciales.

- ✓ Capacidad de interconexión: RS485 es una interfaz multi-enlace con la capacidad de poder tener múltiples transmisores y receptores. Con una alta impedancia receptora, los enlaces con RS485 pueden llegar a tener hasta 32 nodos. Con un simple par de cables.
- ✓ Longitud de enlace: en un enlace RS485 puede tener hasta 4000 pies de longitud, comparado con RS232 que tiene unos límites típicos de 50 a 100 pies.

CI SN75176:

Este dispositivo se encarga de hacer la conversión entre los niveles TTL del microcontrolador y las señales del tipo diferencial que se utilizan el bus RS485.



En los terminales VCC y GND se encuentra la alimentación del circuito, que en este caso es de + 5 V.

La terminal RO y DI recibe un nivel lógico TTL, si sólo si la línea RE se habilita y como se puede observar es con un "0" lógico.

Las terminales DO y \overline{DO} reciben también el nombre de A y B y son sobre estas líneas las que forman el Bus de Transmisión y Recepción.

Como se puede observar cada chip consta de un transmisor y un receptor, si los terminales RE (Pin 2) y DE (Pin 3) se unen entre sí con un solo bit se puede controlar el flujo de la información.

4	PROCEDIMIENTO (DESCRIPCIÓN)	
4.1	EQUIPO NECESARIO	MATERIAL DE APOYO
	<ul style="list-style-type: none"> ✓ 20 metros de cable para probar distancia de envío y recepción. Módulos de práctica en la cual consta de: <ul style="list-style-type: none"> ✓ 1 Microcontrolador PIC16F877A ✓ 1 LCD 16X4 ✓ 1 Conector DB9 ✓ 1 Memory Stick Datalogger ✓ 1 MAX232 ✓ 6 Entradas (Dipswitch) ✓ 6 Salidas LED ✓ LM35 (Sensor de Temperatura) 	Proyector Pizarra Líquida
4.2	DESARROLLO DE LA PRÁCTICA	
<p>Actualmente muy pocos ordenadores cuentan con puerto serial RS232, pero para poder entender este tipo de protocolo usaremos un convertidor RS232 a USB.</p> <p>4.2.1 Transmitir una secuencia de caracteres desde el PC hasta una de las tarjetas de práctica, y luego esta información enviarla a la segunda tarjeta de práctica por medio de RS485.</p> <p><u>Programación para Tarjeta de Práctica # 1 (TRANSMISOR)</u></p> <p>;Definición del Ucontrolador @DEVICEXT_OSC ;Oscilador de Cristal de Cuarzo define osc 4 ;Cristal de cuarzo de 4Mhz</p> <p>;Pines de comunicación del LCD define LCD_DREG PORTD define LCD_DBIT 4 define LCD_RSREG PORTD define LCD_RSBIT 1 define LCD_EREG PORTD define LCD_EBIT 0 DEFINE LCD_BITS 4 DEFINE LCD_LINES 4 DEFINE LCD_COMMANDUS 2000 DEFINE LCD_DATAUS 50</p>		

;Entradas/Salidas de Comunicación Serial TTL

```
Rx_232 var portb.0  
Tx_232 var portc.1
```

;Entradas/Salidas de Comunicación Serial Max485

```
Rx_485 var portb.2  
Tx_485 var portb.3  
Control_1 var portb.4
```

;Variables Generales

```
Dato_2 var byte
```

;Declaración de puertos

```
trisa = %00000111  
trisb = %00100100  
trisc = %00000000  
trisd = %00000000
```

```
LCDOUT $FE,1  
LCDOUT $FE,$80," Practica 4.2.1"  
LCDOUT $FE,$0C0," Transmisor"
```

```
main:
```

```
Lee_Comp: SERIN2 Rx_232,84,[Dato_2] ;Lee de la computadora  
pause 200
```

```
Env_485: high control_1  
SEROUT2 Tx_485,84,[Dato_2]  
Low control_1
```

```
LCDOUT $FE,$90,"Transmision = ", Dato_2  
pause 500  
gotomain
```

```
END
```

Programación para Tarjeta de Práctica # 2 (RECEPTOR)**;Definición del Ucontrolador**

```
@DEVICEXT_OSC ;Oscilador de Cristal de Cuarzo  
define osc 4 ;Cristal de cuarzo de 4Mhz
```

;Pines de comunicación del LCD

```
define LCD_DREG PORTD  
define LCD_DBIT 4  
define LCD_RSREG PORTD  
define LCD_RSBIT 1  
define LCD_EREG PORTD  
define LCD_EBIT 0  
DEFINE LCD_BITS 4  
DEFINE LCD_LINES 4  
DEFINE LCD_COMMANDUS 2000  
DEFINE LCD_DATAUS 50
```

;Entradas/Salidas de Comunicación Serial Max232

Rx_232 var portb.0
Tx_232 var portb.1

;Entradas/Salidas de Comunicación Serial Max485

Rx_485 var portb.2
Tx_485 var portb.3
Control_1 var portb.4

;Variables Generales

Dato_1 var Byte

;Declaración de puertos

trisb = \$00000100
trisd = %00000000

LCDOUT \$FE,1

LCDOUT \$FE,\$80," Practica 4.2.1"

LCDOUT \$FE,\$0C0," Receptor"

main:

Low Control_1
SERIN2 Rx_485,84,[Dato_1]
LCDOUT \$FE,\$90,"Recepcion = ", Dato_1

goto main

END

4.2.2 Con agentes externos se variará la temperatura alrededor del LM35; se tomarán valores de temperatura cada 10 segundos y estos valores serán almacenados en el memory stick datalogger (Comunicación Serial); éstos valores serán enviados a la otra tarjeta de práctica usando el protocolo RS485.

Programación para Tarjeta de Práctica # 2 (TRANSMISOR)

;definición del Ucontrolador

@DEVICEXT_OSC ;Oscilador de Cristal de Cuarzo
define osc 4 ;Cristal de cuarzo de 4Mhz

;Pines de comunicación del LCD

define LCD_DREG PORTD
define LCD_DBIT 4
define LCD_RSREG PORTD
define LCD_RSBIT 1
define LCD_EREG PORTD
define LCD_EBIT 0
DEFINE LCD_BITS 4

```

DEFINE LCD_LINES 4
DEFINE LCD_COMMANDUS 2000
DEFINE LCD_DATAUS 50

;Entradas/Salidas de Comunicación Serial Max485
Rx_485 var portb.2
Tx_485 var portb.3
Control_1 var portb.4

;Entradas/Salidas de Comunicación Serial Max232
Rx_232 var portb.0
Tx_232 var portb.1

;Variables para Manejo de Datalogger
Cant_Datos var byte
Tx_Data var Portd.2

;Variable de Temperatura
Temp_Byte var byte
Temperatura var word
Decimal var byte

;Variables Generales
tiempo var byte
Num_Muestra var byte

;Declaración de puertos
trisa = %00000111
trisb = %00000001
trisc = %00000000
trisd = %00000000
adcon1 = %00001110

;Puerto del LCD
;Configuración de Pines analógicos

temp_Byte = 0
Temperatura = 0
Num_Muestra = 0

pause 1000

SEROUT2 tx_data,84,[$0D]; EnceroDatallogger
LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.2"
LCDOUT $FE,$0C0," Transmisor"

main:

Auto: ADCIN 0, Temp_Byte ; LEO Temperatura
      Temperatura = (19*Temp_Byte)/10
      Decimal = Temperatura dig 0
Num_Muestra = Num_Muestra + 1
gotoCal_Temp

```

;Muestra Temperatura en LCD

```
Cal_Temp: if temperatura <= 9 then
```

```
gotoUnidad
```

```
else
```

```
  If (temperatura>= 10 and Temperatura <=99)then
```

```
gotodecenas
```

```
else
```

```
  If (temperatura>= 100)then
```

```
gotoCentenas
```

```
goto main
```

```
endif
```

```
endif
```

```
endif
```

```
Unidad: LCDOUT $FE,$0D4,"Temp. = ",dec1 Temperatura," ",dec decimal," C "
```

```
gosub env_unidad_485
```

```
gosubGrab_Unidad
```

```
for tiempo = 1 to 10
```

```
  pause 1000
```

```
next
```

```
goto main
```

```
Decenas: LCDOUT $FE,$0D4,"Temp. = ",dec2 Temperatura," ",dec decimal," C "
```

```
gosub env_decenas_485
```

```
gosubGrab_Decena
```

```
for tiempo = 1 to 10
```

```
  pause 1000
```

```
next
```

```
goto main
```

```
Centenas: LCDOUT $FE,$0D4,"Temp. = ",dec3 Temperatura," ",dec decimal," C "
```

```
gosub env_centenas_485
```

```
gosubGrab_Centena
```

```
for tiempo = 1 to 10
```

```
  pause 1000
```

```
next
```

```
goto main
```

;Envío Temperatura por RS485

```
Env_Unidad_485: high control_1
```

```
pause 100
```

```
  serout2 TX_485,84,[Temperatura,decimal]
```

```
  low control_1
```

```
  return
```

```
Env_Decenas_485: high control_1
```

```
  pause 100
```

```
  serout2 TX_485,84,[Temperatura,decimal]
```

```

        low control_1
        return

Env_Centenas_485: high control_1
        pause 100
        serout2 TX_485,84,[Temperatura,decimal]
        low control_1

return

;Grabación de Temperatura de Datalogger

Grab_Unidad:SEROUT2 tx_data,84,[$0D]
        PAUSE 100
        SEROUT2 tx_data,84,["IPA", $0D]
        PAUSE 100
        SEROUT2 tx_data,84,["OPW Tempera.xls",$0D]
        PAUSE 100
        SEROUT2 tx_data,84,["WRF 16",$0D]
PAUSE 20
        SEROUT2 tx_data,84,["Muestra", " = ",Dec1 Temperatura, " ",DEC Decimal, " C"]
;Grabación de la temperatura
PAUSE 20
        SEROUT2 tx_data,84,["CLF Tempera.xls",$0D]
        PAUSE 20
        RETURN

Grab_Decena:SEROUT2 tx_data,84,[$0D]
        PAUSE 100
        SEROUT2 tx_data,84,["IPA", $0D]
        PAUSE 100
        SEROUT2 tx_data,84,["OPW Tempera.xls",$0D]
        PAUSE 100
        SEROUT2 tx_data,84,["WRF 17",$0D]
PAUSE 20
        SEROUT2 tx_data,84,["Muestra", " = ",Dec2 Temperatura, " ",DEC Decimal, "
C", $0D] ;Grabacion de la temperatura
PAUSE 20
        SEROUT2 tx_data,84,["CLF Tempera.xls",$0D]
        PAUSE 20
        RETURN

Grab_Centena:SEROUT2 tx_data,84,[$0D]
        PAUSE 100
        SEROUT2 tx_data,84,["IPA", $0D]
        PAUSE 100
        SEROUT2 tx_data,84,["OPW Tempera.xls",$0D]
        PAUSE 100
        SEROUT2 tx_data,84,["WRF 18",$0D]
PAUSE 20
        SEROUT2 tx_data,84,["Muestra", " = ",Dec3 Temperatura, " ",DEC Decimal, " C"]

;Grabación de la temperatura
PAUSE 20
        SEROUT2 tx_data,84,["CLF Tempera.xls",$0D]
PAUSE 20
        RETURN

END

```

Programación para Tarjeta de Práctica # 1 (RECEPTOR)

;definición del Ucontrolador

@DEVICEXT_OSC ;Oscilador de Cristal de Cuarzo

define osc 4 ;Cristal de cuarzo de 4Mhz

;Pines de comunicacion del LCD

define LCD_DREG PORTD

define LCD_DBIT 4

define LCD_RSREG PORTD

define LCD_RSBIT 1

define LCD_EREG PORTD

define LCD_EBIT 0

DEFINE LCD_BITS 4

DEFINE LCD_LINES 4

DEFINE LCD_COMMANDUS 2000

DEFINE LCD_DATAUS 50

;Entradas/Salidas de Comunicación Serial Max232

Rx_232 var portb.0

Tx_232 var portb.1

;Entradas/Salidas de Comunicación Serial Max485

Rx_485 var portb.2

Tx_485 var portb.3

Control_1 var portb.4

;Variables de Lectura de Temperatura

Temperatura var byte

Decimal var byte

;Declaración de puertos

trisa = \$00000100

trisd = %00000000

LCDOUT \$FE,1

LCDOUT \$FE,\$80," Practica 4.2.2"

LCDOUT \$FE,\$0C0," Receptor"

main:

Low Control_1

SERIN2 Rx_485,84,[temperatura,decimal]

LCDOUT \$FE,\$0D4," Recepcion = ",dec2 temperatura," ",Dec decimal

serout2 TX_232,84,["Temp. = ",dec Temperatura," ",dec decimal," C"]

pause 1000


goto main

END

5 REFERENCIAS

<http://www.i-micro.com/pdf/articulos/rs-485.pdf>

5.5 Uso del Wiznet –WIZ110SR

	UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL
Formato para Prácticas de Laboratorio.	

Carrera:	Nombre de la Asignatura	Profesor:
Ingeniería en Electrónica con Mención en Sistemas Industriales	Digitales	Ing. Luis Córdova

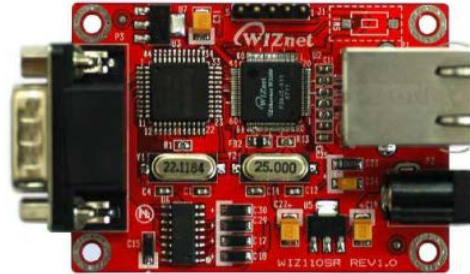
Práctica No.	Nombre de la Práctica	Duración (Horas por semana)
4	Uso del Wiznet –WIZ110SR	2

1	INTRODUCCIÓN
<p>En esta práctica el alumno conocerá un nuevo dispositivo electrónico capaz de enviar y recibir información dentro de una red, aplicando el lenguaje de alto nivel PICBASIC.</p>	

2	OBJETIVOS (COMPETENCIAS)
<p>2.1 Desarrollar las características operativas de las tarjetas diseñadas para realizar prácticas de comunicación.</p> <p>2.2 Reconocer y comprender el funcionamiento de un módulo convertidor de señales RS232 a TCP/IP (módulo Wiznet).</p>	

3	FUNDAMENTO
<p><u>Wiznet:</u></p> <p>El módulo Wiz110SR basado en el chip W5100 funciona como un Gateway que convierte desde el protocolo RS-232 al protocolo TCP/IP y viceversa.</p> <p>Permite entre otras cosas dar conectividad Ethernet a los equipos que no lo poseen, pero sin lugar a dudas el mayor uso que se le da es por medio de la creación de puertos COM virtuales.</p>	

En la figura se observa el aspecto físico del Wiznet para poder tener una mejor referencia.



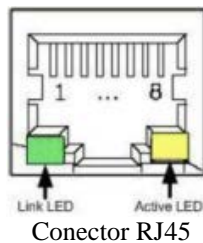
Entre las características principales de WizNet se tienen:

- ✓ Módulo todo incluido para RS-232 y Ethernet.
- ✓ Simple y rápida implementación de la red.
- ✓ Disponibilidad de firmware para varios dispositivos seriales.
- ✓ Alta estabilidad y confiabilidad usando el chip WizNet W5100, con un poderoso stack TCP/IP totalmente diseñado por hardware.
- ✓ Fácil y poderoso programa de configuración por medio del puerto serial.
- ✓ Interfaz Ethernet 10/100Mbps, Interfaz Serial Max 230Kbps.

Especificaciones técnicas

- MCU 8051
- FLASH 62 KB (MCU interna)
- SRAM 16 KB (MCU interna)
- EEPROM 2 KB (MCU interna)

El módulo tiene dos interfaces de comunicación con el exterior, uno de ellos es el denominado RJ45, como indica la figura.



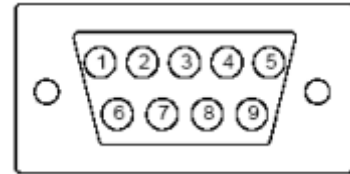
Pin	Signal
1	TX+
2	TX-
3	RX+
6	RX-

Pines de conexión para RJ45

Otro interfaz se basa físicamente en el conector DB9, el cual tiene las siguientes características.

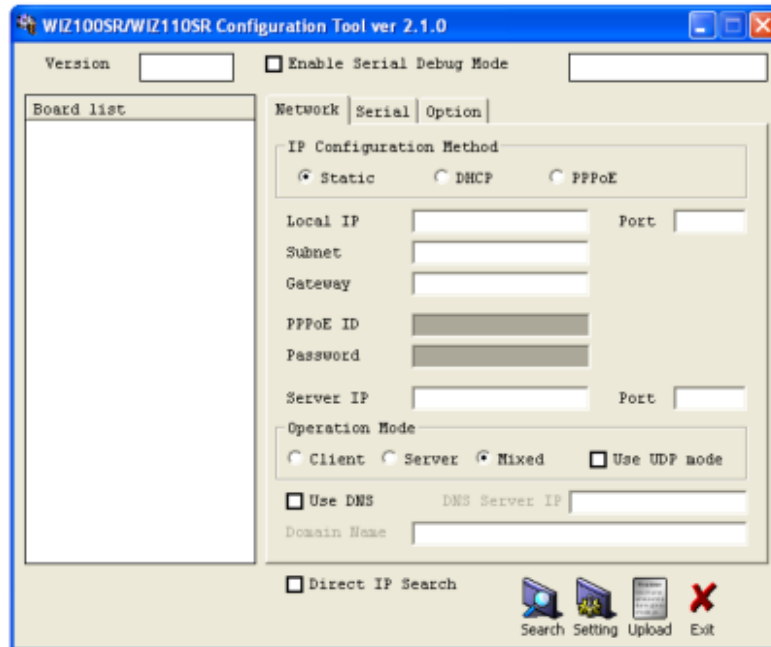
Pin Number	Signal	Description
1	NC	Not Connected
2	RxD	Receive Data
3	TxD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	NC	Not Connected

Descripción de pines del puerto serial



Conector RS232 del Wiznet

Para poder testear si el módulo Wiznet funciona apropiadamente, viene con un programa adicional que puede ser usado para hacer ping, y con el cual se lo configura para diferentes propósitos.



4	PROCEDIMIENTO (DESCRIPCIÓN)	
4.1	EQUIPO NECESARIO	MATERIAL DE APOYO
	Módulos de práctica en la cual consta de: 1 Microcontrolador PIC16F877A 1 LCD 16 X 4 1 Conector DB9 1 Memory Stick Datalogger 1 MAX232	Proyector Pizarra Líquida

	6 Entradas (Dipswitch) 6 Salidas LED LM35 (Sensor de Temperatura)	
4.2	DESARROLLO DE LA PRÁCTICA	
<p>4.2.1 Desarrollar una lógica de programación capaz de transmitir la variación de temperatura del LM35 (con agentes externos se variará la temperatura alrededor del LM35); cada 10 segundos, éstos valores serán enviados por medio de un módulo WIZNET a un router, de tal manera que podemos leer estos valores a tiempo real dentro de una red.</p> <pre> ;definición del Ucontrolador @DEVICEXT_OSC ;Oscilador de Cristal de Cuarzo define osc 4 ;Cristal de cuarzo de 4Mhz ;Pines de comunicación del LCD define LCD_DREG PORTD define LCD_DBIT 4 define LCD_RSREG PORTD define LCD_RSBIT 1 define LCD_EREG PORTD define LCD_EBIT 0 DEFINE LCD_BITS 4 DEFINE LCD_LINES 4 DEFINE LCD_COMMANDUS 2000 DEFINE LCD_DATAUS 50 ;Entradas/Salidas de Comunicación Ethernet Rx_ETH var portb.6 Tx_ETH var portb.7 ;Variable de Temperatura Temp_Byte var byte Temperatura var word Decimal var byte ;Declaración de puertos trisa = %00000111 trisb = \$00000001 trisd = %00000000 ;Puerto del LCD adcon1 = %00001110 ;Configuracion de Pines analogicos temp_Byte = 0 Temperatura = 0 LCDOUT \$FE,1 LCDOUT \$FE,\$80," Practica 4.2.2" </pre>		

```


main:
Auto:   ADCIN 0, Temp_Byte      ; LEO Temperatura
        Temperatura = (19*Temp_Byte)/10
        Decimal = Temperatura dig 0
gotoCal_Temp
;Muestra Temperatura en LCD
Cal_Temp:  if temperatura<= 9 then
gotoUnidad
        else
        If (temperatura>= 10 and Temperatura <=99)then
gotodecenas
        else
        If (temperatura>= 100)then
gotoCentenas
goto main
endif
endif
endif
Unidad:   LCDOUT $FE,$0C0,"Temp. = ",dec1 Temperatura," ",dec decimal," C "
gosubenv_unidad_ETH
        PAUSE 1000
goto main
Decenas:  LCDOUT $FE,$0C0,"Temp. = ",dec2 Temperatura," ",dec decimal," C "
gosubenv_decenas_ETH
        PAUSE 1000
goto main
Centenas: LCDOUT $FE,$0C0,"Temp. = ",dec3 Temperatura," ",dec decimal," C "
gosubenv_centenas_ETH
        PAUSE 1000
goto main

;Envío Temperatura a red Ethernet
Env_Unidad_ETH:  serout2 tx_ETH,84,["Temp. = ",dec1 Temperatura," ",dec decimal,"
C"]
return
Env_Decenas_eth:  serout2 TX_ETH,84,["Temp. = ",dec2 Temperatura," ",dec decimal,"
C"]
return
Env_Centenas_eth: serout2 TX_ETH,84,["Temp. = ",dec3 Temperatura," ",dec decimal,"
C"]
return
END

```

5	REFERENCIAS
http://www.olimex.cl/pdf/WizNet/Guia%20MCI%20%20Conversor%20Serial%20Ethernet%20WizNet%20v1.1.pdf	

5.6 Comunicación Inalámbrica. Módulos XBEE

	UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL
Formato para Prácticas de Laboratorio.	

Carrera:	Nombre de la Asignatura	Profesor:
Ingeniería en Electrónica con Mención en Sistemas Industriales	Digitales	Ing. Luis Córdova

Práctica No.	Nombre de la Práctica	Duración (Horas por semana)
5	Comunicación Inalámbrica. Módulos XBEE	2

1	INTRODUCCIÓN
En esta práctica el alumno comprenderá la comunicación inalámbrica por medio de los módulos XBEE, aplicando el lenguaje de alto nivel PICBASIC.	

2	OBJETIVOS (COMPETENCIAS)
<p>2.1 Mostrar al estudiante una forma de comunicación novedosa para poder implementarla dentro de sus futuros proyectos.</p> <p>2.2 Reconocer y comprender el funcionamiento de un Módulo XBEE.</p> <p>2.2 Mostrar las características operativas de las tarjetas diseñadas para realizar prácticas de comunicación.</p>	

3	FUNDAMENTO
<p><u>Módulo XBEE.</u></p> <p>Una buena forma de agregar conectividad inalámbrica a tu proyecto es utilizando los módulos Xbee de MaxStream. Los módulos Xbee proveen 2 formas amigables de comunicación: Transmisión serial transparente (modo AT) y el modo API que provee muchas ventajas. Los módulos Xbee pueden ser configurados desde el PC utilizando el programa X-CTU o bien desde tu microcontrolador. Los Xbee pueden comunicarse en arquitecturas punto a punto, punto a multi punto o en una red mesh.</p>	

La elección del módulo XBee correcto pasa por escoger el tipo de antena (chip, alambre o conector SMA) y la potencia de transmisión (2mW para 300 pies o 60mW para hasta 1 milla).

Debido a que los módulos Xbee tienen una separación de pines de 2mm, se vuelve necesario utilizar tarjetas adaptadoras para corregir la separación de los pines a 5 mm. Si vas a comunicar un módulo Xbee con un PC recomendamos utilizar un adaptador USB que te va a permitir configurar el módulo fácilmente y probar la configuración antes de utilizar el módulo en una red punto a punto.

Los módulos Xbee son económicos, poderosos y fáciles de utilizar. Algunas de sus principales características son:

- ✓ Buen Alcance: hasta 300ft (100 mts) en línea vista para los módulos Xbee y hasta 1 milla (1.6 Km) para los módulos Xbee Pro.
- ✓ 9 entradas/salidas con entradas analógicas y digitales.
- ✓ Bajo consumo <50mA cuando están en funcionamiento y <10uA cuando están en modo sleep.
- ✓ Interfaz serial.
- ✓ 65,000 direcciones para cada uno de los 16 canales disponibles. Se pueden tener muchos de estos dispositivos en una misma red.
- ✓ Fáciles de integrar

Existen 2 series de estos módulos. La serie 1 está basada en el chipset Freescale y está pensado para ser utilizado en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están basados en el chipset de EMBER y están diseñados para ser utilizados en aplicaciones que requieren repetidores o una red mesh. Ambos módulos pueden ser utilizados en los modos AT y API. Los módulos de la Serie 1 y la Serie 2 tienen el mismo pin-out, sin embargo, NO son compatibles entre sí ya que utilizan distintos chipset y trabajan con protocolos diferentes.

4	PROCEDIMIENTO (DESCRIPCIÓN)	
4.1	EQUIPO NECESARIO	MATERIAL DE APOYO
	Módulos de práctica en la cual consta de: 1 Microcontrolador PIC16F877A 1 LCD 16 X 4 1 Conector DB9 1 Memory Stick Datalogger 1 MAX232 6 Entradas (Dipswitch) 6 Salidas LED LM35 (Sensor de Temperatura)	Proyector Pizarra Líquida
4.2	DESARROLLO DE LA PRÁCTICA	
	<p>4.2.1 Desarrollar una lógica de programación capaz de transmitir una serie de caracteres desde el PC hasta una de las tarjetas de prácticas; luego de esto, la información deberá pasar por comunicación inalámbrica (módulos XBee) hasta la otra tarjeta de práctica.</p> <p><u>Programación para Tarjeta de Práctica # 1 (TRANSMISOR)</u></p> <p>;definición del Ucontrolador @DEVICEEXT_OSC ;Oscilador de Cristal de Cuarzo define osc 4 ;Cristal de cuarzo de 4Mhz</p> <p>;Pines de comunicación del LCD define LCD_DREG PORTD define LCD_DBIT 4 define LCD_RSREG PORTD define LCD_RSBIT 1 define LCD_EREG PORTD define LCD_EBIT 0 DEFINE LCD_BITS 4 DEFINE LCD_LINES 4 DEFINE LCD_COMMANDUS 2000 DEFINE LCD_DATAUS 50</p> <p>;Entradas/Salidas de Comunicación Serial 232 Rx_232 var portb.0 Tx_232 var portc.1</p> <p>;Entradas/Salidas de Comunicación Xbee Rx_Xbee var portc.0 Tx_Xbee var portc.1</p> <p>;Variables Generales Dato_2 var byte</p>	

;Declaración de puertos

```
trisa = %00000111
trish = %00100100
trisc = %00000000
trisd = %00000000
LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.1"
LCDOUT $FE,$0C0," Transmisor"
```

main:

```
Lee_Comp:      SERIN2 Rx_232,84,[Dato_2]      ;Lee de la computadora
pause 200
```

Env_Xbee:

```
SEROUT2 Tx_Xbee,84,[Dato_2]
```

```
LCDOUT $FE,$D4,"Transmision = ", Dato_2
```

```
pause 500
```

goto main

END

Programación para Tarjeta de Práctica # 2 (RECEPTOR)**;definición del Ucontrolador**

```
@DEVICEXT_OSC      ;Oscilador de Cristal de Cuarzo
define osc 4        ;Cristal de cuarzo de 4Mhz
```

;Pines de comunicación del LCD

```
define      LCD_DREG  PORTD
define      LCD_DBIT  4
define      LCD_RSREG PORTD
define      LCD_RSBIT 1
define      LCD_EREG  PORTD
define      LCD_EBIT  0
DEFINE     LCD_BITS 4
DEFINE     LCD_LINES 4
DEFINE     LCD_COMMANDUS 2000
DEFINE     LCD_DATAUS 50
```

;Entradas/Salidas de Comunicación Xbee

```
Rx_Xbee var portc.0
Tx_Xbee var portc.1
```

;Variables Generales

```
Dato_1 var Byte
```

;Declaración de puertos

```
trish = $00000100
trisd = %00000000

LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.1"
LCDOUT $FE,$0C0," Receptor"
```

```

main:
    SERIN2 Rx_xbee,84,[Dato_1]
    LCDOUT $FE,$D4,"Recepcion = ", Dato_1
goto main

END

```

4.2.2 Desarrollar una lógica de programación capaz de transmitir la variación de temperatura del LM35; desde la tarjeta de prácticas #1 (transmisor) hasta la tarjeta de prácticas #2 (receptor).

Programación para Tarjeta de Práctica # 1 (TRANSMISOR)

;definición del Ucontrolador

```

@DEVICEXT_OSC    ;Oscilador de Cristal de Cuarzo
define osc 4     ;Cristal de cuarzo de 4Mhz

```

;Pines de comunicación del LCD

```

define    LCD_DREG  PORTD
define    LCD_DBIT  4
define    LCD_RSREG PORTD
define    LCD_RSBIT 1
define    LCD_EREG  PORTD
define    LCD_EBIT  0
DEFINE    LCD_BITS 4
DEFINE    LCD_LINES 4
DEFINE    LCD_COMMANDUS 2000
DEFINE    LCD_DATAUS 50

```

;Entradas/Salidas de Comunicación Xbee

```

Rx_Xbee var portc.0
Tx_Xbee var portc.1

```

;Variable de Temperatura

```

Temp_Byte var byte
Temperatura var word
Decimal var byte

```

;Declaración de puertos

```

trisa = %00000111
trisb = $00000001
trisd = %00000000 ;Puerto del LCD
adcon1 = %00001110 ;Configuracion de Pines analogicos

```

```

temp_Byte = 0
Temperatura = 0

```

```

LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.2"
LCDOUT $FE,$0C0," Transmisor"

```

```

main:

```

```

Auto:   ADCIN 0, Temp_Byte      ; LEO Temperatura
        Temperatura = (19*Temp_Byte)/10
        Decimal = Temperatura dig 0
gotoCal_Temp

;Muestra Temperatura en LCD
Cal_Temp:  if temperatura<= 9 then
gotoUnidad
        else
        If (temperatura>= 10 and Temperatura <=99)then
gotodecenas
        else
        If (temperatura>= 100)then
gotoCentenas
goto main
endif
endif
endif

Unidad:   LCDOUT $FE,$D4,"Temp. = ",dec1 Temperatura," ",dec decimal," C "
gosubenv_unidad_Xbee
        PAUSE 1000
gotomain
Decenas:  LCDOUT $FE,$D4,"Temp. = ",dec2 Temperatura," ",dec decimal," C "
gosubenv_decenas_Xbee
PAUSE 1000
goto main
Centenas: LCDOUT $FE,$D4,"Temp. = ",dec3 Temperatura," ",dec decimal," C "
gosubenv_centenas_Xbee
        PAUSE 1000
goto main

;Envío Temperatura por Xbee

Env_Unidad_Xbee: serout2 TX_xbee,84,[Temperatura,decimal]
return

Env_Decenas_Xbee: serout2 TX_xbee,84,[Temperatura,decimal]
        return

Env_Centenas_Xbee: serout2 TX_xbee,84,[Temperatura,decimal]
        return

END

Programación para Tarjeta de Práctica # 2 (RECEPTOR)

;definición del Ucontrolador
@DEVICEXT_OSC      ;Oscilador de Cristal de Cuarzo
define osc 4       ;Cristal de cuarzo de 4Mhz

;Pines de comunicación del LCD
define      LCD_DREG  PORTD
define      LCD_DBIT  4

```

```

define      LCD_RSREG  PORTD
define      LCD_RSBIT  1
define      LCD_EREG   PORTD
define      LCD_EBIT   0
DEFINE     LCD_BITS 4
DEFINE     LCD_LINES 4
DEFINE     LCD_COMMANDUS 2000
DEFINE     LCD_DATAUS 50

;Entradas/Salidas de Comunicación Xbee
Rx_Xbee var portc.0
Tx_Xbee var portc.1

;Variables de Lectura de Temperatura
Temperatura var byte
Decimal var byte

;Declaración de puertos
trisb = $00000001
trisd = %00000000

LCDOUT $FE,1
LCDOUT $FE,$80," Practica 4.2.2"
LCDOUT $FE,$0C0," Receptor"

main:
SERIN2 Rx_xbee,84,[temperatura,decimal]
LCDOUT $FE,$D4," Recepcion = ",dec2 temperatura," ",Dec decimal," C"
pause 1000
goto main

END

```

5	REFERENCIAS
---	-------------

http://www.xbee.cl/	
---	--

CAPÍTULO 6

COSTOS DEL PROYECTO

En este capítulo se detallan los costos de los diferentes elementos que se usaron en el proyecto.

6.1 Costo del Módulo Registro de Datos

	Elementos	Cantidad de elementos en cada módulo	Costo Unitario	Costo Total	
				Un Módulo	Cuatro Módulos
MÓDULO REGISTRO DE DATOS	Resistencia de 100 ohm 1/4 w	17	0,05	0,85	3,4
	Dipswitch (6 bits)	1	1	1	4
	Microcontrolador PIC16F877A	1	10	10	40
	Microcontrolador PIC16F876	1	8	8	32
	RTC DS1307	1	5	5	20
	CI. SN75176	1	3	3	12
	Pantalla Cristal Liquido LCD 20X4	1	20	20	80
	Memory Stick Datalogger	1	80	80	320
	Lector ID12	1	50,53	50,53	202,12
	Pen Drive 2Gb	1	22	22	88
	Socket 40 pines	1	0,8	0,8	3,2
	Socket 28 pines	1	0,64	0,64	2,56
	Socket 8 pines	2	0,25	0,5	2
	Socket para CR2032	1	0,89	0,89	3,56
	Batería CR2032	1	0,8	0,8	3,2
	Cristal de cuarzo 4 Mhz	2	0,5	1	4
	Cristal de cuarzo 12367 KHz	1	0,49	0,49	1,96
	Borneras de 2 Contactos	1	0,35	0,35	1,4
	Borneras de 3 Contactos	3	0,35	1,05	4,2
	Borneras de 4 Contactos	1	0,35	0,35	1,4
	Diodos Rectificadores 1N4007	2	0,15	0,3	1,2
	Relé 5V	1	2	2	8
	Transistor 2N3904	2	0,1	0,2	0,8
	Pulsantes	2	0,15	0,3	1,2
	Diodos LED 3mm (color verde)	3	0,15	0,45	1,8
					210,5

Tabla 15 Precio de elementos del Módulo Registro de Datos

Fuente: Autores

TOTAL \$ 842

6.2 Costo del Módulo Base

	Elementos	Cantidad de elementos en cada módulo	Costo Unitario	Costo Total
Módulo BASE	Resistencia de 100 ohm 1/4 w	3	0,05	0,15
	Resistencia de 1 k ohm 1/4 w	4	0,15	0,6
	Pulsantes	1	0,15	0,15
	Diodos LED 3mm (color verde)	1	8	8
	Módulo WIZNET	1	50,53	50,53
	Microcontrolador PIC16F876	1	15	15
	CI. SN75176	1	1	1
	Socket 40 pines	1	1	1
	Socket 8 pines	1	0,25	0,25
	Conector DB9	1	0,7	0,7
	Cristal de cuarzo 4 Mhz	2	0,5	1
	Borneras de 2 Contactos	3	0,35	1,05

Tabla 16 Precio de elementos del Módulo Base

Fuente: Autores

TOTAL \$ 69,20

6.3 Costo del Módulo RFID

	Elementos	Cantidad de elementos en cada módulo	Costo Unitario	Costo Total
Módulo RFID	Resistencia de 1 k ohm 1/4 w	3	0,05	0,15
	Pulsantes	4	0,15	0,6
	Diodos LED 3mm (color verde)	1	0,15	0,15
	PIC16F877A	1	8	8
	Lector ID12	1	50,53	50,53
	Convertor RS232 TO USB	1	15	15
	Socket 40 pines	1	1	1
	Cristal de cuarzo 4 Mhz	1	1	1
				76,43

Tabla 17 Precio de elementos del Módulo RFID

Fuente: Autores

TOTAL \$ 76,43

6.4 Costo de accesorios y otros

	Elementos	Cantidad de elementos en cada módulo	Costo Unitario	Costo Total
Accesorios & Otros	Switch o router	1	40	40
	Chapas magnéticas	4	80	320
	TAG RFID CARD	10	3	30
	Cajas de acrílico para MÓDULO REGISTRO DE DATOS	4	15	60
	Cajas de acrílico para módulo base	1	10	10
	Cajas de acrílico para módulo RFID	1	15	15
	Fuente de alimentación (Modulo Registro de Datos)	1	75	75
	Fuente de alimentación (Modulo Base)	1	10	10
	Circuito impreso para módulo base	1	50	50
	Circuito impreso para MÓDULO REGISTRO DE DATOS	4	100	400
	Circuito impreso para módulo RFID	1	30	30
	Protoboard (para hacer pruebas de laboratorio)	2	20	40
	Pulsadores internos en cada puerta	4	0,5	2
	Fabricación de Maquetas de puertas a escala	4	150	600
	Cable Apantallado	25	0,5	12,5
	Elementos varios para hacer pruebas en el desarrollo del proyecto	6	20	120
				1814,5

Tabla 18 Precio de accesorios y otros

Fuente: Autores

TOTAL \$ 1814,5

6.5 Costo de Módulos de Prácticas

	Elementos	Cantidad de elementos en cada módulo	Costo Unitario	Costo Total
Módulos de Practicas	Memory Stick Datalogger	2	80	160
	Pantalla Cristal Liquido LCD 20X4	2	20	40
	CI. MAX232	2	2,5	5
	Microcontrolador PIC16F877A	2	10	20
	CI. SN75176	2	3	6
	Router LINK	1	45	45
	Fuente de Voltaje	1	75	75
	Regleta	1	5	5
	Wiznet	2	60	120
	XBEE	2	45	90
	Adaptadores Xbee	2	28	56
	Elaboración Placa Electrónica (Practicas)	2	60	120
	Conectores DB9	6	1	6
	Pen Drive de 4Gb	2	12	24
	Acrílicos transparente	2	4	8
	Componentes varios	1	50	50
	Herramientas varias (cautín, estaño, soldadura)	1	30	30
				\$ 860

Tabla 19 Precio de Módulos de Practicas

Fuente: Autores

TOTAL \$ 860

6.6 Costo total del proyecto

DENOMINACIÓN	Cant.	Costo Total
MÓDULO REGISTRO DE DATOS	4 u.	842,00
MÓDULO BASE	1u.	69,20
MÓDULO RFID	1u.	76,43
ACCESORIO & OTROS		1814,5
MÓDULO DE PRACTICAS	2 u.	860,00
Costo total del proyecto		3662,13

Tabla 20 Costo Total del proyecto

Fuente: Autores

CONCLUSIONES:

El proyecto ha logrado cumplir los objetivos propuestos y respetando las especificaciones planteadas, he inclusive se incluyó una importante forma de abordar la enseñanza para aplicarlo en el entorno educativo por medio de módulos de prácticas para un mejor estudio de los elementos usados en el proyecto principal, siendo estos una ayuda fundamental para futuros proyectos.

Además cabe mencionar que la tecnología RFID es una alternativa moderna, robusta y flexible que en nuestro país no es muy aplicada, pero su uso presenta una amplia variedad de aplicaciones de una manera sencilla, desde la administración de personas (control de acceso), seguridad de bienes hasta manejo de inventarios.

Se ha desarrollado módulos capaces de obtener la información y organizarla, todo esto usando una programación estructurada lo que permite una fácil extensibilidad de la aplicación. Las herramientas utilizadas en nuestro proyecto fueron los microcontroladores de la empresa Microchip ya que tienen características de gama alta (capacidad muy grande para el control de procesos, espacio de memoria suficiente, además de todas sus características periféricas); los cuales se han relacionado de manera exitosa con el software, desarrollado en la plataforma de Visual Basic Studio, llegando al punto de obtener datos en tiempo real.

Junto con el proyecto se ha desarrollado un protocolo (conjunto de reglas para comunicación de datos), capaz de relacionar una variedad de comunicaciones seriales desde el mas simple USART, hasta el mas complejo TCP/IP en base al lenguaje de programación usado.

Esta estructura modular permite que se puedan agregar módulos sin que afecte al resto, por otra el flujo de datos es individual con respecto al módulo principal (módulo base), debido a que existe solo un canal de comunicación, siendo los datos enviados en forma paralela a los módulos, respondiendo el módulo cuya dirección sea la indicada.

Como indicamos en párrafos anteriores se dejan bases para proponer proyectos usando tecnología RFID como por ejemplo, control de inventario, administración de bodegas, seguridad de equipos, administración y monitoreo de información, entre otros, siendo estos temas avanzados en otros países.

Tomando como referencia nuestro proyecto, se puede incluir otros módulos electrónicos, como pantallas gráficas con el objetivo de visualizar información mas detallada, he inclusive que el personal ingrese los datos por medio de una pantalla táctil directamente en los módulos acerca de una diversidad de aplicaciones.

BIBLIOGRAFÍA

➤ **LIBROS:**

- ✓ SARAVIA – CORIA y otros, MPLAB X y TECNICAS DE PROGRAMACION CON LIBRERIAS DE MICROCHIP. Edición, Editorial ROLTA, Argentina, Mayo 2011.
- ✓ SARAVIA – CORIA, Arquitectura y Programación de Microcontroladores PIC. Edición, Editorial ROLTA, Argentina, Julio 2010.
- ✓ REYES, Carlos, Microcontroladores PIC: Programación en BASIC – 2da. Edición, Editorial RISPERSGRAF, Quito-Ecuador, 2006
- ✓ RAMOS Guillermo y otros, Curso practico de Electrónica Industrial y Automatización, Edición. Editorial CEKIT. COLOMBIA, 2002

➤ **PÁGINAS WEB:**

- ✓ MICROCHIP TECHNOLOGY INC, Manual de PIC16F877A, Año 2011, <http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf>
- ✓ MICROCHIP TECHNOLOGY INC, Manual de PIC16F876, Año 2011, <http://ww1.microchip.com/downloads/en/devicedoc/30292c.pdf>
- ✓ INGENIERIA MCL Ltda, Guía Wiznet modelo WIZ110SR, Año 2012, <http://ww.olimex.cl>
- ✓ ID INNOVATIONS, ID Series Datasheet, Marzo 1 del 2005, http://www.id-innovations.com/httpdocs/EM%20moudule%20SERIES%202007-10-9_wfinal%20v22.pdf

- ✓ PARALLAX, Memory Stick Datalogger (#27937), Año 2009,
<http://www.parallax.com/Portals/0/Downloads/docs/prod/comm/MemoryStickDataloggerV1.1.pdf>