REPÚBLICA DEL ECUADOR
Escuela Politécnica Nacional
"E SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.

- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.

- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DE SENSORES INMÓTICOS PARA LAS INSTALACIONES DE UPDATECOM Cia. Ltda.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y CONTROL

JÉSSICA GABRIELA BELTRÁN ULLAURI
gabybel20@yahoo.com

MIGUEL ALEJANDRO SEMPERTEGUI SOSA
miguelsemp@yahoo.com

DIRECTOR: ING. JUAN FERNANDO YÉPEZ RODRÍGUEZ
juan.yepez@updatecom.com.ec

CO-DIRECTOR: ING. ROBIN GERARDO ÁLVAREZ RUEDA, PhD
robin.alvarez@epn.edu.ec

Quito, Noviembre 2011

DECLARACIÓN

Nosotros, Jéssica Gabriela Beltrán Ullauri y Miguel Alejandro Sempertegui Sosa, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada 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 nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____	_____
Jéssica Gabriela	Miguel Alejandro
Beltrán Ullauri	Sempertegui Sosa

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Jéssica Gabriela Beltrán Ullauri y Miguel Alejandro Sempertegui Sosa, bajo nuestra supervisión.

Ing. Juan Yépez	**Ing. Robin Álvarez PhD.**
DIRECTOR DEL PROYECTO	**CO-DIRECTOR DEL PROYECTO**

AGRADECIMIENTO

Quiero agradecer a Dios, a mi familia, al Ing. Robin Álvarez, al Ing. Juan Yépez, a la Dra. Carlota y a todos los integrantes de UPDATECOM, quienes han sido fuerza inspiradora y parte decisiva para el desarrollo de este proyecto.

Gaby

AGRADECIMIENTO

Agradezco a mis padres y a mi hermana quienes me han apoyado inmensamente a lo largo de mi vida estudiantil y han sido un soporte en momentos de adversidad; al Ing. Juan Yépez y al Ing. Robin Álvarez quienes han dedicado tiempo y esfuerzo en el desarrollo de este proyecto.

Miguel.

DEDICATORIA

A mi mamita Lidia, por siempre apoyarme y darme ánimos para cumplir cada uno de los proyectos que emprendo y por el cariño que me expresa cada día.

A mis papás Edison y Roberto, quienes siempre se han preocupado por mi bienestar.

A mis hermanos Carolina, Karen y Esteban, quienes siempre han confiado en mí y no han dudado en que puedo lograr mis objetivos.

A mi Galecito, quien siempre ha sido un gran ejemplo para mí y supo educarme con amor y sabiduría de un padre.

A mi Hanita, por darme todos los días mucha alegría.

A Dios, por llenar mi corazón de amor y darme fuerzas para salir adelante y luchar por mis sueños.

Gaby

DEDICATORIA

A mis padres quienes han estado siempre a mi lado sin tomar en cuenta problemas ni horarios.

A mi ñaña que, aun estando lejos, ha sido una fuente de inspiración, una guía y gran amiga.

A mis abuelitos quienes constantemente se preocupan de mi bienestar y me dan palabras de ánimo.

Miguel

INDICE DE CONTENIDOS

DECLARACIÓN i
CERTIFICACIÓN.....ii
AGRADECIMIENTO.....iii
DEDICATORIA.....v
CONTENIDO.....vii
ÍNDICE DE FIGURAS.....xi
ÍNDICE DE TABLAS.....xvii
RESUMEN.....xviii
PRESENTACIÓN.....xix

CONTENIDO

CAPÍTULO 1: MARCO TEÓRICO

1.2 PLANTEAMIENTO DEL PROBLEMA 1
1.3 SOLUCIÓN AL PROBLEMA 2
1.4 DÓMOTICA E INMÓTICA 4
1.4.1 APORTES DE LA DOMÓTICA E INMÓTICA 5
1.5 HERRAMIENTAS DE SOLUCIÓN 6
1.5.1 ESTÁNDARES DE COMUNICACIÓN 6
1.5.1.1 EIA RS-232 6
1.5.1.1.1 Funcionamiento 6
1.5.1.1.2 Niveles de voltaje 7
1.5.1.2 EIA RS-485 8
1.5.1.2.1 Topología 8
1.5.1.2.2 Funcionamiento 9
1.5.1.2.3 Niveles de voltaje 10
1.5.2 TECNOLOGÍAS DE COMUNICACIÓN 10
1.5.2.1 Bluetooth 10
1.5.2.2 Servicio SMS 11
1.5.2.2.1 Acceso al Servicio SMS 12
1.5.2.3 Correo electrónico 13
1.5.2.3.1 Tipos de Correo Electrónico 14
1.5.3 HARDWARE 15
1.5.3.1 Sensores de corriente 15
1.5.3.1.1 Pinza amperométrica de núcleo de hierro 16
1.5.3.1.2 Transductor de núcleo de aire 17

1.5.3.2 Sensores de humo 18
1.5.3.2.1 Detector óptico 18
1.5.3.2.2 Detector iónico 20
1.5.3.3 Sensores de temperatura 21
1.5.3.3.1 Termopares 21
1.5.3.3.2 Sensores de circuito integrado (CI) 22
1.5.3.4 Detectores de presencia de agua 23
1.5.3.5 Microcontroladores AVR ATmega 23
1.5.3.5.1 Microcontrolador ATmega 164P 23
1.5.3.5.2 Microcontrolador ATmega 48. 25
1.5.4 SOTWARE 27
1.5.4.1 Visual C# 27
1.5.4.2 XAMPP 27
1.5.4.2.1 Aplicaciones 28
1.5.4.3 Gestor de bases de datos phpMyAdmin 29
1.5.4.4 My SQL 30
1.5.4.4.1 Características Principales 30
1.5.4.4.2 Ventajas 31
1.5.4.4.3 Desventajas 31
1.5.4.5 Comandos AT 31
1.5.4.6 NetBeans IDE 32
1.5.4.7 Java Micro Edition (JME) 33
1.5.4.7.1 Definición de Java 33
1.5.4.7.2 Campo de aplicación de Java ME 34
1.5.4.7.3 Midlet 35
1.5.4.7.4 Interfaz de usuario 35
1.5.4.8 Bascom AVR 36
1.5.4.8.1 Características 37
1.5.4.8.2 Comandos e instrucciones 37

CAPÍTULO 2: DISEÑO DE HARDWARE E IMPLEMENTACIÓN

2.1 ESQUEMA GENERAL 39
2.2 CIRCUITOS INTEGRADOS COMUNES EN LOS NODOS 39
2.2.1 MAX487 / SN75176 39
2.2.1.1 Control 40
2.2.1.2 Reemplazo 42
2.2.1.3 Configuración de la red RS-485 43
2.2.2 REGULADORES 44
2.2.2.1 LM7805 44
2.2.2.2 LM7809 44
2.3 NODO MAESTRO 45
2.3.1 FUENTE DE ALIMENTACIÓN 48
2.3.1.1 Elementos implementados 49

2.3.2 MICROCONTROLADOR 51
2.3.2.1 Selección del microcontrolador 51
2.3.2.2 Compatibilidad 52
2.3.2.3 Conexiones 54
2.3.2.4 Pines para programación 56
2.3.3 COMUNICACIÓN CON LA INTERFAZ ESTACIONARIA (PC) 57
2.3.4 COMUNICACIÓN CON LA RED RS-485 58
2.3.5 ENTRADAS OPTOACOPLADAS 59
2.3.6 SALIDAS DE RELÉ 62
2.3.7 SEÑALIZACIÓN 66
2.3.8 CAPACIDAD DE AMPLIACIÓN 70
2.4 NODOS ESCLAVOS 70
2.4.1 NODO DE PRESENCIA DE AGUA 70
2.4.1.1 Circuito de comparación de detección de agua. 71
2.4.1.1.1 Circuito de filtrado de señal. 72
2.4.1.1.2 Circuito de amplificación de señal. 74
2.4.1.1.3 Circuito de comparación de voltaje con histéresis 76
2.4.1.2 Circuito de Adquisición de Datos y Enlace a la red RS-485 79
2.4.1.2.1 Selección del microcontrolador 81
2.4.1.2.2 Controlador del circuito. 81
2.4.1.2.3 Comunicación con la red RS-485. 82
2.4.1.2.4 Fuente de alimentación. 83
2.4.1.2.5 Señalización. 83
2.4.1.2.6 Pines para programación del microcontrolador. 84
2.4.1.2.7 Adquisición del estado del detector de agua. 84
2.4.1.2.8 Medición de humedad en el ambiente. 85
2.4.2 NODO DE CORRIENTE 86
2.4.2.1 Sensado de corriente 87
2.4.2.2 Acondicionamiento de la señal 89
2.4.2.3 Circuito de Adquisición de Datos y Enlace a la red RS-485 91
2.4.2.3.1 Selección del microcontrolador 93
2.4.2.3.2 Controlador 93
2.4.2.3.3 Fuente de alimentación 94
2.4.2.3.4 Comunicación con la red RS-485 95
2.4.2.3.5 Programación del microcontrolador. 96
2.4.3 NODOS DE TEMPERATURA 96
2.4.3.1 Sensado de temperatura 97
2.4.3.2 Circuito de Adquisición de Datos y Enlace a la red RS-485 100
2.4.3.2.1 Selección del microcontrolador 102
2.4.3.2.2 Controlador 103
2.4.3.2.3 Fuente de alimentación 103
2.4.3.2.4 Comunicación con la red RS-485 104
2.4.3.2.5 Pines para programación del microcontrolador. 104
2.5 NODO DE HUMO 105
2.5.1.1 Sensor implementado 106
2.5.1.2 Acondicionamiento de la señal 108
2.6 DIAGRAMA UNIFICADO DEL SISTEMA 114

2.7 IMPLEMENTACIÓN 115
2.7.1 DEPARTAMENTO TÉCNICO 115
2.7.1.1 Ubicación del nodo de temperatura. 116
2.7.1.2 Ubicación del nodo de presencia de agua. 117
2.7.1.3 Ubicación del nodo de presencia de humo. 119
2.7.2 TALLER 121
2.7.3 UBICACIÓN DEL NODO DE CORRIENTE 125
2.7.4 UBICACIÓN DEL SENSOR MAGNÉTICO 126
2.7.5 ACTIVACIÓN DE CERRADURA ELÉCTRICA. 127

CAPÍTULO 3: DISEÑO DE SORTWARE

3.1 INTRODUCCIÓN 128
3.2 DESARROLLO DE LA APLICACIÓN DE LA INTERFAZ ESTACIONARIA 128
3.2.1 ACCESO A RECURSOS NECESARIOS 130
3.2.2 ADMINISTRACIÓN 136
3.2.2.1 Login 136
3.2.2.2 Parámetros 137
3.2.2.3 Usuarios 139
3.2.2.3.1 Registro de usuarios 140
3.2.2.3.2 Eliminación de usuarios 143
3.2.2.4 Conectividad 144
3.2.2.5 SMS/Mail 145
3.2.2.6 Destinatarios 146
3.2.2.6.1 Registro de destinatarios para envío de mensajes de texto 148
3.2.2.6.2 Eliminar destinatarios para envío de mensajes de texto 150
3.2.2.6.3 Registro de destinatarios para envío de correo electrónico 152
3.2.2.6.4 Eliminar destinatarios para envío de correo electrónico 154
3.2.2.7 Guardar configuraciones 155
3.2.3 ALERTAS 156
3.3 DESARROLLO DE LA APLICACIÓN DE LA INTERFAZ MÓVIL 157
3.3.1 DIAGRAMA DE FLUJO DE LA APLICACIÓN DE LA INTERFAZ MÓVIL 158
3.3.2 PANTALLA DE BIENVENIDA 159
3.3.3 SEGURIDAD DE LA APLICACIÓN 159
3.3.4 CONEXIÓN A LA INTERFAZ ESTACIONARIA 161
3.3.5 MENÚ DE SENSORES 163
3.3.5.1 Opción “Temperatura” 164
3.3.5.2 Opción “Corriente” 166
3.3.5.3 Opción “Inundación” 167
3.3.5.4 Opción “Humo” 168
3.3.5.5 Opción “Magnetico” 169
3.4 SOFTWARE DE CONTROL DE LOS NODOS DE LA RED RS−485 169
3.4.1 NODO MAESTRO 170
3.4.2 NODO DE PRESENCIA DE AGUA 175
3.4.3 NODO CORRIENTE 177

3.4.4 NODO TEMPERATURA 179

CAPÍTULO 4: PRUEBAS Y RESULTADOS

4.1 INTRODUCCION 183
4.2 PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA SECURE SYS. 183
4.2.1 PRUEBA DE ACTIVACION DE ALERTAS DE TEMPERATURA. 183
4.2.2 PRUEBA DE ACTIVACION DE ALERTAS DE PRESENCIA DE AGUA. 187
4.2.3 PRUEBA DE ACTIVACION DE ALERTAS DE PRESENCIA DE HUMO. ... 191
4.2.4 PRUEBA DE ACTIVACION DE ALERTAS DE CORRIENTE. 194
4.2.5 PRUEBA DE ACTIVACION DE ALERTA DE APERTURA DE PUERTA. ... 197
4.2.6 PRUEBA DE ALCANCE DE INTERFAZ MÓVIL. 199

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES 200
5.2 RECOMENDACIONES 202

REFERENCIAS BIBLIOGRÁFICAS…… 205

ANEXO A ..... A

ANEXO B ..... B

ANEXO C C

ANEXO D ..... D

INDICE FIGURAS

Figura 1-1.- Estructura del sistema SECURE SYS 3
Figura 1-2.- Trama RS-232 7
Figura 1-3.- Niveles de voltaje RS-232 8
Figura 1-4.- Topología de la Red RS-485 9
Figura 1-5.- Servicio SMS 12
Figura 1-6.- Esquema de Conexión de un sistema digital a la red GSM 13
Figura1-7.- Esquema de una pinza amperométrica de núcleo de hierro 17
Figura 1-8.- Esquema del sensor de Rogowski 18
Figura 1-9.- Esquema del sensor de Humo de Barrera Infrarroja 19
Figura 1-10.- Esquema del sensor de Humo de Efecto Tyndall 20
Figura 1-11.- Esquema del sensor de Humo de Laser 20
Figura 1-12.- Esquema del sensor de Humo de Ionización 21
Figura 1-13.- Sensor de CI DS18B20 1Wire 23

Figura 1-14.- ATmega 164P 24
Figura 1-15.- ATmega 48 25
Figura 1-16.- Panel de control de XAMPP 28
Figura 1-17.- Gestor de base de datos PHPMyAdmin 29
Figura 1-18.- Pantalla Inicial de NetBeans IDE 33
Figura 1-19.- Entornos de Java 34
Figura 2-1.- Distribución de pines circuito integrado MAX487 40
Figura 2-2.- Distribución de pines del circuito integrado SN75176 42
Figura 2-3.- Red RS-485 half-duplex con MAX487 43
Figura 2-4.- Esquema general del nodo maestro 46
Figura 2-5.- Diagrama esquemático del nodo maestro 47
Figura 2-6.- Circuito de regulación de voltaje a 5Vdc 48
Figura 2-7.- Fuente de alimentación +/- 12V 49
Figura 2-8.- Circuito de adaptación para el microcontrolador Atmega164 53
Figura 2-9.- Conexiones al microcontrolador 54
Figura 2-11.- Conexión del zócalo de programación 57
Figura 2-12.- Circuito de comunicación con la interfaz estacionaria 57
Figura 2-13.- Configuración MAX232 58
Figura 2-14.- Circuito para la comunicación RS-485 59
Figura 2-14.- Entrada opto-acoplada para el sensor magnético 60
Figura 2-15.- Entradas opto-acopladas para los sensores de humo 61
Figura 2-16.- Salida de relé para apertura de puerta 62
Figura 2-17.- Salida de relé para expansión 65
Figura 2-28.- Señalización de la comunicación con los nodos esclavos 66
Figura 2-29.- Demultiplexor 74HC154 67
Figura 2-30.- Señalización para el sensor de humo y magnético 69
Figura 2-31.- Esquema general del nodo de presencia de agua 71
Figura 2-32.- Circuito de detección de agua 72
Figura 2-33.- Diseño del filtro activo pasa bajos de segundo orden 72
Figura 2-34.- Implementación del filtro activo de Butterworth pasa bajos de segundo orden 74
Figura 2-35.- Circuito amplificador diferencial 75
Figura 2-36.- Diseño del amplificador diferencial para el nodo de presencia de agua 76
Figura 2-37.- Ventana de histéresis del circuito de comparación de voltaje 76
Figura 2-38.- Circuito comparador de voltaje con histéresis no inversor con desplazamiento de ventana 77
Figura 2-40.- Esquema general del circuito de adquisición de datos y procesamiento del nodo de presencia de agua 80

Figura 2-41.- Circuito de reset del microcontrolador 82
Figura 2-42.- Fuente de alimentación del nodo de presencia de agua 83
Figura 2-43.- Pines de programación del microcontrolador 84
Figura 2-44.- Entrada opto-acoplada del circuito de adquisición de datos del nodo de presencia de agua 85
Figura 2-45.- Sensor capacitivo de humedad en el ambiente 85
Figura 2-46.- Circuito oscilador para la mediación de humedad ambiental 86
Figura 2-47.- Esquema general del nodo de corriente 87
Figura 2-48.- Transductor de Corriente AKR 50 B420L 88
Figura 2-49.- Fuente de 24Vdc para sensor de corriente 88
Figura 2-50.- Diagrama de conexión del transductor de corriente AKR 50 B420L 89
Figura 2-51.- Circuito restador a base de un amplificador operacional 90
Figura 2-52.- Circuito de acondicionamiento de señal para el transductor de corriente 91
Figura 2-53.- Circuito de adquisición de datos y enlace a la red RS-485 del nodo de corriente 92
Figura 2-54.- Conector USB-A, empleado para adquirir la señal de voltaje proporcional a la corriente de alimentación 94
Figura 2-55.- Fuente de alimentación del nodo de corriente 95
Figura 2-56.- Circuito de enlace a la red RS-485 95
Figura 2-57.- Pines de programación del microcontrolador 96
Figura 2-58.- Esquema general del nodo de temperatura 97
Figura 2-59.- Sensor de temperatura 1wire DS18B20 [53] 97
Figura 2-60.- Circuito de adquisición de datos y enlace a la red RS-485 del nodo de Temperatura 101
Figura 2-61.- Fuente de alimentación del nodo de temperatura. 103
Figura 2-62.- Circuito de enlace a la red RS-485 104
Figura 2-63.- Pines de programación del microcontrolador 105
Figura 2-64.- Detector de humo 106
Figura 2-65.- Partes principales del sensor BRK 9120B [50] 107
Figura 2-66.- Señalización del sensor BRK 9120B [50] 108
Figura 2-67.- Señal de salida del pin de interconexión del sensor de humo 109
Figura 2-68.- Circuito de acondicionamiento para el sensor de humo 109
Figura 2-69.- Circuito equivalente del acondicionamiento del sensor de humo sin diodo zener 111
Figura 2-70.- Circuito equivalente del acondicionamiento del sensor de humo con diodo zener 112
Figura 2-71.- Señal acondicionada del sensor de humo 113
Figura 2-72.- Diagrama unificado del hardware del sistema SECURE SYS 114
Figura 2-73.- Ubicación del nodo de temperatura en el departamento técnico 116

Figura 2-74.- Ubicación del sensor de temperatura T1 para el departamento técnico ... 117
Figura 2-75.- Ubicación sensor de temperatura T2 para el departamento técnico 117
Figura 2-76.- Ubicación del nodo de presencia de agua en el departamento técnico 118
Figura 2-77.- Ubicación de electrodos en el departamento técnico 119
Figura 2-78.- Ubicación del detector de humo en el departamento técnico 120
Figura 2-79.- Ubicación del circuito de acondicionamiento del detector de humo para el departamento técnico 120
Figura 2-80.- Ubicación del nodo de temperatura en el taller 121
Figura 2-81.- Ubicación sensor de temperatura T3 en el taller 122
Figura 2-82.- Ubicación sensor de temperatura T4 en el taller 122
Figura 2-83.- Ubicación del nodo de presencia de agua en el taller 123
Figura 2-84.- Ubicación de electrodos en el área del taller 123
Figura 2-85.- Ubicación del detector de humo en el taller 124
Figura 2-86.- Ubicación del circuito de acondicionamiento del detector de humo para el taller 124
Figura 2-87.- Ubicación del nodo de corriente 125
Figura 2-88.- Ubicación del sensor de corriente 126
Figura 2-89.- Ubicación del sensor magnético 126
Figura 2-89.- Accionamiento de la cerradura eléctrica 127
Figura 3-1.- Diagrama de flujo de la aplicación para computador 129
Figura 3-2.- Ventana de configuración de puertos 130
Figura 3-3.- Ventana de error de configuración de puertos 131
Figura 3-4.- Ventana de error en la primera conexión a la base de datos 131
Figura 3-5.- Ventana de error en la segunda conexión a la base de datos 132
Figura 3-6.- Ventana de error en la tercera conexión a la base de datos 132
Figura 3-7.- Ventana de error en la comunicación con nodo maestro 133
Figura 3-8.- Ventana de error en la apertura del puerto para envío de SMS 133
Figura 3-9.- Ventana de error en la apertura del puerto para comunicación vía Bluetooth 134
Figura 3-10.- Ventana principal sin comunicación con el nodo maestro 134
Figura 3-11.- Ventana principal una vez establecida la comunicación con el nodo maestro 135
Figura 3-12.- Ventana Login para administración de la aplicación 136
Figura 3-13.- Ayudas en la ventana Login 137
Figura 3-14.- Error de usuario y/o contraseña 137
Figura 3-15.- Pestaña Parámetros de la ventana Administración 138
Figura 3-16.- Pestaña Usuarios de la ventana Administración 139
Figura 3-17.- Mensajes de ayuda para registro de usuarios 140

Figura 3-18.- Indicadores de datos inválidos en nuevo usuario 141
Figura 3-19.- Mensaje de error por no coincidencia de contraseñas 141
Figura 3-20.- Mensaje que alerta la ausencia de datos requeridos 142
Figura 3-21.- Mensaje de ingreso exitoso de un nuevo usuario 142
Figura 3-22.- Mensaje de ayuda para eliminación de usuarios 143
Figura 3-23.- Confirmación de eliminación de usuario 143
Figura 3-24.- Pestaña Conectividad de la ventana Administración 144
Figura 3-25.- Pestaña SMS/Mail de la ventana Administración 145
Figura 3-26.- Mensajes de ayuda de la pestaña SMS/Mail 146
Figura 3-27.- Pestaña Destinatario de la ventana Administración 147
Figura 3-28.- Mensajes de ayuda de la pestaña Destinatario 148
Figura 3-29.- Pestaña Agregar de la ventana Edición Destinatarios SMS 148
Figura 3-30.- Mensajes de ayuda de la pestaña Agregar de la ventana Editar Destinatario SMS 149
Figura 3-31.- Indicadores de datos inválidos de la ventana Editar Destinatario SMS 150
Figura 3-32.- Ventana de ingreso exitoso de un nuevo destinatario SMS 150
Figura 3-33.- Pestaña Eliminar de la ventana Edición Destinatario SMS 151
Figura 3-34.- Mensajes de ayuda para eliminar un destinatario SMS 151
Figura 3-35.- Confirmación de eliminación de un destinatario SMS 152
Figura 3-36.- Pestaña Agregar de la ventana Editar Destinatario Mail 152
Figura 3-37.- Mensajes de ayuda para el registro de destinatarios de correo electrónico 153
Figura 3-38.- Indicadores de datos inválidos en el registro de destinatarios de correo electrónico 153
Figura 3-39.- Pestaña Eliminar de la ventana Editar Destinatario Mail 154
Figura 3-40.- Mensajes de ayuda para eliminar destinatarios de correo electrónico 155
Figura 3-41.- Ventana Guardar Cambios 156
Figura 3-42.- Diagrama de flujo de la aplicación móvil 158
Figura 3-43.- Pantalla de bienvenida – “Bienvenido” 159
Figura 3-44.- Pantalla de identificación de usuario – “Login” 160
Figura 3-45.- Pantalla “Conectividad” 161
Figura 3-46.- Opciones de pantalla “Conectividad” 162
Figura 3-47.- Opciones de pantalla “Conectividad” – Botón Actualizar 162
Figura 3-48.- Menú principal – “Sensores” 163
Figura 3-49.- Opciones del menú principal 164
Figura 3-50.- Pantalla “Temperatura” 165
Figura 3-51.- Mensaje de error en la comunicación – Temperatura 166
Figura 3-52.- Pantalla “Sensor Corriente” 167

Figura 3-53.- Pantalla “Sensores Inundacion” 168
Figura 3-54.- Pantalla “Presencia de Humo” 168
Figura 3-55.- Pantalla “Puerta” 169
Figura 4-1.- Alerta de temperatura en el Taller 184
Figura 4-2.- Mensaje SMS de alerta de temperatura en el Taller 185
Figura 4-3.- Correo electrónico de alerta de temperatura en el Taller 185
Figura 4-4.- Alerta de temperatura en el Dpto.Técnico 186
Figura 4-5.- Mensaje SMS de alerta de temperatura en el Dpto. Técnico 187
Figura 4-6.- Correo electrónico de alerta de temperatura en el Dpto. Técnico 187
Figura 4-7.- Alerta de detección de agua en el Taller 188
Figura 4-8.- Mensaje SMS de alerta de detección de agua en el taller 188
Figura 4-10.- Correo electrónico de alerta de detección de agua en el Taller 189
Figura 4-11.- Alerta de detección de agua en el Dpto. Técnico 189
Figura 4-12.- Mensaje SMS de alerta de detección de agua en el Dpto. Técnico 190
Figura 4-13.- Correo electrónico de alerta de detección de agua en el Dpto. Técnico ... 190
Figura 4-14.- Alerta de detección de humo en el taller 191
Figura 4-15.- Mensaje SMS de alerta de detección de humo Taller 192
Figura 4-16.- Correo electrónico de alerta de detección de humo taller 192
Figura 4-17.- Alerta de detección de humo Dpto. Técnico 193
Figura 4-18.- Mensaje SMS de alerta de detección de humo Dpto. Técnico 193
Figura 4-19.- Correo electrónico de alerta de detección de humo Dpto. Técnico 194
Figura 4-20.- Alerta de sobrecorriente 195
Figura 4-21.- Alerta de subcorriente. 195
Figura 4-22.- Mensaje SMS de alerta de sobrecorriente 196
Figura 4-23.- Mensaje SMS de alerta de subcorriente 196
Figura 4-24.- Correo electrónico de alerta de sobrecorriente 197
Figura 4-25.- Correo electrónico de alerta de subcorriente 197
Figura 4-29.- Alerta de apertura de puerta 198
Figura 4-30.- Mensaje SMS de alerta de apertura de puerta 198
Figura 4-31.- Correo electrónico de alerta de apertura de puerta 199
Figura 4-32.- Alcance de la interfaz móvil 199

INDICE TABLAS

Tabla 1-1. Resumen de estándares de comunicación 10
Tabla 2-1.- Tabla de función para transmisión usando MAX487 41
Tabla 2-2.- Tabla de función para recepción usando MAX487 41
Tabla 2-3.- Tabla de función para transmisión usando CI SN75176 42
Tabla 2-4.- Tabla de función para recepción usando CI SN75176 43
Tabla 2-5.- Comparación Atmega164 vs. Atmega64 52
Tabla 2-6.- Función de los pines del microcontrolador Atmega164 53
Tabla 2-7.- Función del los pines de microcontrolador del nodo maestro 55
Tabla 2-8.- Tabla de función del demultiplexor 74HC154 67
Tabla 2-9.- Tabla de función del demultiplexor 74HC154 68
Tabla 2-10. Especificaciones requeridas para el microcontrolador 81
Tabla 2-11. Especificaciones requeridas para el microcontrolador del nodo de corriente 93
Tabla 2-12. Características del sensor de temperatura DS18B20 98
Tabla 2-13. Comandos para realizar la transferencia de datos 98
Tabla 2-14. Cadena de Bytes recibidos del sensor de temperatura 100
Tabla 2-15. Especificaciones requeridas para el microcontrolador 102

RESUMEN

El buen funcionamiento de los equipos en una empresa o industria y la seguridad de las personas que laboran en la misma son aspectos importantes que se deben cuidar constantemente. Por esta razón, se diseñó e implementó el sistema denominado SECURE SYS en las instalaciones de empresa UPDATECOM Cia. Ltda. para darle un cierto grado de seguridad en estos aspectos.

El objetivo de este sistema es la determinación y comunicación de estados de alerta por medio de mensajes de correo electrónico y SMS. Para esto, se sensan las variables siguientes: temperatura, corriente, presencia de humo, presencia de agua y apertura de puerta.

El hardware de SECURE SYS está conformado por un nodo maestro y una red de nodos esclavos que se comunican por medio del protocolo RS-485 para adquirir y procesar la información de las variables antes mencionadas.

El software de SECURE SYS consiste en dos aplicaciones. La primera aplicación fue diseñada para ser instalada en un computador, mediante la cual se pueden monitoriar las distintas variables y acceder a la configuración de ciertos parámetros del sistema en caso de ser usuarios registrados, siendo los más importantes la configuración de los límites de referencia para la determinación de alertas y la configuración de los destinatarios para envío de mensajes de correo electrónico y SMS. La segunda aplicación fue diseñada para correr en un dispositivo móvil, por medio de la cual se pueden hacer consulas del estado de las variables sensadas y comandar la apertura de la puerta de entrada, de forma remota.

Como resultado se consiguió una comunicación rápida y sin errores entre los diferentes nodos y una recepción de los mensajes de alerta en un tiempo menor a un segundo después de haberse suscitado la alerta.

PRESENTACIÓN

El presente trabajo fue desarrollado en cinco capítulos en los cuales se abordan los siguientes temas:

En el capítulo 1 se empieza por describir el problema propuesto y la solución que se da al mismo; a continuación, se detallan las distintas herramientas, tanto de hardware como de software, empleadas para el desarrollo de este proyecto.

En el capítulo 2 se explica el diseño e implementación de los componentes de hardware empleados para llevar a cabo la adquisición de datos, acondicionamiento de señal y comunicación.

En el capítulo 3 se expone el diseño y funcionamiento de cada una de las aplicaciones instaladas en la interfaz estacionaria y móvil.

En el capítulo 4 se presentan las pruebas realizadas al sistema y los resultados obtenidos en cada una de ellas.

En el capítulo 5 se exponen las conclusiones a las que se llegó al finalizar el proyecto y ciertas recomendaciones que son de utilidad al momento de usar el sistema SECURE SYS.

CAPÍTULO 1
MARCO TEÓRICO

1.1 INTRODUCCIÓN

Desde el inicio de la humanidad, el hombre ha buscado su bienestar. Por esta razón, a lo largo del tiempo, por medio del desarrollo de la ciencia, se han creado nuevas tecnologías y mejorado otras con el objetivo de incrementar su confort y seguridad. Así, en los últimos años, el avance agigantado de la electrónica ha permitido la implementación de sistemas de automatización en hogares, industrias, hospitales, oficinas, entre otros. Los cuales pueden ser capaces de realizar una acción específica al suscitarse un evento determinado como puede ser: el control de iluminación por comandos de voz, el envío de alertas al detectarse humo o presencia de agua, el control de apagado de equipos al no detectar presencia para lograr un ahorro energético, climatización del área de trabajo de acuerdo a la medición de temperatura.

Gracias a los innumerables beneficios que ha proporcionado este tipo de automatización, hoy en día está en pleno auge la implementación de sistemas domóticos e inmóticos.

1.2 PLANTEAMIENTO DEL PROBLEMA

En la actualidad las empresas que se dedican a la automatización de procesos industriales y/o a la instalación de sistemas automáticos dentro de ambientes de trabajo, emplean para sus actividades cotidianas equipos que pueden poner en riesgo la integridad de las instalaciones si se suscitara el caso en que estos queden encendidos sin supervisión o sean empleados de una manera incorrecta.

La empresa UPDATECOM Cia. Ltda., que realiza actividades como el desarrollo y mantenimiento de infraestructura de redes, el diseño de sistemas domóticos y el mantenimiento de equipos electrónicos, emplea para su desempeño diario equipos que deben ser manejados con cuidado para evitar percances. A pesar de que dentro de la empresa se mantienen reglas sobre la utilización de los diferentes equipos, el riesgo de que se produzca un accidente no es nulo, y de presentarse esta situación, las instalaciones y los equipos dentro de esta corren un alto riesgo de sufrir graves daños debido a la naturaleza de los equipos que se emplean.

1.3 SOLUCIÓN AL PROBLEMA

Para enfrentar este problema se dispone el diseño e implementación de un sistema de comunicación de alertas y monitoreo de variables que se consideran adecuadas para verificar el estado general de las instalaciones. Estas variables son temperatura, presencia de humo, presencia de agua, consumo de corriente, y el estado de abierto o cerrado de la puerta de entrada a la empresa. Este sistema determina estados de alerta cuando los valores de las variables sensadas se encuentren fuera de los valores preestablecidos como normales por el usuario.

De manera general el sistema de monitoreo (Figura 1-1), al que se hará referencia a partir de ahora como SECURE SYS, está conformado por lo siguiente:

- Nodos esclavos.- Circuitos encargados de obtener la información de tres de las diferentes variables sensadas (corriente, temperatura y presencia de agua) y de enviarlas al nodo maestro bajo petición del mismo.

    - Nodo de corriente.- Este nodo obtiene la información del valor TrueRMS de corriente que alimenta las instalaciones. Esta información es enviada al nodo maestro.

- Nodos de temperatura.- Se encargan de la obtención del valor de temperatura en las áreas de interés, a través de un par de sensores de temperatura con salida digital conectado a cada nodo.

- Nodos de presencia de agua.- Se encarga de determinar la presencia o no de agua cerca de las áreas de interés dentro de las instalaciones de la empresa. Esta información es enviada al nodo maestro bajo petición.

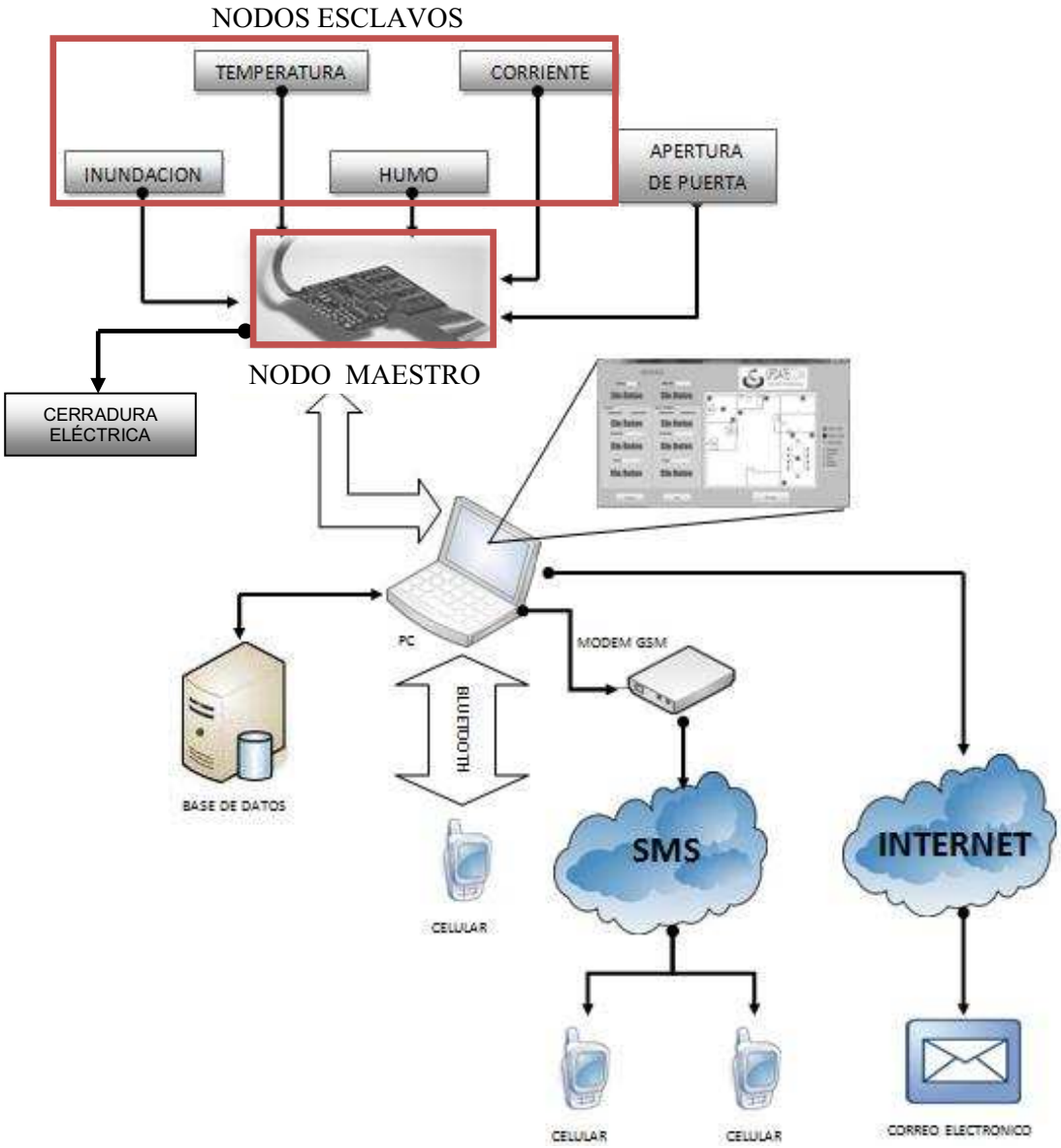


Figura 1-1.- Estructura del sistema SECURE SYS

- Nodo maestro.- Este nodo es el encargado de coordinar la comunicación con los demás nodos; a demás de adquirir y procesar la información de los nodos esclavos. El nodo maestro también se encarga de monitorear el estado de los sensores de humo y de apertura de puerta, y de comandar el accionamiento de la cerradura eléctrica de la puerta de ingreso a las instalaciones.

- Interfaz estacionaria.- Ordenador en el que se instala la aplicación que se encarga de administrar el sistema SECURE SYS. Esta aplicación tiene la tarea de comparar la información obtenida por la red de nodos de adquisición de datos con los valores establecidos como de normal funcionamiento por el usuario del sistema. Al presentarse una anomalía en los valores obtenidos, la aplicación se encarga de enviar una alerta escrita a través de mensajes de texto (empleando para esto el modem GSM) y correo electrónico, y de mostrar una alerta visible en el ordenador.

- Interfaz móvil.- El usuario puede realizar una consulta del estado del sistema aún estando alejado del ordenador, a través de una aplicación instalada en un dispositivo móvil, la cual se comunica mediante Bluetooth con la unidad de interfaz estacionario para obtener la información del sistema y comandar la apertura de la puerta de las instalaciones.

1.4 DÓMOTICA E INMÓTICA

La domótica es la automatización y control de una vivienda mediante la inclusión de sistemas de gestión de seguridad, climatización, energía, accesibilidad y comunicación. Una vivienda inteligente tiene coordinados los sistemas domótico, de distribución de energía y de comunicación.

Un sistema domótico tiene la capacidad de adquirir información a través de sensores o nodos del sistema, procesarla y tomar acciones basados en estos

datos y en los parámetros de funcionamiento que han sido configurados en el mismo; además, permite realizar un control y monitoreo del confort de la vivienda. Un sistema domótico puede constar de sistemas adicionales ideados por el diseñador, viéndose limitado únicamente por la imaginación del mismo.

De igual manera, la inmótica maneja la gestión de diferentes sistemas con el objetivo de proveer seguridad, comunicación, confort y gestión de energía. Sin embargo, la inmótica está orientada a ofrecer estas soluciones a inmuebles no destinados a la vivienda como: hospitales, edificios de oficinas, hoteles, centros comerciales y de formación.

1.4.1 APORTES DE LA DOMÓTICA E INMÓTICA. [21]

- EN SEGURIDAD:

Ofrece al usuario la capacidad de monitorear los sistemas de seguridad de su inmueble y de recibir alertas de seguridad por parte de este en caso de que se detecten anomalías. También, ofrece la posibilidad de simular la presencia de personas dentro del inmueble en los casos en que este permanezca desocupado por un largo periodo de tiempo.

- EN ACCESIBILIDAD:

El usuario de este tipo de sistemas puede realizar un control de diferentes dispositivos y cargas eléctricas dentro del inmueble desde unidades remotas o localizadas en lugares de fácil accesibilidad.

- EN ENERGÍA:

Se realiza un monitoreo continuo de la cantidad de energía que se consume, ofreciendo la posibilidad de controlar la cantidad de energía consumida por medio de la desconexión de sistemas no esenciales.

- EN COMUNICACIONES:

Se realiza una recopilación continua de información sobre el estado del inmueble. Esta información está disponible para el usuario bajo pedido, tanto en los nodos del sistema como en dispositivos remotos que se enlacen a este.

1.5 HERRAMIENTAS DE SOLUCIÓN

A continuación se exponen las herramientas consideradas para llevar a cabo la solución planteada.

1.5.1 ESTÁNDARES DE COMUNICACIÓN

1.5.1.1 EIA RS-232

La institución de normalización americana (EIA[1]) ha escrito la norma RS[2]-232-C que regula el protocolo de transmisión de datos, el cableado, las señales eléctricas y los conectores en los que debe basarse una conexión RS-232. [2]
La transmisión de datos es serial digital a través de una línea asimétrica entre dos equipos, un DTE[3] y un DCE[4]. La distancia máxima de enlace es de 15 metros y la velocidad de transmisión máxima es de 20 kbps. [1]

1.5.1.1.1 Funcionamiento [3]

Cada palabra es sincronizada con un bit de Start, uno de Stop y una frecuencia interna a cada uno de los dispositivos acordada previamente.
En el estándar RS-232 C se tiene siempre la línea de datos a "1", pasando esta a "0" cuando se quiere empezar la transmisión. Se trabaja con lógica negativa, es

[1] Electronics Industries Association (Asociación de Industrias Electrónicas) [1]
[2] Recommended Standard (Estandar Recomendado)
[3] Data Terminal Equipment (Equipo de Datos Terminal)[1]
[4] Data Communications Equipment (Equipo de Comunicación de Datos) [1]

decir, cuando en la línea de datos aparece un nivel de tensión positivo significa que se está recibiendo un "0" y si es negativo un "1".
La forma de onda típica de la transmisión de una palabra es:

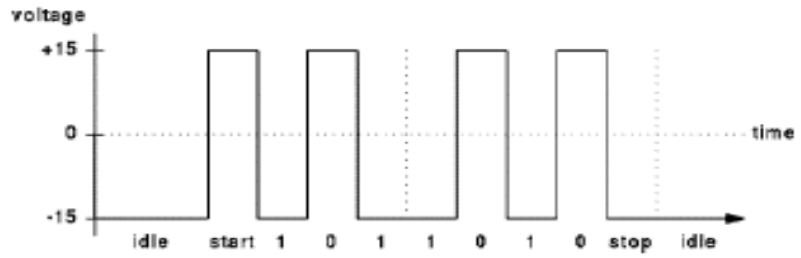


Figura 1-2.- Trama RS-232 [4]

1.5.1.1.2 Niveles de voltaje [3]

Los niveles de voltaje en las comunicaciones RS-232 C son distintos según se trate de entrada de datos o de salida de estos. El término salida se refiere a la transferencia de datos desde un ordenador a un dispositivo externo. Recíprocamente, la trasferencia de datos desde un dispositivo externo al ordenador se conoce como entrada.

Sus voltajes pueden oscilar entre +15 y -15 voltios. La única diferencia entre la definición de salida y de entrada es el ancho de la región de transición, de -3V a +3V en la entrada y de -5V a +5V en la salida. Esta diferencia entre las definiciones de voltajes mínimos permisibles se conoce como el margen de ruidos del circuito. Este margen de seguridad es de gran utilidad cuando los cables deben pasar por zonas cercanas a elementos que generan interferencias eléctricas como motores, transformadores, reguladores o equipos de comunicación.

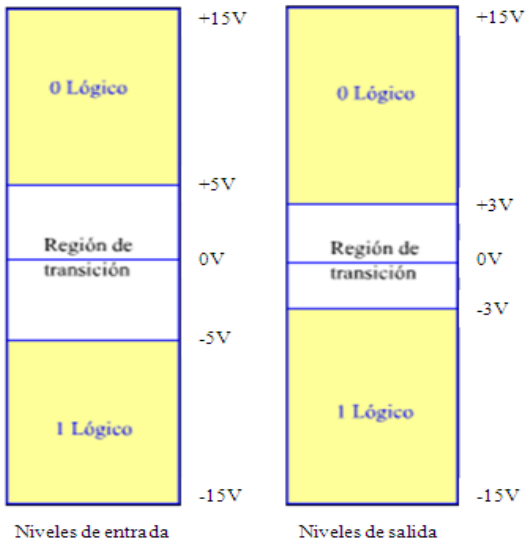


Figura 1-3.- Niveles de voltaje RS-232

1.5.1.2 EIA RS-485

El estándar EIA RS-485 define un sistema en bus de transmisión multipunto diferencial, el cual es ideal para transmitir a altas velocidades sobre largas distancias (35 Mbps hasta 10 metros y 100 Kbps en 1.200 metros) y a través de canales ruidosos, ya que reduce el ruido que aparece en la línea de transmisión. El medio físico de transmisión es un par entrelazado que admite hasta 32 estaciones en 1 solo hilo, con una longitud máxima de 1.200 metros operando entre 300 y 19.200 bps y la comunicación half-duplex (semiduplex). La transmisión diferencial permite múltiples drivers dando la posibilidad de una configuración multipunto (32 transmisiones y 32 receptores). Al tratarse de un estándar bastante abierto permite muchas y muy diferentes configuraciones y utilizaciones. [5]

1.5.1.2.1 Topología [7]

Cuando se emplean los receptores RS485 comunes, con una resistencia de entrada de 12 kΩ, es posible conectar 32 dispositivos en la red. Mientras tanto, si

se utilizan receptores RS485 con alta impedancia de entrada, actualmente disponibles, el número de dispositivos podrá ser expandido a 256.
Repetidores RS485 pueden hacer posible incrementar el número de nodos a varios cientos, abarcando algunos kilómetros.

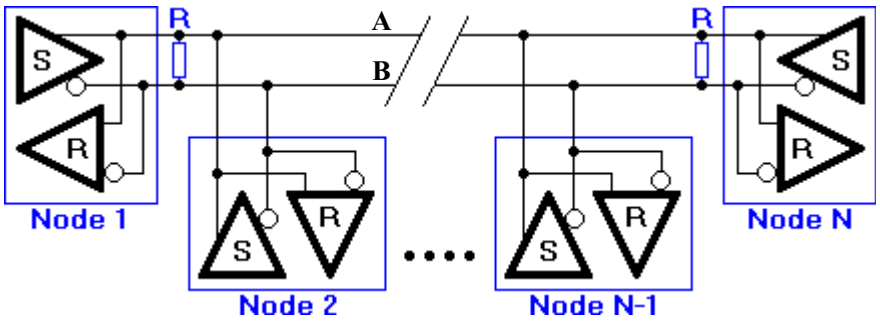


Figura 1-4.- Topología de la Red RS-485

1.5.1.2.2 Funcionamiento [7]

Por defecto, todos los transmisores en el bus RS485 están en alta impedancia. En protocolos de alto nivel, uno de los nodos es definido como master, el cual envía peticiones o comandos por el bus RS485. Todos los otros nodos reciben estos datos. Dependiendo de la información contenida en los datos enviados, uno o más nodos en la línea pueden o no responder al master. Así, el ancho de banda puede ser utilizado al 100%.
Existen otras implementaciones de las redes RS485, donde cada nodo puede empezar una sesión de datos por su cuenta. Esto es comparable con la forma en la que la red Ethernet funciona. Debido a que se produce una colisión de datos en esta implementación, teóricamente el 37% del ancho de banda será efectivamente utilizado; por esta razón, es necesario que se utilice un protocolo con detección de errores para activar el reenvío de información en caso de que esta haya sido corrompida.

1.5.1.2.3 Niveles de voltaje [6]

En el protocolo RS485, las líneas de transmisión A y B (ver Figura 1-4) funcionan como par diferencial (sin tierra absoluta como el RS232). El 1 lógico se representa por la condición en que el voltaje en la línea A, es mayor que el de la línea B, y al contrario para el 0 lógico. Típicamente A - B = 5 volts para 1 lógico y A - B = -5 volts para 0 lógico.
La comunicación RS485 es bidireccional Half duplex, a diferencia del RS422 que usa los mismos niveles de voltaje pero con 2 pares de líneas para comunicación Full duplex.

A continuación se muestra un resumen de las características de los distintos estándares de comunicación:

Tabla 1-1. Resumen de estándares de comunicación [1]

Estándar	RS232	RS423	RS422	RS485
Modo de trabajo	Asimétrica	Asimétrica	Diferencial	Diferencial
N° drivers/receivers	1 driver 1 receiver	1 driver 10 receivers	1 driver 10 receivers	32 drivers 32 receivers
Longitud max. Cable	15 m	1200 m	1200 m	1200 m
Velocidad max.	20 kb/s	100 kb/s	10 Mb/s	10 Mb/s
Carga por driver	3 kΩ a 7 kΩ	450 Ω min.	100 Ω min	54 Ω
Rango tensión de entrada en receptor	± 15 V	± 12 V	± 7 V	-7V a 12 V
Sensibilidad del receptor	± 3 V	± 200 mV	± 200 mV	± 200 mV

1.5.2 TECNOLOGÍAS DE COMUNICACIÓN

1.5.2.1 Bluetooth [2]

Bluetooth es la norma que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes equipos mediante

radiofrecuencia. Los principales objetivos que se pretenden conseguir son esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre estos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre diferentes dispositivos.

La especificación de Bluetooth define un canal de comunicación máximo de 720 Kb/s con rango óptimo de 10m (opcionalmente 100m).

La frecuencia de radio con la que trabaja está en el rango de 2.4 a 2.48 GHz con amplio espectro y saltos de frecuencia capaz de transmitir en Full Duplex con un máximo de 1600 saltos/seg. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1MGhz; esto permite dar seguridad y robustez.

1.5.2.2 Servicio SMS [23]

El servicio de mensajes cortos o SMS (Short Message Service) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. SMS fue diseñado originariamente como parte del estándar de telefonía móvil digital GSM[5], pero en la actualidad está disponible en una amplia variedad de redes, incluyendo las redes 3G. El inconveniente es que los mensajes SMS tienen una longitud limitada de 160 caracteres en GSM, 256 bytes en IS-136 (TDMA[6]) y 255 caracteres en CDMA[7]. También, el uso de los canales de control limita el ancho de banda disponible para los servicios SMS.

Técnicamente el servicio SMS consiste en transferir un mensaje de texto entre una estación móvil (MS[8]) y otra entidad (SME[9]) a través de un centro de servicio

[5] Global System for Mobile Communication (Sistema Global de Comunicación Móvil) [24]
[6] Time-Division Multiple Access (Acceso Múltiple por División de Tiempo) [35]
[7] Code-Division Multiple Access (Acceso Múltiple por División de Código) [35]
[8] Mobile Station (Estación Móvil) [23]
[9] Short Messaging Entity (Entidad de Mensajería Corta) [24]

(SC[10]). El servicio final ofrecido es una comunicación extremo-extremo entre la estación móvil (MS) y la entidad (SME). La entidad puede ser otra estación móvil o puede estar situado en una red fija. En el caso de envío de un mensaje entre dos móviles, ambas partes son estaciones móviles. Cuando se envía un mensaje para solicitar algún tipo de servicio, un extremo es una estación móvil y la otra es un servidor que atiende las peticiones.

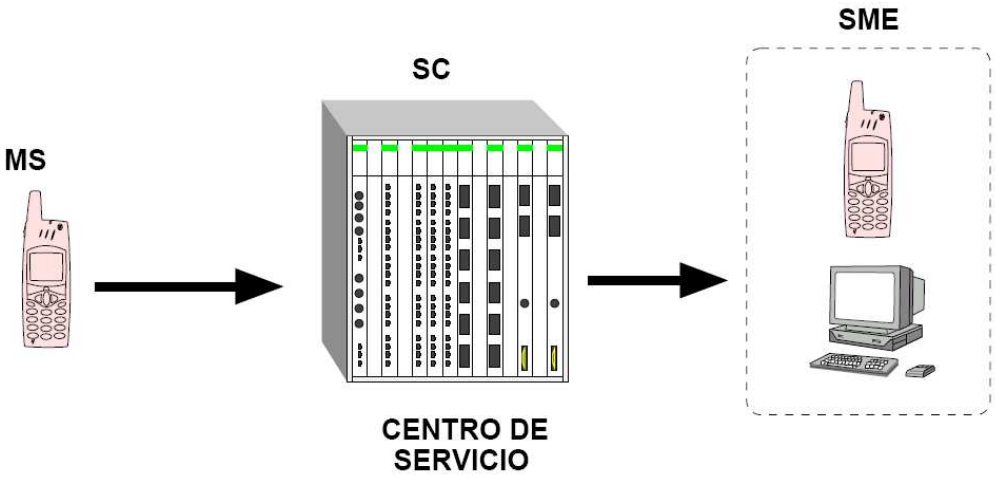


Figura 1-5.- Servicio SMS [24]

1.5.2.2.1 Acceso al Servicio SMS [24]

Actualmente están apareciendo gran cantidad de servicios basados en mensajes cortos. Para hacer posible la utilización de estos servicios se ha visto necesaria la creación de software y hardware que tenga la capacidad de acceder a los servicios SMS. Esto se puede lograr de las siguientes maneras:

- Algunos teléfonos se pueden conectar directamente a un PC y mediante un software propietario se puede acceder a los datos del móvil (agenda, tarjeta SIM), así como enviar y recibir mensajes SMS. El principal problema

[10] Service Center (Centro de Servicio) [23]

de esta solución es que no es abierta, y los fabricantes no proporcionan suficiente información como para poder realizar aplicaciones con ellos.

- Utilización de un MÓDEM GSM. Mediante un MÓDEM GSM se puede conectar cualquier sistema digital a la red GSM, no sólo para enviar mensajes SMS sino también para transmitir datos.

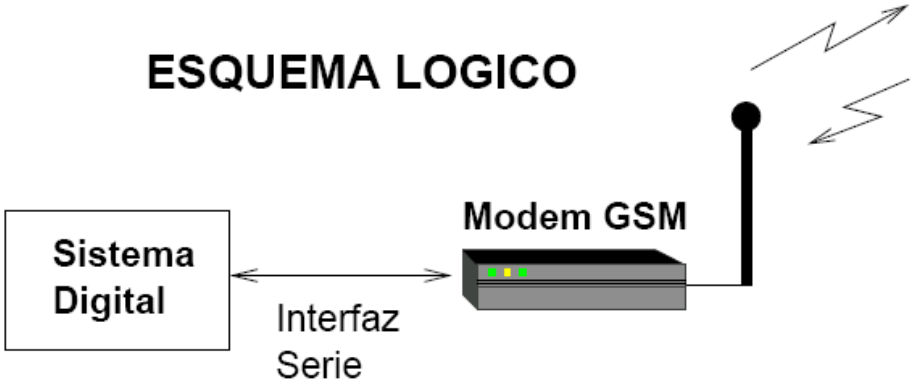


Figura 1-6.- Esquema de Conexión de un sistema digital a la red GSM

1.5.2.3 Correo electrónico [25]

El correo electrónico, también llamado E-MAIL (Electronic Mail), es una forma de enviar correo, mensajes o cartas electrónicas de un ordenador a otro. Tanto la persona que envía el correo electrónico, como la persona que lo recibe, deben tener una cuenta de correo en INTERNET.

El E-MAIL es un servicio que ofrece prácticamente todo tipo de redes existentes. La diferencia es la forma de ese E-MAIL, que en muchos casos será diferente para las distintas redes. El E-MAIL es mucho más rápido que el correo tradicional. Cuando se envía correo electrónico, puede ser cuestión de minutos que llegue a su destino, sea cual sea el lugar del mundo donde se encuentre el destinatario del mensaje. El mensaje electrónico pasa de un servidor a otro. Cada servidor que recibe un mensaje, comprueba la dirección y lo envía por la ruta correcta a otro servidor. Este proceso se repite hasta que el mensaje llega al servidor de destino,

entonces se almacena en el buzón electrónico correspondiente (espacio de disco destinado a almacenar el correo electrónico de un usuario de dicho servidor). Sin embargo con el correo tradicional suele ser cuestión de días, semanas e incluso meses.

1.5.2.3.1 Tipos de Correo Electrónico

- Correo POP[11]

Este tipo de correo se hace a través de una conexión a internet, pero sin necesidad de estar conectado permanentemente a la red, simplemente se conecta a través de un módem o ADSL[12] a un servidor dedicado de correo (llamado POPSERVER), y éste se conecta para recibir, enviar los mensajes que estén en nuestro buzón o casilla de correo y corta la comunicación. Esto se hace a través de programas específicos (Outlook Express, Eudora, Pegasus Mail, Nestcape Communicator, etc.) Los mensajes se borran del servidor y se almacenan en el disco rígido de la PC del usuario. Se tiene capacidad ilimitada de almacenar mails, es más rápido y los mensajes se pueden escribir y leer sin necesidad de estar conectados a la red, se pueden consultar en cualquier momento, redactarlos y enviarlos mas tarde. La conexión solo se realiza al enviar y recibir.
Generalmente no se pueden consultar los mensajes recibidos desde otras PC o vía Web, hay mayor posibilidad de ingreso de virus, es más difícil combatir y filtrar el correo no deseado (SPAM).

- Web Mail o Correo Web:

Este es un servicio de correo generalmente gratuito que se encuentra en portales y/o buscadores (HOTMAIL, YAHOO, GMAIL, UOL, UBBI, etc.) que ofrecen este servicio, en donde un usuario se subscribe (define su nombre de usuario o ID y contraseña), y obtiene su casilla de correo personal.

[11] Post Office Protocol (Protocolo de Oficina Postal) [25]
[12] Asymmetric Digital Subscriber Line (Línea Asimétrica de Subscriptor Digital) [26]

Para usar este tipo de correo es imprescindible contar con una conexión a Internet (pagada o gratis) y un navegador o browser. Para leer y escribir hay que estar permanentemente conectado (ONLINE), es lento, y la cantidad de mails que almacena están limitados, el tamaño de los archivos que se envían, etc. Los mensajes quedan en el servidor donde está alojada la cuenta de correo Web (en el disco rígido del servidor de correo).
Para utilizar este correo sólo hace falta el navegador o browser, ya que, al entrar al servicio, se convierte en el administrador de correo.
Este tipo de correo tiene filtros antispam y antivirus y se puede consultar desde cualquier PC.

1.5.3 HARDWARE

1.5.3.1 Sensores de corriente [27]

La medición de este parámetro presenta una exigencia tecnológica importante dado el gran rango de variación al que está sometida la corriente. Es mandatorio contar con sensores de rango dinámico no inferiores a 500:1, o bien la posibilidad de modificar el alcance para adecuarlos a los distintos regímenes de carga.
Se distinguen dos tipos principales según la geometría del circuito magnético: transductor de geometría fija y transductor de geometría variable.

Los transductores de geometría rígida tienen menos dependencia en su exactitud en función del conexionado o instalación, ya que su estructura física es constante y no varía entre una medida y otra. Se destacan los transformadores de intensidad de núcleo cerrado y los shunts[13]. En ambos casos la principal desventaja para un trabajo “in situ” es que requieren interrumpir el circuito para su instalación. El caso del shunt posee una desventaja adicional y es que queda vinculado eléctricamente a la red, constituyendo esto una verdadera limitación en sistemas trifásicos.

[13] Derivaciones de línea

Los transductores de geometría variable también conocidos como transformadores de corriente de núcleo partido o abrible, son los preferidos para medición en terreno ya que reducen el riesgo operativo, agilizan la operatoria y permiten su instalación sin la interrupción del circuito eléctrico. Los de uso más difundido son los transductores tipo pinza de núcleo ferroso y los transductores flexibles de núcleo de aire.

1.5.3.1.1 Pinza amperométrica de núcleo de hierro

- Principio de funcionamiento.

El material constructivo del núcleo posee una excelente permeabilidad magnética (μh 2000 a 60000) lo que le confiere una sensibilidad excepcional (entre 1mV/A y 100mV/A) presentando una muy buena relación señal ruido, aún midiendo bajos niveles de corriente.

La tensión entregada por el transductor será proporcional a la variación del flujo magnético en su núcleo, y por tanto será proporcional al campo magnético y al área del núcleo.

Para el diseño de pinzas amperométricas concebidas para la medición de grandes corrientes (miles de amperes) deberá tenerse en cuenta que la sección del núcleo sea adecuada para evitar la saturación del mismo. De aquí se deriva la necesidad de un aumento en el volumen (y por consecuente la masa) del núcleo.

La curva típica del hierro es una limitación importante para la construcción de pinzas de alto rango dinámico. Actualmente, el tratamiento digital de la señal mejora este déficit logrando alcanzar rangos de hasta 1000:1.

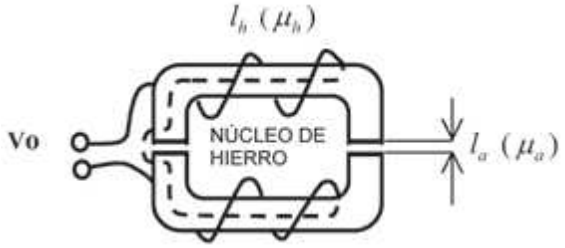


Figura1-7.- Esquema de una pinza amperométrica de núcleo de hierro

- Sensibilidad posicional

Excepto en pinzas de pequeño diámetro o dimensión de ventana, siempre hay una incidencia en la exactitud por la posición relativa del conductor dentro de la misma. El desajuste del entrehierro también afectará significativamente la sensibilidad posicional, obligando a procurar el alineamiento adecuado de los transductores al instalarlos en el circuito de medida, lo cual no siempre es factible en el terreno.

1.5.3.1.2 Transductor de núcleo de aire

El transductor de núcleo de aire es también llamado Bobina de Rogowski.

- Principio de funcionamiento

El sensor de corriente está constituido por un inductor con núcleo de aire de tipo toroidal, cuya salida de tensión es proporcional a la derivada de la corriente a medir.

Dadas las características magnéticas del núcleo, la curva de respuesta a diferencia del hierro, es absolutamente lineal. No existe saturación del núcleo por lo cual puede medir grandes corrientes sin preocupación por deformación o rotura. Tampoco requiere de circuitos o algoritmos de compensación digital de su salida.

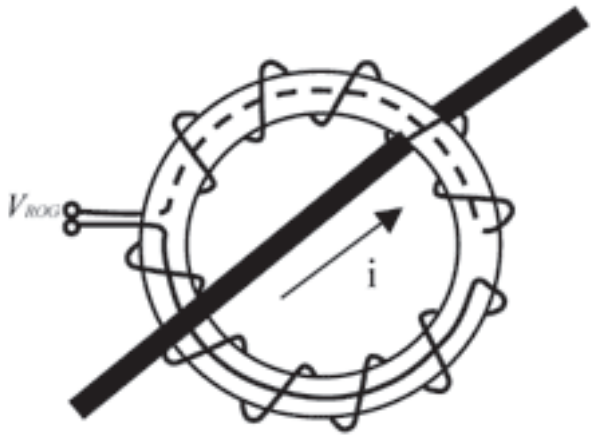


Figura 1-8.- Esquema del sensor de Rogowski

La característica derivativa de la señal entregada por la bobina de Rogowski obliga a una integración en el tiempo a fin de obtener una señal proporcional a la corriente a medir.

En mediciones de frecuencia industrial y hasta algunos miles de hertz la integración es llevada a cabo por un integrador del tipo activo analógico o digital. Este acondicionamiento de la señal hace necesario disponer de tensiones de alimentación, lo que en determinados casos se torna poco práctico. Esto no constituye una limitación si el sistema de medición contempla de antemano el empleo de este tipo de transductor.

1.5.3.2 Sensores de humo [28]

1.5.3.2.1 Detector óptico

Puede ser de varios tipos, según detecten el humo por oscurecimiento o por dispersión del aire en un espacio:

- Barrera infrarroja

Compuestos por un dispositivo emisor y otro receptor.
Cuando se oscurece el espacio entre ellos debido al humo sólo una fracción de la luz emitida alcanza al receptor provocando que la señal eléctrica producida por éste sea más débil y se active la alarma.

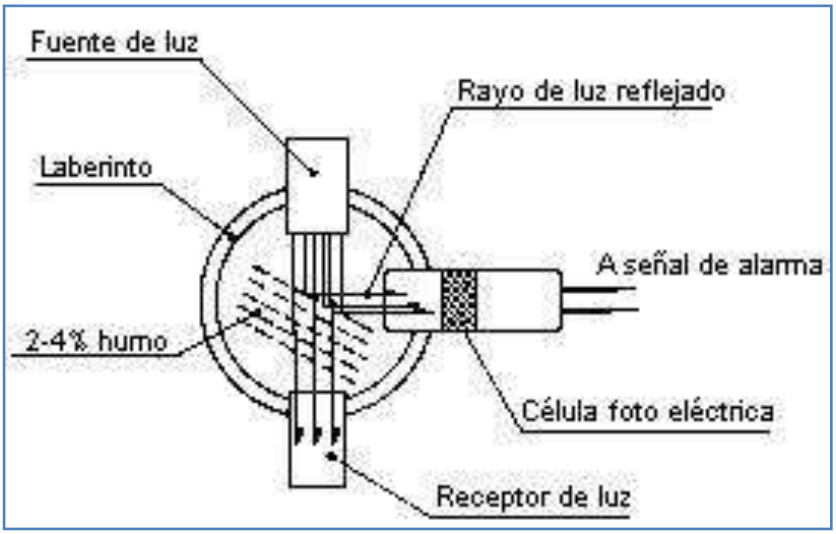


Figura 1-9.- Esquema del sensor de Humo de Barrera Infrarroja

- Puntual o de efecto Tyndall

El emisor y receptor se encuentran alojados en la misma cámara pero no se ven al formar sus ejes un ángulo mayor de 90º y estar separados por una pantalla, de manera que el rayo emitido no alcanza el receptor.
Cuando entra humo en la cámara el haz de luz emitido se refracta y puede alcanzar al receptor, activándose la alarma

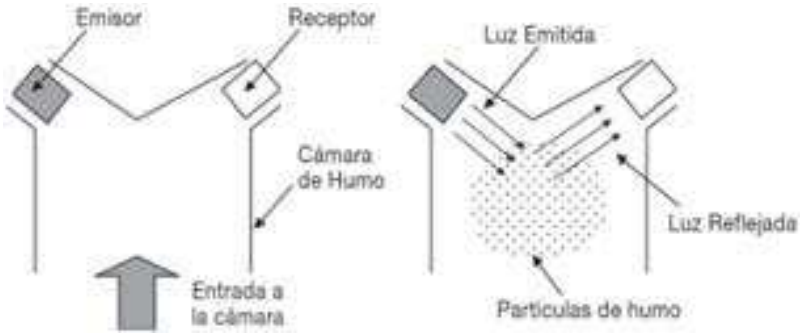


Figura 1-10.- Esquema del sensor de Humo de Efecto Tyndall [29]

- Detector láser

Detectan oscurecimiento de una cámara de aglutinación con tecnología láser

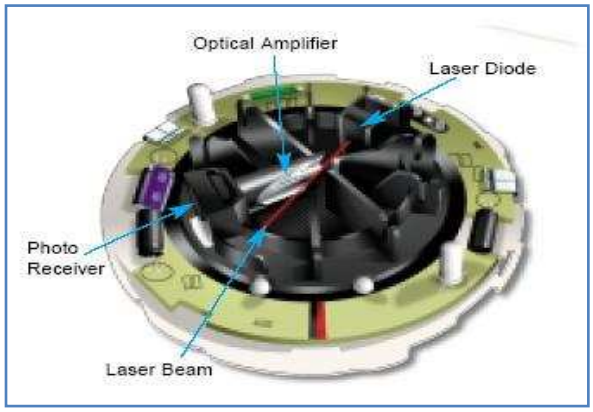


Figura 1-11.- Esquema del sensor de Humo de Laser

1.5.3.2.2 Detector iónico

Este tipo de detector es más barato que el óptico y puede detectar partículas que son demasiado pequeñas para influir en la luz.

Está compuesto por una pequeña cantidad del isótopo radioactivo (americio-241) que emite radiación alfa.
La radiación pasa a través de una cámara abierta al aire en la que se encuentran dos electrodos, permitiendo una pequeña y constante corriente eléctrica.
Si entra humo en la cámara se reduce la ionización del aire y la corriente disminuye o incluso se interrumpe, con lo que se activa la alarma.

Figura 1-12.- Esquema del sensor de Humo de Ionización

1.5.3.3 Sensores de temperatura [36]

1.5.3.3.1 Termopares

Los termopares se utilizan extensamente, ya que ofrecen una gama de temperaturas mucho más amplia y una construcción más robusta que otros tipos. Además, no precisan alimentación de ningún tipo y su reducido precio los convierte en una opción muy atractiva para grandes sistemas de adquisición de datos. Sin embargo, para superar algunos de los inconvenientes inherentes a los termopares y obtener resultados de calidad, es importante entender la naturaleza de estos dispositivos.
Estudios realizados sobre el comportamiento de termopares han permitido establecer tres leyes fundamentales:

Ley del circuito homogéneo. En un conductor metálico homogéneo no puede sostenerse la circulación de una corriente eléctrica por la aplicación exclusiva de calor.

Ley de metales intermedios. Si en un circuito de varios conductores la temperatura es uniforme desde un punto de soldadura A a otro punto B, la suma algebraica de todas las fuerzas electromotrices es totalmente independiente de los conductores metálicos intermedios y es la misma que si se pusieran en contacto directo A y B.

Ley de las temperaturas sucesivas. La f.e.m. generada por un termopar con sus uniones a las temperaturas T1 T3 es la suma algebraica de la f.e.m. del termopar con sus uniones a T1 T2 de la f.e.m. del mismo termopar con sus uniones a las temperaturas T2 T3.

1.5.3.3.2 Sensores de circuito integrado (CI)

Los sensores de circuitos integrados resuelven el problema de la linealidad y ofrecen altos niveles de rendimiento. Son, además, relativamente económicos y bastante precisos a temperatura ambiente.

Sin embargo, los sensores de CI no tienen tantas opciones de configuraciones del producto o de gama de temperaturas, y además son dispositivos activos, por lo que requieren una fuente de alimentación.

Los sensores de CI forman parte de la tendencia hacia los "sensores inteligentes", que son unos transductores cuya inteligencia incorporada facilita las actividades de reducción y análisis de datos que el usuario debe realizar normalmente en el sistema de adquisición de datos.

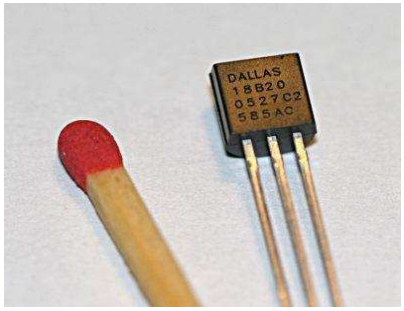


Figura 1-13.- Sensor de CI DS18B20 1Wire [37]

1.5.3.4 Detectores de presencia de agua

Los detectores de presencia de agua permiten conocer la existencia de acumulación de agua en un área determinada. Generalmente estos se basan en la conductividad del agua para determinar su presencia.

Se localizan generalmente cerca de las fuentes de agua como lavabos o inodoros, también se colocan cerca de los lugares en los que se puede acumular liquido.

1.5.3.5 Microcontroladores AVR ATmega

1.5.3.5.1 Microcontrolador ATmega 164P [38] [39]

El ATmega164P es un microcontrolador CMOS[14] de 8 bits de bajo consumo basado en la arquitectura RISC[15] mejorada. Sus instrucciones se ejecutan en un ciclo de máquina, el ATmega164P (Figura 1-14) consigue transferencia de información alrededor de 1 MIPS[16] por MHz admitido por el sistema, permitiendo al diseñador del sistema optimizar el consumo de energía versus la velocidad de procesamiento.

[14] Complementary Metal Oxide Semiconductor (Semiconductor de Óxido/Metal Complementario) [41]
[15] Reduced Instruction Set Computer (Computador con Set de Instrucciones Reducido) [42]
[16] Milliones de Instrucciones por Segundo [43]

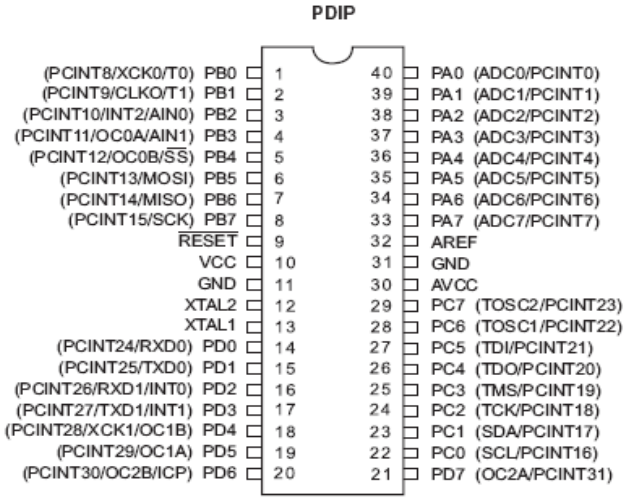


Figura 1-14.- ATmega 164P

Características de los periféricos:

- Dos Timer/Contadores de 8 bits con prescalamiento separado y modo comparación.
- Un Timer/Contador de 16 bits con prescalamiento separado, modo comparación y modo de captura.
- Contador en Tiempo Real con Oscilador separado
- 6 Canales para PWM[17]
- ADC[18] de 10 bits y 8 canales
- Modo Diferencial con ganancia seleccionable a x1, x10 o x200.
- Interfase serie de dos hilos con byte orientado.
- Dos puertos Seriales USART Programables
- Interfaz Serial SPI maestro-esclavo
- Watchdog Timer programable con oscilador independiente, dentro del mismo chip.

[17] Pulse Width Modulation (Modulación de Ancho de Pulso)
[18] Analog Digital Converter (Conversor Análogo Digital)

Características especiales del microcontrolador:

- Power-on Reset (en el encendido) y detección de Brown-out (pérdida de polarización) programable.
- Oscilador RC interno calibrado.
- Fuentes de interrupción externas e internas.
- 6 modos de descanso: Idle, Reducción de Ruido ADC, Power-save, Power-down, Standby y Standby extendido.
- Comparador Analógico dentro del mismo Chip

1.5.3.5.2 Microcontrolador ATmega 48. [40]

El ATmega48 es un microcontrolador CMOS de 8 bits de bajo consumo basado en la arquitectura RISC mejorada. Sus instrucciones se ejecutan en un ciclo de máquina, el ATmega164P (Figura 1-15) consigue transferencia de información alrededor de 1 MIPS por MHz admitido por el sistema, permitiendo al diseñador del sistema optimizar el consumo de energía versus la velocidad de procesamiento.

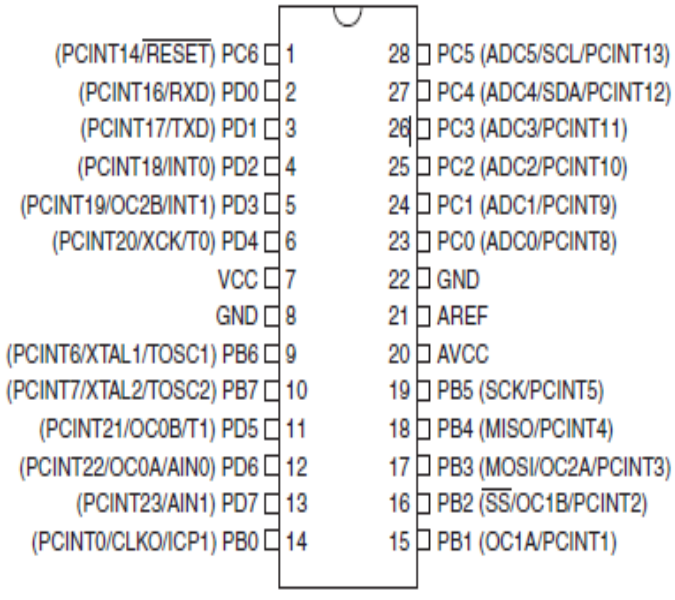


Figura 1-15.- ATmega 48

Características de los periféricos:

- Dos Timer/Contadores de 8 bits con prescalamiento separado y modo comparación.
- Un Timer/Contador de 16 bits con prescalamiento separado, modo comparación y modo de captura.
- Contador en Tiempo Real con Oscilador separado
- 6 Canales para PWM
- ADC de 10 bits y 8 canales
- Modo Diferencial con ganancia seleccionable a x1, x10 o x200.
- Interfase serie de dos hilos con byte orientado.
- Dos puertos Seriales USART Programables
- Interfaz Serial SPI maestro-esclavo
- Watchdog Timer programable con oscilador independiente, dentro del mismo chip.

Características especiales del microcontrolador:

- Power-on Reset (en el encendido) y detección de Brown-out (pérdida de polarización) programable.
- Oscilador RC interno calibrado.
- Fuentes de interrupción externas e internas.
- 5 modos de descanso: Idle, Reducción de Ruido ADC, Power-save, Power-down y Standby.

1.5.4 SOTWARE

1.5.4.1 Visual C#

Es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework[19]. C# es simple, eficaz, seguro y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C. [11]

Visual Studio admite Visual C# con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas. La biblioteca de clases .NET Framework ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa. [11]

Es software libre y multiplataforma. [12] Aunque C# forma parte de la plataforma .NET, es un lenguaje de programación independiente, diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows, Unix y GNU/Linux. [13]

1.5.4.2 XAMPP

XAMPP es un servidor, independiente de plataforma, de código libre. Permite instalar de forma sencilla Apache en el ordenador, sin importar el sistema operativo (Linux, Windows, MAC o Solaris). Incluye además servidores de bases de datos como *MySQL* y *SQLite* con sus respectivos gestores *phpMyAdmin* y

[19] Infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican el desarrollo de aplicaciones. [10]

phpSQLiteAdmin. Incorpora también el intérprete de PHP[20], el intérprete de Perl, servidores de FTP como *ProFTPD* ó *FileZilla* FTP[21] Server. [30]

Su nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. [31]
El panel de control de XAMPP se observa en la siguiente figura:

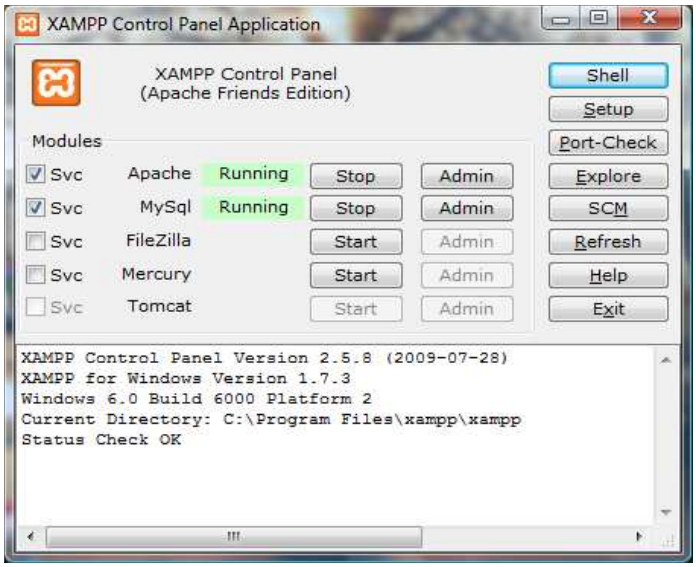


Figura 1-16.- Panel de control de XAMPP

1.5.4.2.1 Aplicaciones [31]

Oficialmente, los diseñadores de XAMPP sólo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica, sin embargo, XAMPP es utilizado actualmente para servidor de sitios Web y, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo.

[20] Hypertext Preprocessor: Código abierto, generalmente usado para desarrollo web. [32]
[21] File Transfer Protocol: Sistema para enviar y recibir fichero a través de la Internet. [34]

1.5.4.3 Gestor de bases de datos phpMyAdmin [33]

PhpMyAdmin es un programa de libre distribución, que consiste en una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva, como se muestra en la figura:

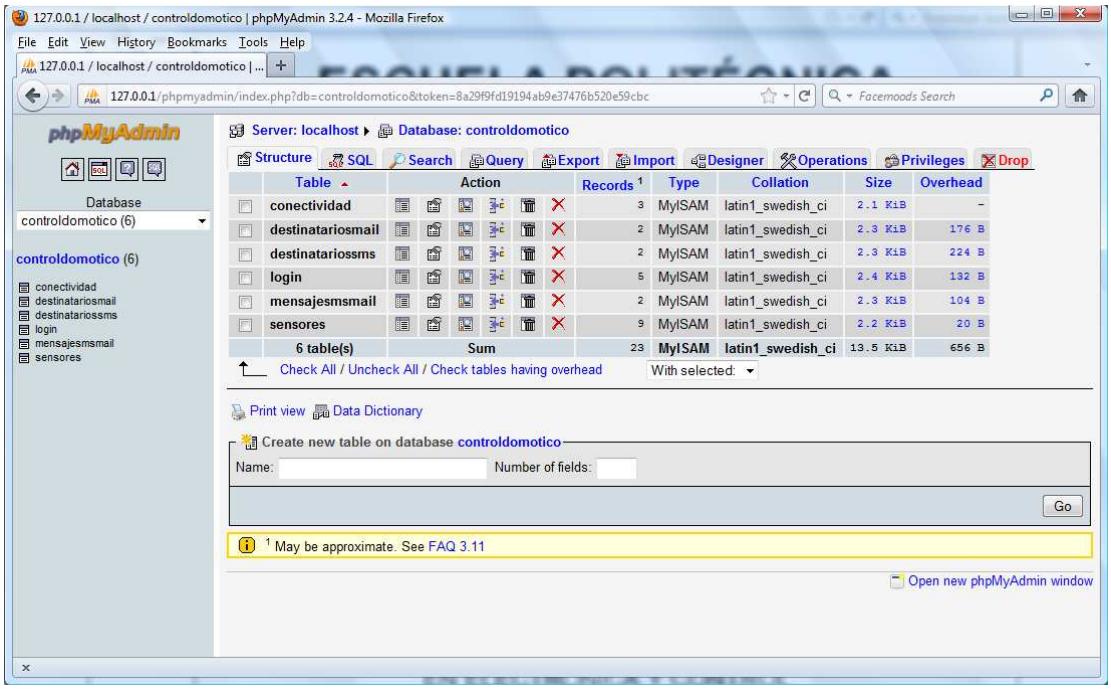


Figura 1-17.- Gestor de base de datos PHPMyAdmin

La aplicación en si es un conjunto de archivos escritos en PHP, que se pueden copiar en un directorio de un servidor web, de modo que al acceder a esos archivos se muestran unas páginas donde se pueden encontrar las bases de datos y todas sus tablas, a las que se tienen acceso en el servidor. La herramienta permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

1.5.4.4 My SQL

Es un sistema administrativo relacional de bases de datos (RDBMS[22]). Este tipo de bases de datos puede ejecutar desde acciones tan básicas, como insertar y borrar registros, actualizar información o hacer consultas simples, hasta realizar tareas tan complejas como la aplicación lo requiera. MySQL es un servidor multi-usuarios muy rápido y robusto de ejecución de instrucciones en paralelo, es decir, que múltiples usuarios, distribuidos a lo largo de una red local o Internet, podrán ejecutar distintas tareas sobre las bases de datos localizadas en un mismo servidor. El lenguaje que emplea es SQL[23], el cual es el estándar de consulta a bases de datos a nivel mundial. [8]

1.5.4.4.1 Características Principales [9]

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas

[22] Relational Database Management System (Sistema de Manejo Relacional de Base de Datos) [8]
[23] Structured Query Language (Lenguaje Estructural de Interrogantes) [8]

1.5.4.4.2 Ventajas [9]

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

1.5.4.4.3 Desventajas [9]

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS)

1.5.4.5 Comandos AT [14]

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal Modem. Se denominan así por la abreviación de ATtention.

Aunque la finalidad principal de los comandos AT es la comunicación con módems, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal.

Entre los comandos AT más usados se pueden encontrar los siguientes:

- Comandos generales:
  - AT+CGMI: Identificación del fabricante
  - AT+CGSN: Obtener número de serie
  - AT+CIMI: Obtener el IMSI.
  - AT+CPAS: Leer estado del modem

- Comandos de seguridad:
  - AT+CPIN: Introducir el PIN
  - AT+CPINC: Obtener el número de reintentos que quedan
  - AT+CPWD: Cambiar password

- Comandos para SMS:
  - AT+CPMS: Seleccionar lugar de almacenamiento de los SMS
  - AT+CMGF: Seleccionar formato de los mensajes SMS
  - AT+CMGR: Leer un mensaje SMS almacenado
  - AT+CMGL: Listar los mensajes almacenados
  - AT+CMGS: Enviar mensaje SMS
  - AT+CMGW: Almacenar mensaje en memoria
  - AT+CMSS: Enviar mensaje almacenado
  - AT+CSCA: Establecer el Centro de mensajes a usar
  - AT+ WMSC: Modificar el estado de un mensaje

1.5.4.6 NetBeans IDE [15]

NetBeans IDE[24] es un entorno de desarrollo integrado, es decir, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso.
Existe además un número importante de módulos para extender el NetBeans IDE.

[24] Integrated Development Environment (Ambiente de Desarrollo Integrado) [15]

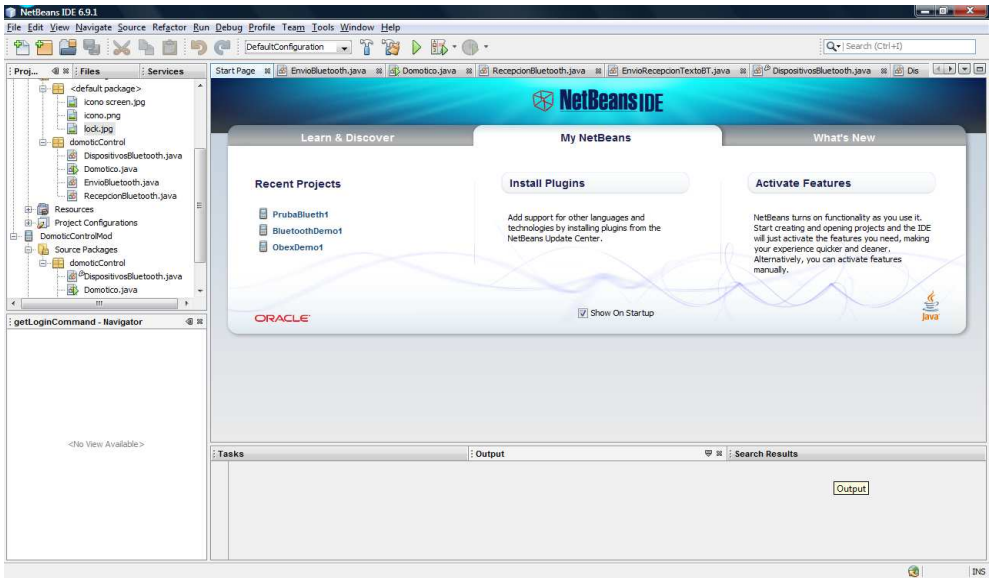


Figura 1-18.- Pantalla Inicial de NetBeans IDE

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

1.5.4.7 Java Micro Edition (JME)

1.5.4.7.1 Definición de Java [17]

Java es una plataforma virtual de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales (Multiplataforma)
La *plataforma Java* consta de las siguientes partes:

- El lenguaje de programación
- La máquina virtual de Java o JRE, que permite la portabilidad en ejecución.

- El API Java, una biblioteca estándar para el lenguaje.

Los programas en Java generalmente son compilados a un lenguaje intermedio llamado *bytecode*, que luego son interpretados por una máquina virtual (JVM). Esta última sirve como una plataforma de abstracción entre la máquina y el lenguaje permitiendo que se pueda "escribir el programa una vez, y correrlo en cualquier lado".

1.5.4.7.2 Campo de aplicación de Java ME [17]

El estándar Java define tres entornos diferentes pensando en las necesidades del usuario. Estos entornos pueden apreciarse en la siguiente figura:

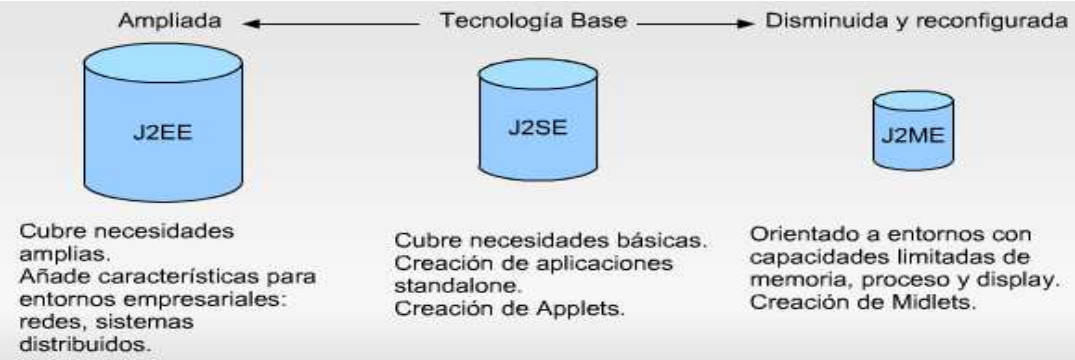


Figura 1-19.- Entornos de Java [19]

De estos tres entornos, la plataforma JME es una familia de especificaciones que definen varias versiones minimizadas de la plataforma Java; estas versiones minimizadas pueden ser usadas para programar en dispositivos electrónicos; desde teléfonos celulares, en PDAs, hasta en tarjetas inteligentes, etc. Estos dispositivos presentan en común que no disponen de abundante memoria ni mucha potencia en el procesamiento, ni tampoco necesitan de todo el soporte que brinda el JSE (Java Standard Edition), la plataforma estándar de Java usada en sistemas de escritorio y servidor. [16]

1.5.4.7.3 Midlet [18]

Midlet es un programa desarrollado en lenguaje de programación Java para dispositivos embebidos (se dedican a una sola actividad), más específicamente para la máquina virtual Java MicroEdition (Java ME). Generalmente son juegos y aplicaciones que corren en un teléfono móvil. Está desarrollada bajo la especificación MIDP (Mobile Information Device Profile).

1.5.4.7.4 Interfaz de usuario

La interfaz de usuario (User interface, UI) MIDP está compuesta de dos APIs (Interfaces de Programación de Aplicaciones): una de alto nivel y otra de bajo nivel.

- La primera (high-level API) se diseña para aplicaciones de negocio para las que interesa una portabilidad del cliente. Para ello, se sacrifica el control sobre la apariencia (look & feel) de los distintos elementos que la componen.
- El API de bajo nivel (low-level API), proporciona poca abstracción. Se ha diseñado para que las aplicaciones que necesiten un control fino de los elementos gráficos o acceso a dispositivos de entrada puedan hacerlo directamente. A cambio de todo esto, se sacrifica la posible portabilidad de este tipo de aplicaciones. Un ejemplo típico de estas aplicaciones son los juegos. Si se utiliza este API se podrá:

    - Tener un control absoluto de lo que se dibuja en la pantalla.
    - Controlar los eventos primitivos del tipo pulsación y liberación de teclas.
    - Acceder a teclas concretas y otros dispositivos de entrada.

El paquete javax.microedition.lcdui define las APIs que se pueden utilizar en la implementación de aplicaciones MIDP.

Las pantallas que implementan el API de alto nivel son subclases de Screen. Las clases Canvas y Graphics implementan el API de bajo nivel.

- Interfaces:

CommandListener: Interfaz para aplicaciones que han de recibir eventos de alto nivel que vienen de la implementación.
ItemStateListener: Interfaz que utilizan las aplicaciones que han de recibir eventos que indican cambios en el estado interno de los elementos interactivos dentro de un formulario.

- Clases:

Canvas: Clase base para la escritura de aplicaciones que han de manejar eventos de bajo nivel y lanzar invocaciones gráficas para dibujar en la pantalla del dispositivo.
ChoiceGroup: Grupo de elementos seleccionables para ser ubicado dentro de un formulario.
Command: Construcción para encapsular la información semántica de una acción.
Display: Representa el gestor de la pantalla y dispositivos de entrada del sistema.
Displayable: Un objeto que tiene la capacidad de ubicarse en el dispositivo.
Form: Representa un formulario. Subclase de Screen que contiene una mezcla arbitraria de elementos: imágenes, textos de sólo lectura, campos editables y no editables, campos de fecha y hora, contadores y grupos de selección.
List: Subclase de Screen que contiene una lista de elementos a elegir.
Screen: Superclase de todas las clases de interfaz de usuario de alto nivel.

1.5.4.8 Bascom AVR [20]

El BASCOM-AVR© es un compilador de BASIC para la familia AVR de ATMEL, desarrollado por la empresa Holandesa MCS Electronic.

1.5.4.8.1 Características

- BASIC estructurado con etiquetas.
- Programación estructurada con sentencias IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND, SELECT- CASE.
- Generación de código máquina nativo en lugar de código interpretado.
- Bit, Byte, Integer, Word, Long, y variables tipo String.
- Los programas compilados trabajan con todos los microprocesadores (no-MEGA) de AVR que tienen memoria interior. La edición profesional apoyará la serie de MEGA también. Puesto que los 1200 no tienen SRAM, no funcionará con los 1200.
- Las instrucciones y comandos son bastante similares a las del Visual Basic y QuickBASIC de Microsoft.
- Comandos específicos para el manejo de displays LCD, integrados I2C e integrados 1WIRE Chips, teclado de PC, teclado de matriz, recepción RC5, software UART, SPI, LCD Gráficos, envió de IR RC5 o código Sony.
- Soporta variables locales, uso de funciones, y librerías
- Emulador terminal integrado con opción de download.
- Simulador integrado por probar.
- Editor con subrayador de sentencias.
- Ayuda ON LINE en el editor.

1.5.4.8.2 Comandos e instrucciones

- De estructura y condicionales:

IF, THEN, ELSE, ELSEIF, END IF, DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, DOWNTO, STEP, EXIT FOR, GOTO/GOSUB, SELECT, CASE.

- De entrada/salida:

PRINT, INPUT, INKEY, PRINT, INPUTHEX, LCD, UPPERLINE, LOWERLINE,DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME,

LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, WAITKEY, INPUTBIN, PRINTBIN, OPEN, CLOSE, DEBOUNCE, SHIFTIN, SHIFTOUT, GETATKBD, SPC.

- Funciones numéricas:

AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD, LOG, EXP, SQR, SIN, COS, TAN, ATN, ATN2, ASIN, ACOS, FIX, ROUND, MOD, SGN, POWER, RAD2DEG, DEG2RAD, LOG10, TANH, SINH, COSH.

- 1Wire:

1WWRITE, 1WREAD, 1WRESET, 1WIRECOUNT, 1WSEARCHFIRST, 1WSEARCHNEXT.

- Gestión de interrupciones:

ON INT0/INT1/TIMER0/TIMER1/SERIAL, RETURN, ENABLE, DISABLE, COUNTERx, CAPTUREx, INTERRUPTS, CONFIG, START, LOAD.

- Variables:

DIM, BIT, BYTE, INTEGER, WORD, LONG, SINGLE, STRING, DEFBIT, DEFBYTE, DEFINT, DEFWORD.

- Directivas:

$INCLUDE, $BAUD and $CRYSTAL, $SERIALINPUT, $SERIALOUTPUT, $RAMSIZE, $RAMSTART, $DEFAULT XRAM, $ASM-$END ASM, $LCD, $EXTERNAL, $LIB.

- Cadenas:

STRING, SPACE, LEFT, RIGHT, MID, VAL, HEXVAL, LEN, STR, HEX, LTRIM, RTRIM, TRIM, LCASE, UCASE, FORMAT, FUSING, INSTR.

CAPÍTULO 2
DISEÑO DE HARDWARE E IMPLEMENTACIÓN

2.1 ESQUEMA GENERAL

Como se describió en el Capítulo 1, en el numeral 1.3 solución del problema, el hardware diseñado para el sistema SECURE SYS está conformado por un nodo maestro y varios nodos esclavos, comunicados mediante una red RS-485, dos nodos de humo y un sensor magnético.

2.2 CIRCUITOS INTEGRADOS COMUNES EN LOS NODOS

A continuación se describen las características de los circuitos integrados que se emplean en los circuitos eléctricos de todos los nodos esclavos del sistema SECURE SYS.

2.2.1 MAX487 / SN75176

El circuito integrado MAX487 es un transceptor de baja potencia para comunicación RS-485 o RS-422. Está diseñado para comunicaciones bidireccionales half-duplex o full-duplex en líneas de transmisión multipunto tipo bus. Es capaz de minimizar las interferencias electromagnéticas (EMI) y reducir las reflexiones causadas por una inadecuada terminación de los cables, permitiendo así la transmisión de datos sin error hasta 250kbps. [44]

Este dispositivo combina un transmisor y un receptor diferencial, los cuales operan con una fuente de alimentación de 5Vdc. Este receptor tiene una sensibilidad de +/−200mV. [44]

2.2.1.1 Control

La distribución de pines del circuito integrado MAX487 se puede apreciar en la siguiente figura:

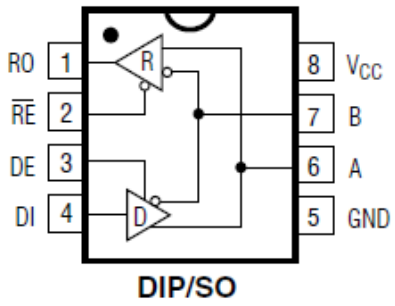


Figura 2-1.- Distribución de pines circuito integrado MAX487

De acuerdo a las tablas de función de este circuito integrado que se muestran adelante, la transmisión de datos se habilita al poner un uno lógico (H = 5Vdc) en el pin DE; mientras que, con cero lógico (L = 0Vdc) las salidas A y B estarán en alta impedancia (Z) y se deshabilitará la transmisión. Una vez habilitada la transmisión, si la entrada DI es un estado lógico TTL alto (5Vdc), la diferencia de potencial entre las líneas A y B se fija en $1.5Vdc \leq V_{AB} \leq 5Vdc$, haciendo la línea A más positiva que B. Al contrario, si la entrada DI es un estado lógico TTL bajo (0Vdc), la diferencia de potencial entre las líneas A y B se fija en $-5Vdc \leq V_{AB} \leq -1.5Vdc$, es decir, la línea B es más positiva que A.

La diferencia de potencial entre las líneas A y B no se mantendrá en el valor fijado por el transmisor, ya que debido a la longitud de los cables, su resistividad y otros agentes que causan interferencia, este voltaje variará hasta llegar a su destino. Pero gracias a las características de una comunicación de tipo diferencial no se perderán los datos enviados, ya que como se verá más adelante, el receptor puede determinar una salida en lógica TTL, si la diferencia de potencial entre las líneas A y B es diferente de $-0.2Vdc < V_{AB} < 0.2Vdc$.

Tabla 2-1.- Tabla de función para transmisión usando MAX487

INPUTS			OUTPUTS	
$\overline{RE}$	DE	DI	B	A
X	1	1	0	1
X	1	0	1	0
0	0	X	High-Z	High-Z
1	0	X	High-Z*	High-Z*

X = Don't care
High-Z = High impedance
* Shutdown mode for MAX481/MAX483/MAX487

En tanto, para habilitar la recepción de datos se debe fijar un cero lógico en el pin $\overline{RE}$. De esta manera, si el voltaje diferencial entre las líneas A y B (entrada) es mayor o igual a 0.2Vdc (200mVdc), es decir, el voltaje en la línea A es mayor a B, la salida (RO) se pondrá en estado lógico TTL alto (5Vdc). Mientras que, si este voltaje es menor o igual −0.2Vdc, es decir, el voltaje en la línea B es mayor a A, la salida (RO) se pondrá en estado lógico TTL bajo (0Vdc). Así, de acuerdo a la diferencia de potencial entre las líneas de entrada A y B, el receptor interpreta y envía a través de la salida RO estados lógicos en niveles TTL (0 ó 5 Vdc) a un dispositivo final.

Tabla 2-2.- Tabla de función para recepción usando MAX487

INPUTS			OUTPUT
$\overline{RE}$	DE	A-B	RO
0	0	≥ +0.2V	1
0	0	≤ -0.2V	0
0	0	Inputs open	1
1	0	X	High-Z*

X = Don't care
High-Z = High impedance
* Shutdown mode for MAX481/MAX483/MAX487

2.2.1.2 Reemplazo [45]

Un reemplazo del circuito integrado MAX487 y que normalmente está disponible en el mercado es el integrado SN75176. Este último elemento es de igual forma un transceptor para comunicación bidireccional tipo bus y tiene las mismas características de sensibilidad, alimentación y control que el circuito integrado MAX487.

Como se observa en la figura 2-1 y 2-2, estos dos circuitos integrados son compatibles pin a pin.

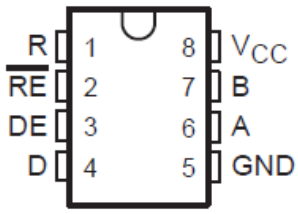


Figura 2-2.- Distribución de pines del circuito integrado SN75176

La habilitación de la transmisión de datos y su funcionamiento es igual que con el circuito integrado MAX487. En este caso, de acuerdo a la siguiente tabla de función, la entrada está denominada como D, la cual corresponde a la entrada DI del elemento MAX487.

Tabla 2-3.- Tabla de función para transmisión usando CI SN75176

INPUT D	ENABLE DE	OUTPUTS	
		A	B
H	H	H	L
L	H	L	H
X	L	Z	Z

Similarmente, la recepción de datos funciona tal como se explicó en el control del dispositivo MAX487. En la tabla siguiente, la salida R corresponde a la salida RO del elemento MAX487.

Tabla 2-4.- Tabla de función para recepción usando CI SN75176

DIFFERENTIAL INPUTS A – B	ENABLE $\overline{RE}$	OUTPUT R
$V_{ID} \geq 0.2$ V	L	H
-0.2 V $< V_{ID} < 0.2$ V	L	?
$V_{ID} \leq -0.2$ V	L	L
X	H	Z
Open	L	?

H = high level, L = low level, ? = indeterminate,
X = irrelevant, Z = high impedance (off)

2.2.1.3 Configuración de la red RS-485

El circuito de una red RS485 empleando el circuito integrado MAX487 o SN75176 se muestra en la siguiente figura:

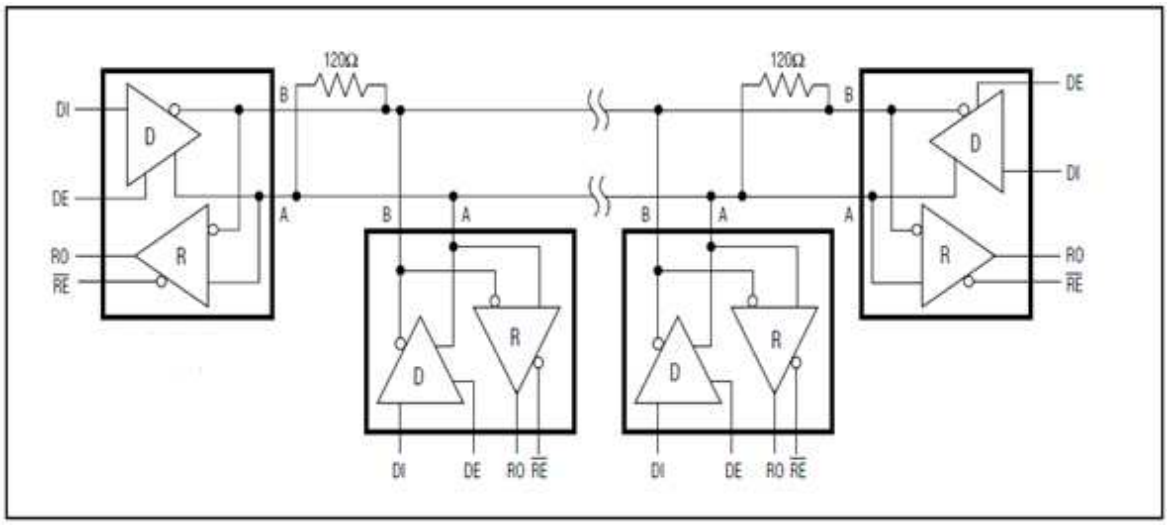


Figura 2-3.- Red RS-485 half-duplex con MAX487 [44]

El total de corriente utilizada por un enlace RS-485 puede variar debido a la impedancia de los componentes, incluyendo los transmisores, receptores, cables y la terminación de los componentes. Una baja impedancia a la salida del transmisor y una baja impedancia en los cables, facilitan los cambios de nivel y asegura que el receptor vea la señal, no importa cuán larga sea la línea de transmisión. En tanto, una alta impedancia en el receptor decrementa la corriente en el enlace e incrementa la vida de la fuentes de voltaje. [46]

La terminación de los componentes tiene un gran efecto sobre la corriente en el enlace. Los fabricantes suelen recomendar la utilización de una resistencia de 100Ω o 120Ω entre las líneas A y B en cada extremo de las mismas, como se muestra en la figura.

2.2.2 REGULADORES [47]

En cada fuente de alimentación de los distintos nodos del sistema SECURE SYS se emplean dos reguladores de voltaje: LM7805 Y LM7809. A continuación se describen sus principales características:

2.2.2.1 LM7805

Voltaje de entrada máximo = 35Vdc
Voltaje de salida = 5Vdc
Corriente de salida máxima = 1A

2.2.2.2 LM7809

Voltaje de entrada máximo = 35Vdc
Voltaje de salida = 9Vdc
Corriente de salida máxima = 1A

2.3 NODO MAESTRO

El nodo maestro es el encargado de las siguientes funciones:

- Solicitar continuamente información acerca de los valores de temperatura, nivel de corriente consumida y presencia de agua a los respectivos nodos esclavos, a través de la red RS-485.

- Monitorear la existencia de humo.

- Monitorear el estado del sensor magnético para detección de apertura de puerta.

- Comandar la apertura de puerta en caso de que así lo haya solicitado la interfaz estacionaria o móvil.

- Procesar la información adquirida y darle el formato adecuado para que la interfaz estacionaria interprete correctamente los datos.

- Enviar los datos solicitados por la interfaz estacionaria, utilizando el protocolo de comunicación RS-232.

Para cumplir con todas estas funciones se diseñó y construyó la circuitería del nodo maestro basándose en el siguiente esquema:

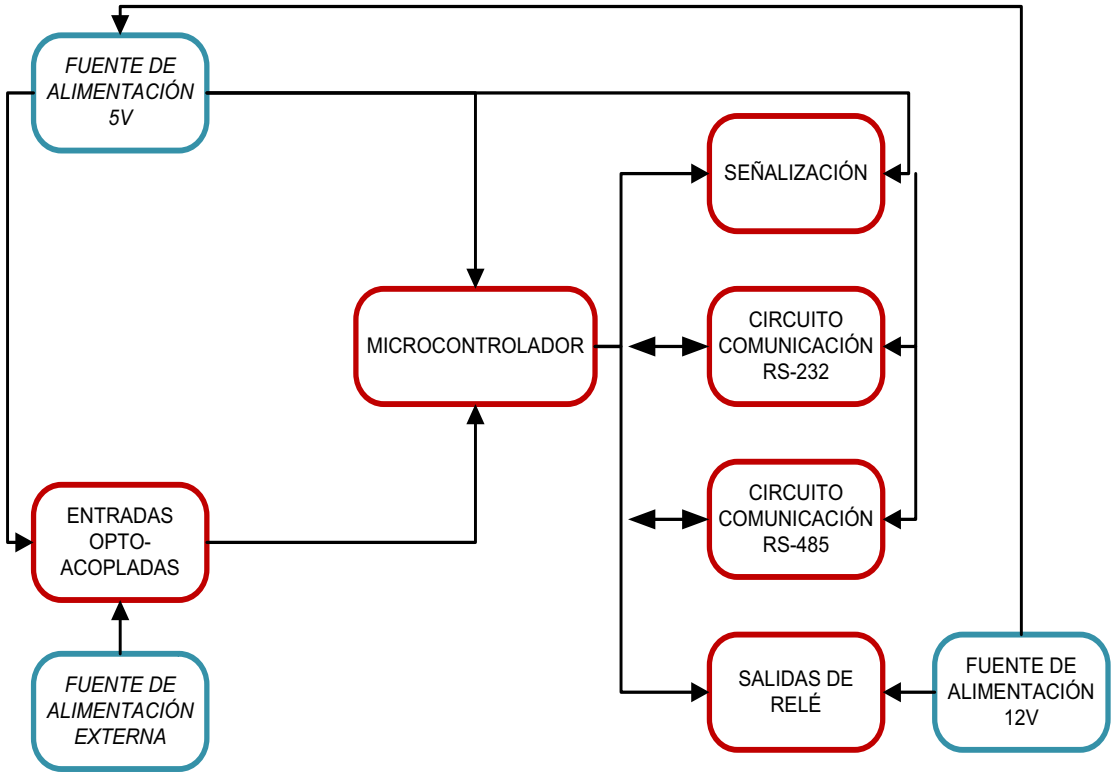
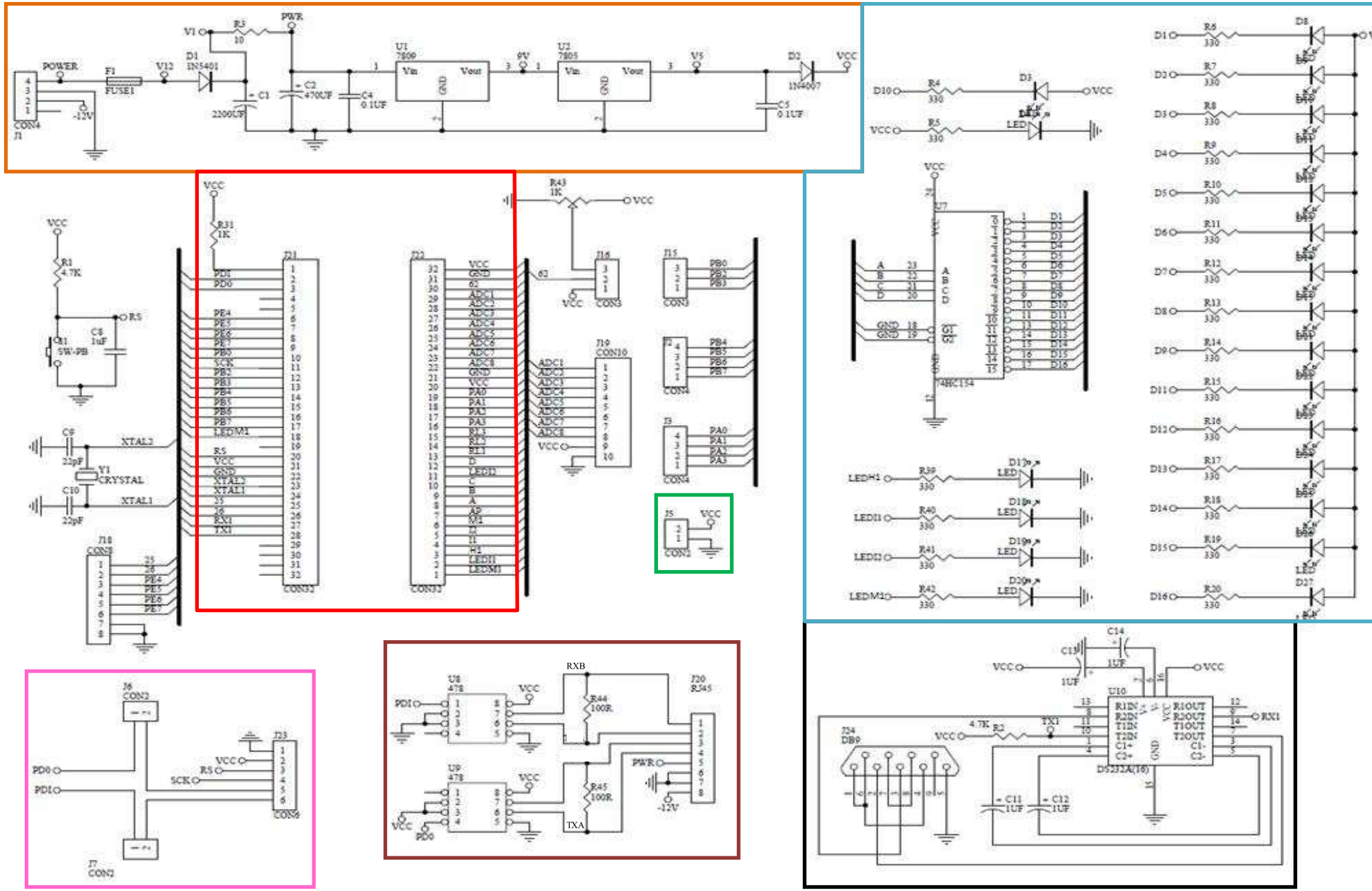


Figura 2-4.- Esquema general del nodo maestro

En la Figura 2.5, guiándose por los respectivos colores, se puede visualizar el hardware empleado para los siguientes propósitos:

- Salida de 5V
- Pines para programación del microcontrolador
- Comunicación con la interfaz estacionaria (PC)
- Comunicación con la red RS-485
- Fuente de alimentación
- Entradas opto-acopladas
- Microcontrolador
- Señalización
- Salidas de relé
- Otras fuentes de alimentación



POWER
F1
FUSE1
V12
D1
1N5401
V1
R3
10
PWR
C1
2200UF
C2
470UF
C4
0.1UF
U1
7809
Vin
Vout
GND
9V
U2
7805
V5
D2
1N4007
VCC
C5
0.1UF
CON4
J1
-12V
R1
4.7K
RS
C8
1uF
SW-PB
C9
22pF
XTAL2
Y1
CRYSTAL
C10
22pF
XTAL1
J18
CON8
R31
1K
J21
CON32
J22
CON32
R43
1K
J16
CON3
J19
CON10
J15
CON3
J2
CON4
J3
CON4
J5
CON2
D10
R4
330
D3
R5
330
LED
U7
74HC154
LEDH1
R39
330
LEDI1
R40
330
LEDI2
R41
330
LEDM1
R42
330
D1
D2
D3
D4
D5
D6
D7
D8
D9
D11
D12
D13
D14
D15
D16
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
330
D8
D27
J6
CON2
J7
CON2
J23
CON6
PD0
PD1
SCK
RS
VCC
U8
478
U9
478
RXB
R44
100R
R45
100R
TXA
J20
RJ45
PWR
-12V
C13
1UF
C14
1UF
U10
DS232A(16)
R1IN
R2IN
T1IN
T2IN
C1+
C2+
R1OUT
R2OUT
T1OUT
T2OUT
C1-
C2-
RX1
TX1
R2
4.7K
J24
DB9
C11
1UF
C12
1UF

A continuación se detalla el hardware del nodo maestro, explicando cada una de sus funciones por separado. En el Anexo A se puede encontrar el circuito eléctrico completo del nodo maestro y el ruteado realizdo previo a laconstrucción su placa.

2.3.1 FUENTE DE ALIMENTACIÓN

El circuito de la Figura 2-6 se emplea para regular el voltaje de +12Vdc (Power) de la fuente de entrada (Figura 2-7), fijando a la salida (Vcc) un voltaje de +5Vdc para alimentar el microcontrolador, el multiplexor 74HC154 y los leds para señalización, los circuitos integrados MAX487 para la comunicación RS-485 y el circuito integrado MAX232 para la comunicación RS-232.

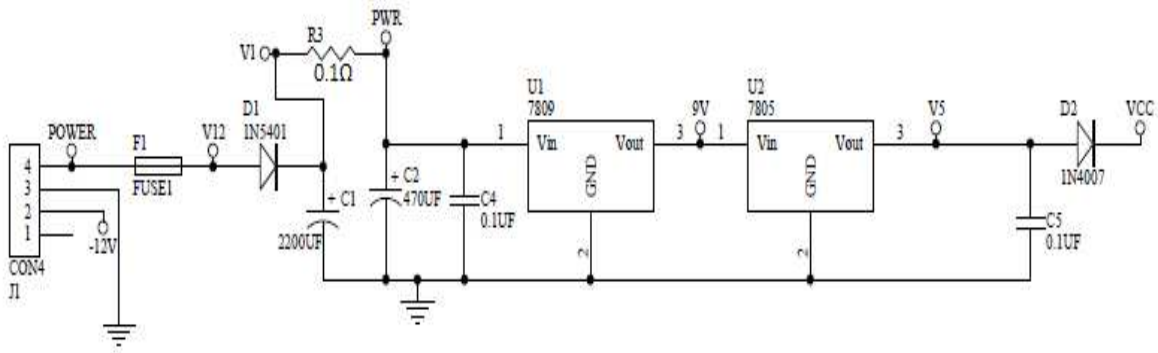
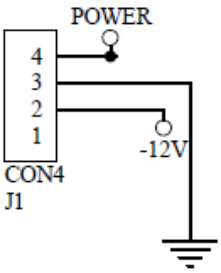


Figura 2-6.- Circuito de regulación de voltaje a 5Vdc

El voltaje de 12Vdc (POWER) de la fuente de alimentación que se visualiza en la siguiente figura, energiza el nodo maestro y los circuitos para adquisición de datos y comunicación con la red RS-485 de los nodos esclavos, como se describirá posteriormente. El voltaje de −12Vdc es empleado para el circuito de acondicionamiento del nodo de corriente.



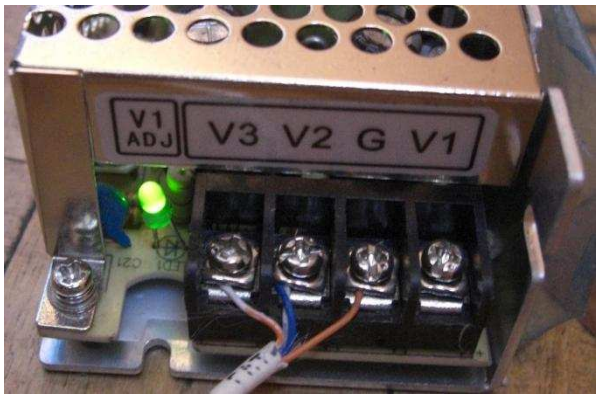


Figura 2-7.- Fuente de alimentación +/- 12V

Esta fuente se debe energizar con: 100 – 240V AC / 1,2A / 50-60Hz, y sus salidas tienen las siguientes características de voltaje y corriente:

- V1 = +5V / 3A
- V2 = +12V / 1A
- V3 = -12V / 1A

2.3.1.1 Elementos implementados

- Reguladores de voltaje

En este circuito se implementaron dos reguladores de voltaje: un LM7805 y un LM7809. Se implementó estos dos reguladores para que el tiempo de vida de los mismos no disminuya a causa de calentamiento excesivo, ya que se requiere funcionen continuamente.

De acuerdo a la figura del circuito de alimentación, se observa que la señal filtrada de 12Vdc (PWR) es el voltaje de entrada para el regulador 7809, cuya salida de 9Vdc ingresa al regulador LM7805 obteniéndose a su salida un voltaje de 5V.

- Fusible

Se empleó este elemento como medio de protección para el sistema SECURE SYS, ya que de acuerdo a las mediciones de corriente que se realizaron en este sistema, se determinó que consume alrededor de 2.5 A. Por esta razón se colocó un fusible de 3 A en la línea de energía de 12Vdc, permitiendo así una posible expansión del sistema.

El nodo maestro consume alrededor de 0,6 A, lo cual está dentro del límite de corriente que pueden soportar los reguladores de voltaje LM7805 y LM7809. La corriente restante es consumida por los nodos esclavos.

- Diodos

Se emplearon diodos rectificadores para evitar la polarización incorrecta de los elementos del nodo maestro, ya que por las características propias de los diodos, estos conducirán cuando estén correctamente polarizados.
Se empleó el diodo 1N5401 antes de los capacitores electrolíticos para evitar que estos se polaricen incorrectamente y se ocasionen daños. La corriente máxima que soporta este diodo es de 3 A, lo que va de acuerdo con lo permitido por el fusible.

También se utilizó un diodo de señal 1N4007 a la salida de 5Vdc como protección en caso de que por alguna falla circule corriente en sentido contrario y pueda causar más daños.

- Resistencias

Se utilizó una resistencia de 0.1Ω para que el tiempo de carga del capacitor de 470uF sea pequeño y la caída de voltaje en esta resistencia sea mínima. Así la constante de tiempo RC es de 0.1 x 470u = 47us. Por tanto el capacitor llegará al 63% de su carga total en dicho tiempo.

- Capacitores

El capacitor electrolítico de 2200uF, la resistencia de 0.1Ω y el capacitor electrolítico de 470uF se emplean para suavizar los cambios bruscos en el voltaje de entrada, en caso de que se susciten, evitando posibles daños y para reducir el rizado antes de entrar a la etapa de regulación.

Se emplean dos capacitores cerámicos de 0.1uF para filtrar señales de alta frecuencia, como ruido, que pueden causar interferencia.

2.3.2 MICROCONTROLADOR

El microcontrolador del nodo maestro se encarga de llevar a cabo todas las funciones antes descritas. A continuación se describen todos los aspectos que se tomaron en cuenta para su selección y sus principales características.

2.3.2.1 Selección del microcontrolador

Para cumplir con las funciones propuestas para el nodo maestro se requiere:

- 2 puertos seriales USART[25]: Un puerto para comunicación con la interfaz estacionaria (PC) y el segundo puerto para comunicación con los nodos esclavos mediante la red RS-485.
- 4 pines para señalizar que se ha logrado una comunicación exitosa con cada esclavo.
- 3 pines para señalización del estado (activado o desactivado) de los sensores de humo y magnético.
- 1 pin de señalización de presencia de alimentación de voltaje.
- 1 pin para comandar la apertura de puerta.
- 2 pines para monitorear los dos sensores de humo.

[25] Universal Asynchronous Receiver Transmitter (Receptor/Transmisor Asíncrono Universal) [50]

- 1 pin para monitoreo del sensor magnético.

Los microcontroladores disponibles en el mercado que permiten emplear dos puertos seriales USART son el Atmega164 y el Atmega64. A continuación se describen sus principales características:

Tabla 2-5.- Comparación Atmega164 vs. Atmega64

	N° Pines	Pines I/O	Interrupciones Externas	Canales ADC[26]	Puertos USART
Atmega164	44	32	32	8	2
Atmega64	64	53	8	8	2

De acuerdo a esta tabla comparativa, el microcontrolador Atmega164 es la mejor alternativa, ya que se podrá cumplir con los requerimientos establecidos y permitir una posible ampliación sin desaprovechar demasiado su capacidad.

2.3.2.2 Compatibilidad

Para dar versatilidad al nodo maestro se diseñó y construyó una placa que hace compatible el microcontrolador Atmega164 con el Atmega64. Así se podrá utilizar indistintamente ambos microcontroladores de acuerdo a las necesidades que se requiera suplir. En caso de necesitar mayor capacidad de ampliación para el sistema SECURE SYS se podrá emplear el microcontrolador Atmega64 ya que dispone de mayor número de puertos y mayor cantidad de memoria.

A continuación se muestra el diseño de la placa de adaptación que permite la compatibilidad entre los dos microcontroladores antes mencionados.

[26] Analog Digital Converter (Conversor Análogo Digital)

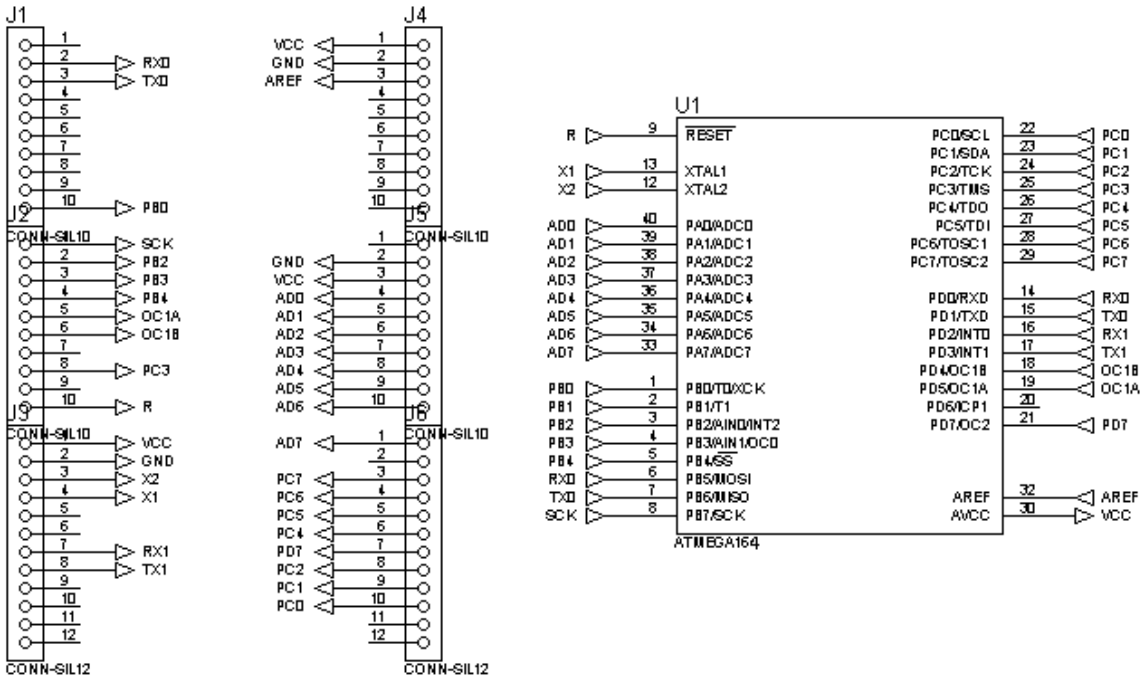


Figura 2-8.- Circuito de adaptación para el microcontrolador Atmega164

Como se observa, no todos los pines de los zócalos J1 y J4, que representan los pines del microcontrolador Atmega64, están conectados a un pin del microcontrolador Atmega164 ya que no tienen la misma cantidad de pines. Por esta razón, se hicieron las conexiones de tal manera de cubrir con todas las funciones que necesita realizar el nodo maestro y dejar pines para posibles expansiones.

En la siguiente tabla se muestra la función que tiene cada pin del microcontrolador Atmega164:

Tabla 2-6.- Función de los pines del microcontrolador Atmega164

FUNCIÓN	PINES ATMEGA164	
	Número	Nombre
Comunicación RS232	14, 15	RX0, TX0
Comunicación RS485	16, 17	RX1, TX1

Señalización de comunicación con esclavos	27, 28, 29, 33	PC5, PC6, PC7, AD7
Señalización sensor magnético	25	PC3
Detección de apertura de puerta	21	PD7
Detección de alarma de humo	23, 24	PC1, PC2
Apertura de puerta	26	PC4
Salidas de relé	36, 35, 34	AD4, AD5, AD6
Expansión	40, 39, 38, 37, 22, 1, 3, 4, 5	AD0, AD1, AD2, AD3, PC0, PB0, PB2, PB3, PB4

2.3.2.3 Conexiones

A continuación se muestran todas las conexiones realizadas a los pines de los zócalos en donde se puede colocar el microcontrolador Atmega64 o la placa de adaptación con el microcontroldor Atmega164:

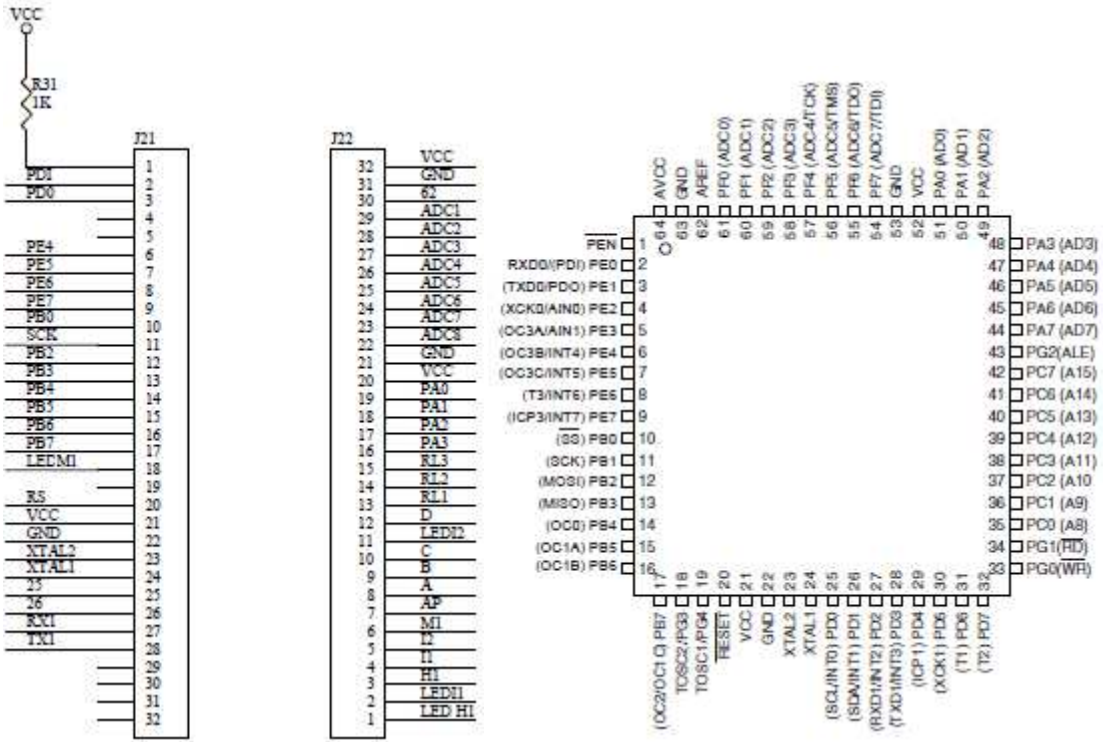


Figura 2-9.- Conexiones al microcontrolador

Los pines del zócalo J21 corresponden a los pines del 1 al 32 del microcontrolador Atmega64 o de la placa de adaptación. En tanto, los pines del zócalo J22 corresponden a los pines del 33 al 64 de los mismos.

Las conexiones se realizaron de tal manera de aprovechar, lo más posible, las funcionalidades del microcontrolador Atmega64, el cual posee 64 pines. En el caso de usar la placa de adaptación para el microcontrolador Atmega164, debido a que este posee únicamente 44 pines, se omiten algunas conexiones a terminales de entrada o salida como se vio anteriormente, que al necesitar una expansión del sistema SECURE SYS podrán ser usados con el microcontrolador Atmega64, como se describirá posteriormente.

Se empleó un cristal de oscilación externo, ya que es más confiable que utilizar el cristal interno del microcontrolador. El valor del cristal implementado es de 8MHz.
En la tabla siguiente se puede encontrar la función que tiene a cargo cada pin en caso de emplear el microcontrolador Atmega64. Además, se puede observar qué funciones están disponibles con el microcontrolador Atmega164.

Tabla 2-7.- Función del los pines de microcontrolador del nodo maestro

PINES	FUNCIÓN	ATMEGA64	ATMEGA164
2, 3	Comunicación RS-485 y programación	√	√
6, 7, 8, 9	Expansión interrupciones externas	√	–
10	Expansión	√	√
11, 20	Programación	√	√
12, 13, 14	Expansión	√	√
15, 16, 17	Expansión	√	–
18	Señalización sensor magnético	√	–
21, 22, 52, 53, 63, 64	Alimentación	√	√
23, 24	Cristal externo	√	√
27, 28	Comunicación RS-232	√	√
33	Expansión	√	–
34, 43	Señalización sensores	√	–

	de humo		
35	Expansión entrada optoacoplada	√	–
36, 37	Detección de alarma de humo	√	√
38	Detección de apertura de puerta	√	√
39	Comandar apertura de puerta	√	√
40, 41, 42, 44	Señalización de comunicación con esclavos	√	√
45, 46, 47	Salidas de relé	√	√
48, 49, 50, 51	Expansión	√	√
54, 55, 56, 57, 58, 59, 60, 61	Expansión conversor análogo digital	√	–
62	Referencia de voltaje para los conversores análogo digital	√	√

2.3.2.4 Pines para programación

La placa del nodo maestro dispone de un zócalo para programación del microcontrolador (Figura 2-11). Los elementos J6 Y J7 son jumpers que se utilizan para determinar si se va a programar el microcontrolador o se va a emplear el puerto serial, ya que los pines PDO Y PDI del microcontrolador Atmega64 son Tx0 (transmisión) y Rx0 (recepción) respectivamente y a la vez son pines de programación.

Para programar el microcontrolador se deben cortocircuitar los pines de los jumpers J6 y J7. Mientras que, para usar la comunicación serial se debe retirar el contacto entre dichos pines.

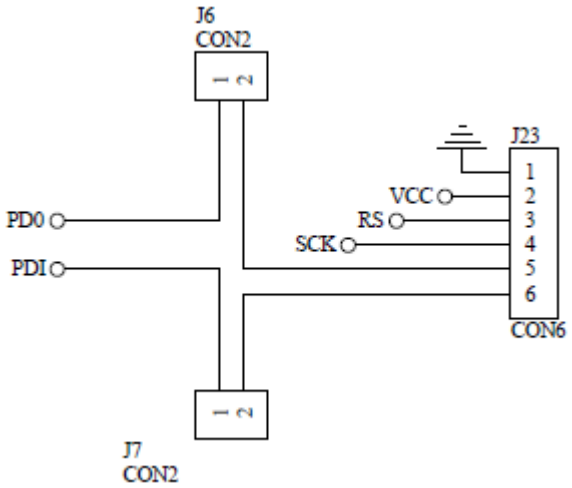


Figura 2-11.- Conexión del zócalo de programación

2.3.3 COMUNICACIÓN CON LA INTERFAZ ESTACIONARIA (PC)

Se emplea comunicación serial RS-232 para comunicar el nodo maestro con la interfaz estacionaria (PC). Para llevar a cabo esta tarea, se utilizó el circuito integrado MAX 232 y un conector DB-9 en el siguiente circuito eléctrico:

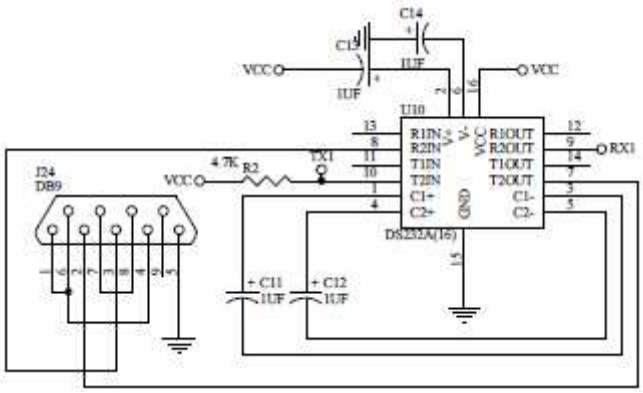
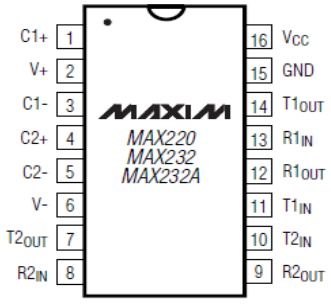


Figura 2-12.- Circuito de comunicación con la interfaz estacionaria

Las conexiones al circuito integrado MAX232 se realizaron de acuerdo a las especificaciones del fabricante, como se observa en la siguiente figura:



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

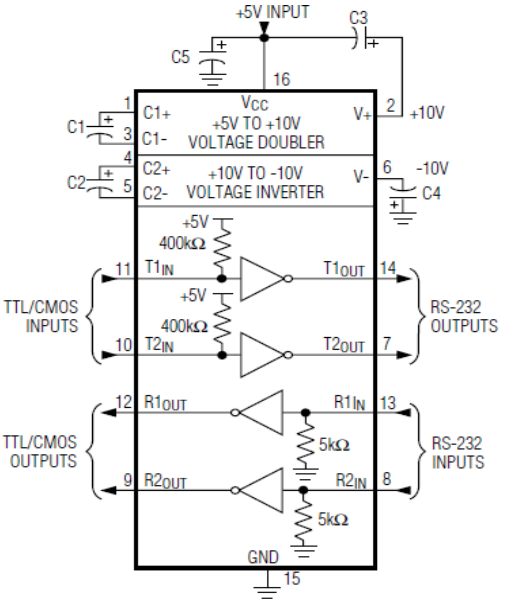


Figura 2-13.- Configuración MAX232

Como se muestra, los capacitores C1, C2, C4 y C5 son de 1uF para el caso del circuito integrado MAX232.

De acuerdo al circuito de comunicación implementado, el pin 7 (salida RS-232) del elemento MAX232 se conecta al pin 2 del conector DB-9, el cual a su vez se conecta a la línea de recepción del computador. En tanto, el pin 8 (entrada RS-232) del circuito integrado MAX232 se conecta al pin 3 del conector DB-9, el mismo que se enlaza a la línea de transmisión del computador. Además, el pin 5 del conector DB-9 está conectado a tierra. Por consiguiente, debido estas conexiones, el pin 10 del elemento MAX232 se conecta a la línea de transmisión del microcontrolador, y el pin 9 se conecta a la línea de recepción del mismo.

2.3.4 COMUNICACIÓN CON LA RED RS-485

Para comunicar el nodo maestro con los nodos esclavos vía RS-485, se empleó la configuración configuración full-duplex que propone el fabricante del circuito

integrado SN75176, cuya distribución de pines y funcionamiento se describió previamente en el apartado 2.2.

Como se aprecia en la siguiente figura, se empleó una resistencia de 100Ω en los terminales de las líneas de comunicación, tal como lo recomiendan los fabricantes. Además, se utiliza el circuito integrado U9 para transmisión y U8 para recepción, logrando así una comunicación full-duplex.

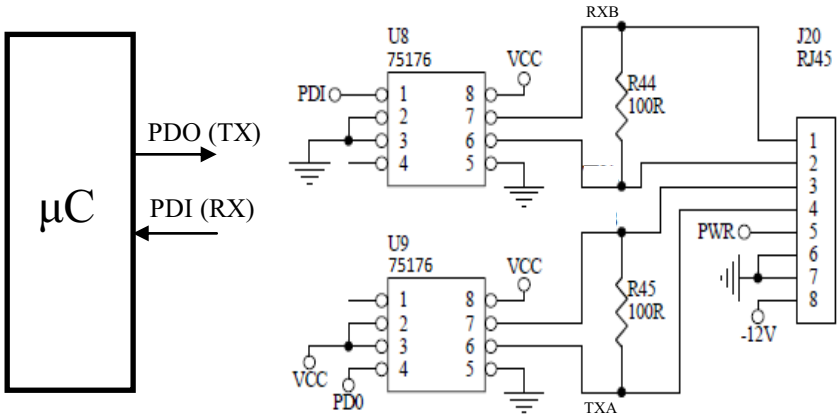


Figura 2-14.- Circuito para la comunicación RS-485

Para habilitar la recepción en U8 se fija 0Vdc en sus pines 2 y 3. Mientras que, para habilitar la transmisión en U9 se fija 5Vdc en sus pines 2 y 3. El pin 1 de U8 va conectado al pin de recepción, y el pin 1 de U9 se conecta al pin de transmisión del puerto USART del microcontrolador dedicado a la comunicación RS-485.

2.3.5 ENTRADAS OPTOACOPLADAS

En el nodo maestro se dispusieron cuatro entradas opto-acopladas, de las cuales, dos se emplean para los dos sensores de humo y una para el sensor magnético, quedando una en caso de que se requiera una ampliación del sistema SECURE SYS.

Las conexiones para cada entrada opto-acoplada se realizaron de acuerdo a las siguientes figuras:

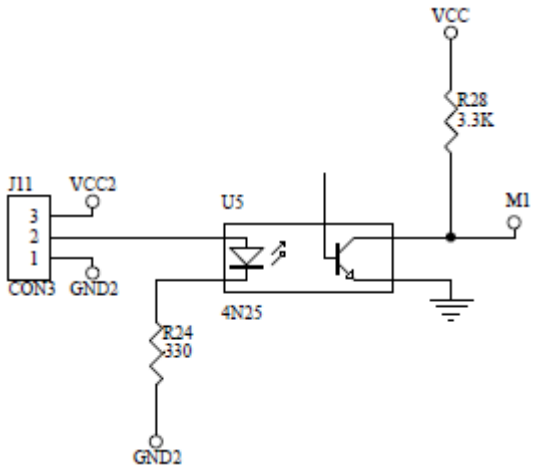


Figura 2-14.- Entrada opto-acoplada para el sensor magnético

Como se puede apreciar, se emplea el circuito integrado 4N25, el cual consiste en un transistor opto-aislado. Este elemento es capaz de modular una señal luminosa, partiendo de una señal eléctrica, para luego convertirla otra vez en señal eléctrica. De esta forma, se establece un aislamiento galvánico entre los circuitos de entrada y salida [48]. Este aislamiento es muy aconsejable para evitar lazos de corriente en la línea de tierra, ya que debido a la longitud de los cables puede generarse una diferencia de potencial entre los distintos puntos de tierra.

Al pin 2 del zócalo J11 se conecta la salida de sensor magnético y de acuerdo a esta señal el transistor del opto-acoplador se saturará o no. El terminal M1 está normalmente en estado lógico TTL alto y al momento de saturarse el transistor, este terminal se fija en 0Vdc, lo cual indica que la puerta de las instalaciones está abierta.

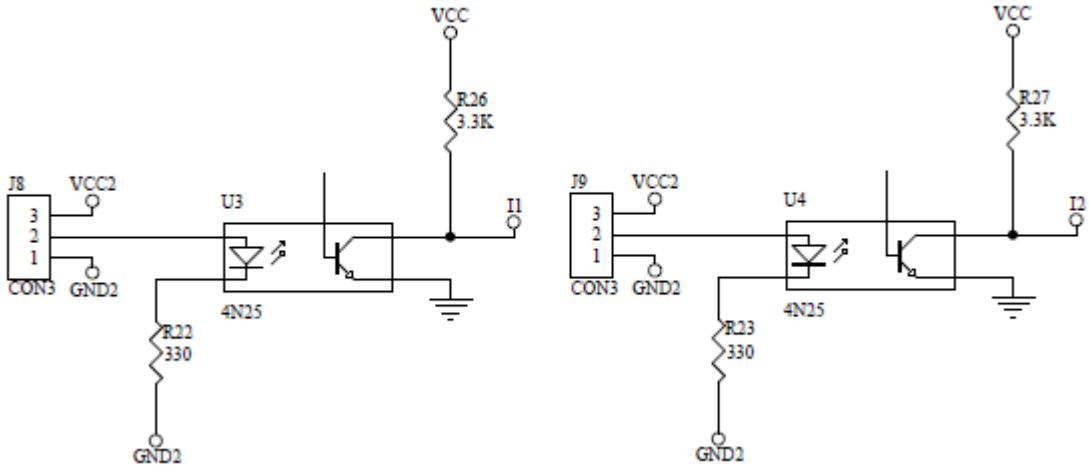


Figura 2-15.- Entradas opto-acopladas para los sensores de humo

Basándose en la figura de arriba, en el pin 2 de los zócalos J8 y J9 se conecta la salida de los circuitos de acondicionamiento de los sensores de humo. Estas señales determinan si hay presencia o no de humo. De igual forma que en la entrada opto-acoplada para el sensor magnético, los terminales I1 e I2 están normalmente en estado lógico TTL alto. Cuando los sensores detectan presencia de humo, se saturan los transistores y el voltaje en dichos terminales se hace 0Vdc.

La salida del sensor magnético y las salidas de los circuitos de acondicionamiento para los sensores de humo son 0Vdc o 5Vdc como se verá posteriormente.

En estos circuitos opto-acoplados se emplea una resistencia de 330Ω, la cual limita la corriente que circula por el led del circuito integrado 4N25 a 15mA, como se en la solución de la siguiente fórmula, cuyo máximo valor puede ser hasta 60mA de acuerdo al fabricante.

$$I_{LED} = \frac{V}{R}$$

$$I_{LED} = \frac{5Vdc}{330\Omega} = 15mA$$

Además, se emplea una resistencia de 3.3kΩ, la misma que limita la corriente que puede circular por los pines (I1, I2 y M1) del microcontrolador a 1.5mA, ya que las entradas del los microcontroladores Atmega64 y Atmega164 soportan hasta 40mA.

En los pines Vcc2 y Gnd2 va conectado una fuente externa, que en este caso se puede tomar de los mismos sensores de humo, para cumplir con el objetivo del opto-acoplamiento, que es la separación de tierras entre los circuitos que están antes y después del opto-acoplador.

2.3.6 SALIDAS DE RELÉ

En el nodo maestro se dispusieron cuatro salidas de relé. Una de estas fue empleada para accionar la apertura de la puerta de entrada a las instalaciones y las otras tres servirán como medio de ampliación para el sistema SECURE SYS.

A continuación se muestra el circuito eléctrico empleado para abrir la puerta una vez que se haya comandado hacerlo.

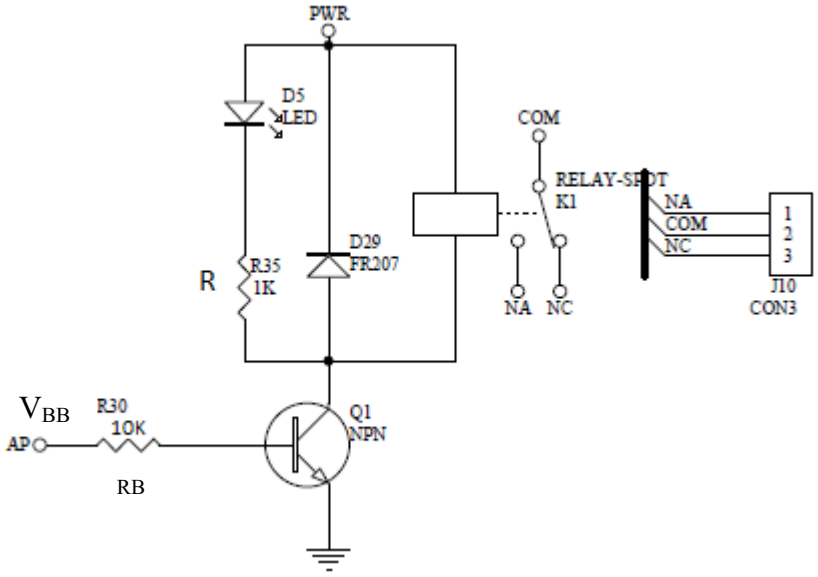


Figura 2-16.- Salida de relé para apertura de puerta

El terminal AP corresponde a un pin del microcontrolador, como se vio anteriormente, por medio del cual se controla el accionamiento de la cerradura eléctrica de la puerta.

El elemento Q1 es el transistor NPN 2N3904, cuyas características principales, necesarias para el dimensionamiento de los elementos del circuito expuesto, son:

I_{MAX} Colector = 200mA (Corriente máxima en el colector del transistor)
I_{MAX} Base = 2mA (Corriente máxima en la base del transistor)
$V_{CE\ <sat>}$ = 200mVdc (Voltaje colector-emisor al saturarse el transistor)
$100 \leq h_{FE} \leq 300$, para Ic = 10mA. (Ganancia de corriente)

De acuerdo al circuito de la figura 2-16 se tiene que:

$$Ic_{<sat>} = \frac{PWR - V_{CE<sat>} - V_{LED}}{R} \quad (1)$$

Donde:

$Ic_{<sat>}$ = Corriente de colector de saturación.
$V_{CE<sat>}$ = Voltaje colector emisor de saturación.
V_{LED} = Voltaje en el led al momento de saturarse el transistor.
R = Resistencia en serie al led.

El voltaje que cae en el led cuando el transistor está saturado es aproximadamente de 1,6V y el valor de la resistencia en serie debe ser tal forma, que limite la corriente por el led a 10mA aconsejablemente, ya que la corriente máxima que puede circular sin que el mismo sufra daño o disminuya su vida útil es de 20mA.

Por tanto, despejando de la ecuación (1), el valor de la resistencia R debe ser:

$$R \approx \frac{12Vdc - 200mVdc - 1{,}6\,Vdc}{10mA}$$

$$R \approx 1000\Omega$$

Para dimensionar la resistencia de la base del transistor (RB) se empleó las siguientes fórmulas:

$$I_{B<sat>} = \frac{I_{C<sat>}}{\beta} \quad (2)$$

$$V_{BB} = RB * I_{B<sat>} + V_{BE<sat>} \quad (3)$$

Donde:

$I_{B<sat>}$ = Corriente de base de saturación
$\beta = h_{FE}$ = Ganancia de corriente en emisor común
$V_{BE<sat>}$ = Voltaje base emisor en saturación
Evaluando en las peores condiciones, es decir, cuando β = 100, se tiene:

Reemplazando valores en la ecuación (2):

$$I_{B<sat>} = \frac{10mA}{100} = 0{,}1mA$$

Este valor calculado para de $I_{B<sat>}$ es la mínima corriente que debe circular por la base para que el transistor se sature.

Reemplazando valores en la ecuación (3):

$$5Vdc = RB * 0{,}1mA + 850mVdc$$

$$RB = \frac{5Vdc - 850mVdc}{0{,}1mA}$$

$$RB = 41{,}5\ K\Omega$$

Por tanto para que $I_{B<sat>}$ sea mayor que 0,1mA, RB debe ser menor a 41,5kΩ. Por esta razón se empleó una resistencia RB de 10kΩ, fijando así una corriente de base de 0,4mA lo cual está dentro del límite máximo antes especificado.

En paralelo a la bobina del relé se colocó un diodo 1N4007 como protección, ya que al momento de la desconexión de los contactos, se induce una corriente inversa que puede dañar el transistor. Los relés empleados se accionan con 12Vdc.

En el zócalo J10, el pin 1 se conecta al terminal de alimentación para la cerradura eléctrica y en el pin 2 se conecta la fuente de 22Vac, que es con lo que se acciona la misma. Así, al activarse el relé, estos dos pines hacen contacto, abriendo la puerta.

La siguiente figura muestra una de las tres salidas de relé que se dejaron para posibles expansiones:

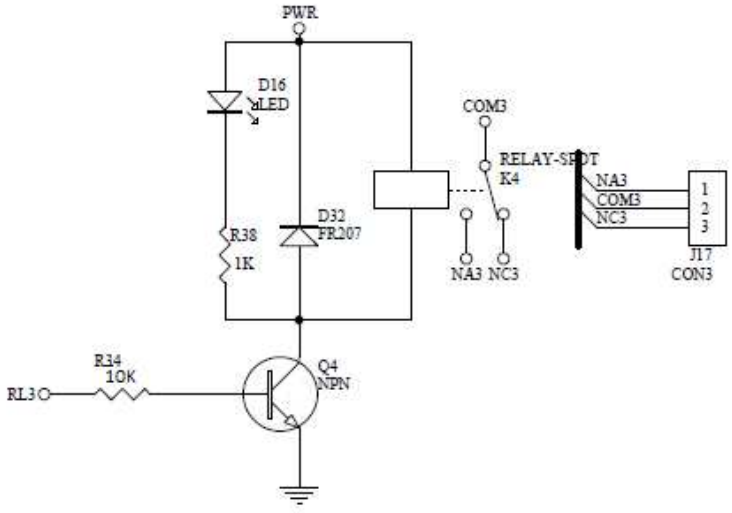


Figura 2-17.- Salida de relé para expansión

2.3.7 SEÑALIZACIÓN

La señalización es muy importante y útil en un circuito eléctrico, ya que permite conocer con facilidad que ciertas acciones se están llevando a cabo. Por esta razón, se decidió emplear el siguiente circuito de señalización para comprobar que la comunicación entre los nodos esclavos y en el nodo maestro es exitosa.

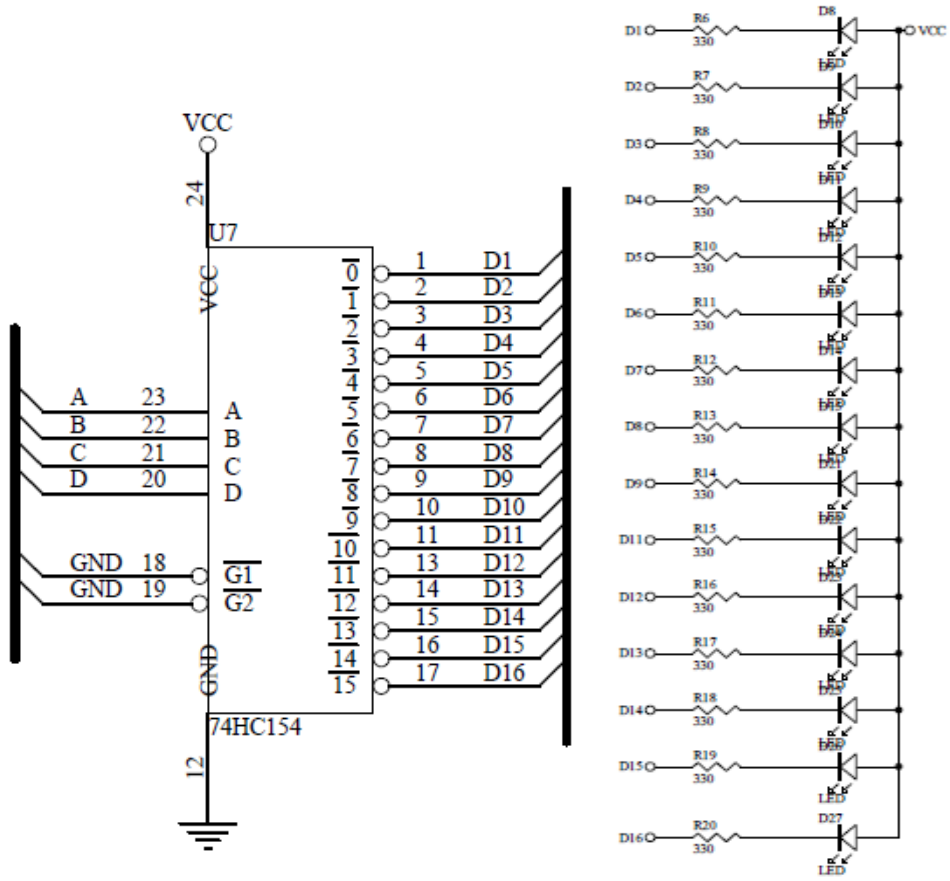


Figura 2-28.- Señalización de la comunicación con los nodos esclavos

Para llevar a cabo dicha tarea, se utilizó el circuito integrado 74HC154, el mismo que es un demultiplexor, que se puede observar a continuación:

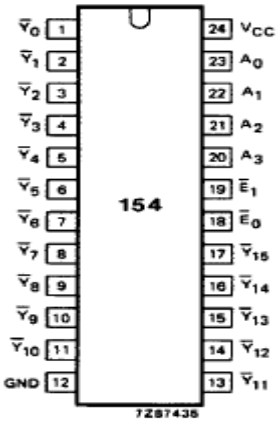


Figura 2-29.- Demultiplexor 74HC154 [49]

La función de este demultiplexor es decodificar un estrada de 4 bits en código binario a una de las 16 posibles salidas que se muestran en la tabla siguiente.

Tabla 2-8.- Tabla de función del demultiplexor 74HC154 [49]

		INPUTS				OUTPUTS															
$\overline{E}_0$	$\overline{E}_1$	A_0	A_1	A_2	A_3	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$	$\overline{Y}_8$	$\overline{Y}_9$	$\overline{Y}_{10}$	$\overline{Y}_{11}$	$\overline{Y}_{12}$	$\overline{Y}_{13}$	$\overline{Y}_{14}$	$\overline{Y}_{15}$
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L

Donde:

H = Nivel TTL alto (5Vdc)
L = Nivel TTL bajo (0Vdc)
X = Cualquier valor

De acuerdo a esta última tabla, se fijó un cero lógico (0Vdc) en los pines $\overline{E_0}$ y $\overline{E_1}$ para poder obtener cualquiera de las 16 salidas posibles correspondientes a las entradas A_0, A_1, A_2 y A_3.

Como se observa, para cada caso todos los pines de salida se ponen en nivel TTL alto excepto uno. Por esta razón, los ánodos de los diodos led están conectados a Vcc (5Vdc) y en serie a sus cátodos va conectada una resistencia de 330Ω, la cual se conecta a los respectivos pines del demultiplexor. Así sólo uno de los diodos led se encenderá en cada caso. Dicha resistencia limita la corriente que circula por el diodo led a 15mA.

El microcontrolador del nodo maestro se programó de tal forma que se pueda señalizar ciertos eventos importantes de acuerdo a la siguiente tabla.

Tabla 2-9.- Tabla de función del demultiplexor 74HC154

Codificación en binario				Señalización
A_0	A_1	A_2	A_3	
0	0	0	0	Inicia a correr el programa en el microcontrolador
1	0	0	0	Nodo esclavo de temperatura 1 responde
0	1	0	0	Nodo esclavo de temperatura 2 responde
1	1	0	0	Nodo esclavo de temperatura 3 responde
0	0	1	0	Nodo esclavo de temperatura 4 responde
1	0	1	0	Nodo esclavo de corriente responde

0	1	1	0	Nodo esclavo de humedad 1 responde (Expansión)
1	1	1	0	Nodo esclavo de humedad 2 responde (Expansión)
0	0	0	1	Nodo esclavo de presencia de agua 1 responde
1	0	0	1	Maestro pide información a esclavos
0	1	0	1	Nodo esclavo de presencia de agua 2 responde
1	1	0	1	Expansión
0	0	1	1	Expansión
1	0	1	1	Nodo maestro envía información al computador
0	1	1	1	Nodo maestro se comunica con el computador
1	1	1	1	Expansión

En el microcontrolador Atmega64 o en la placa de adaptación para el microcontrolador Atmega164, los pines A, B, C y D están conectados a los pines A_0, A_1, A_2 y A_3 del multiplexor, respectivamente.

También, se dispuso diodos led para indicar cuando haya presencia de humo y/o la puerta esté abierta. La figura siguiente muestra las conexiones realizadas.

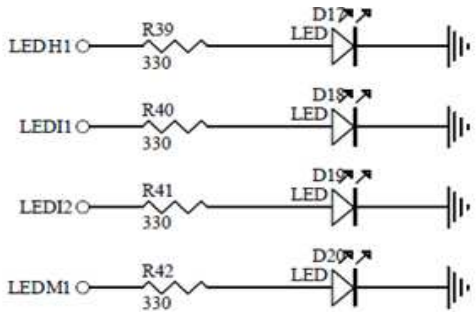


Figura 2-30.- Señalización para el sensor de humo y magnético

De igual forma se empleó resistencias de 330Ω para limitar la corriente por los diodos led a 15mA.

2.3.8 CAPACIDAD DE AMPLIACIÓN

Como se vio en los apartados anteriores, el nodo maestro se diseñó de tal forma que se pueda aprovechar al máximo la capacidad del microcontrolador Atmega64, lo cual brinda la posibilidad de ampliar el sistema SECURE SYS.

A continuación se enlista el hardware que está disponible para una ampliación:

- 3 Salidas de relé
- 12 Pines de entrada/salida
- 1 entrada opto-acoplada
- 8 Conversores análogo digital
- 4 Interrupciones externas

2.4 NODOS ESCLAVOS

En los siguientes apartados se explica detalladamente cómo fueron diseñados y construidos los diferentes nodos esclavos.
En el Anexo A se pueden visualizar el ruteado y las placas resultantes de los respectios nodos.

2.4.1 NODO DE PRESENCIA DE AGUA

El nodo de presencia de agua cumple con la función de determinar la existencia de agua dentro de un área de interés. Para cumplir esta función se ha diseñado y construido circuitería basándose en el esquema siguiente:

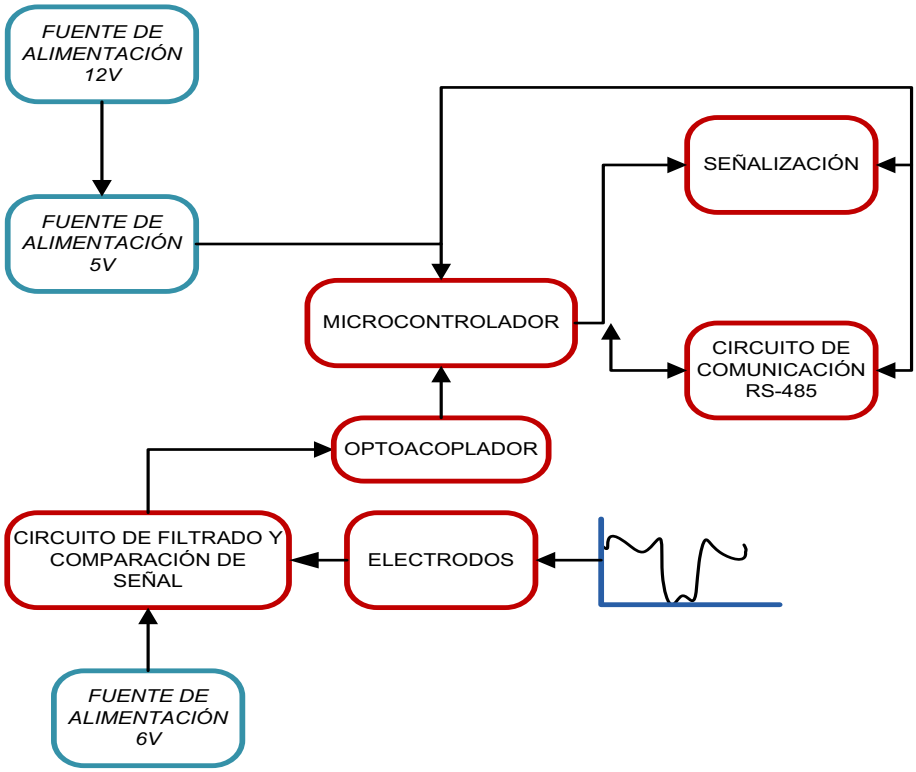


Figura 2-31.- Esquema general del nodo de presencia de agua

El nodo de presencia de agua está constituido por un circuito de comparación de detección de agua y un circuito de adquisición de datos y enlace a la red de comunicación RS-485.

2.4.1.1 Circuito de comparación de detección de agua.

Para detectar la presencia de agua en las instalaciones se emplea un circuito comparador de voltaje. La salida de este circuito se encarga de señalar la presencia de agua al cambiar su estado de 0 lógico a 1 lógico en niveles de voltaje TTL.

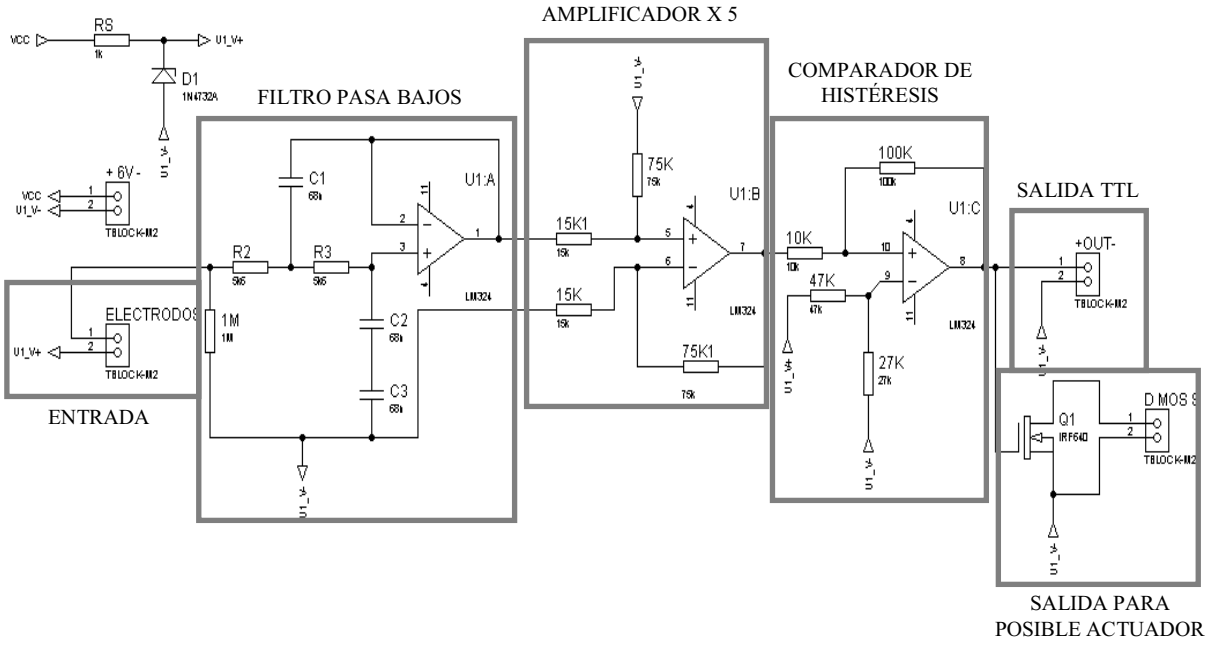


Figura 2-32.- Circuito de detección de agua

El circuito de detección de agua consta de las siguientes tres etapas:

2.4.1.1.1 Circuito de filtrado de señal.

Con el fin de evitar que el ruido electromagnético captado por los electrodos provoque una errónea activación de la salida del detector de agua, se colocó un filtro activo de Butterworth pasa bajos de segundo orden con frecuencia de filtrado resultante de Fc = 85.5Hz, debido a los elementos disponibles en el mecado. El diseño de este circuito se presenta a continuación:

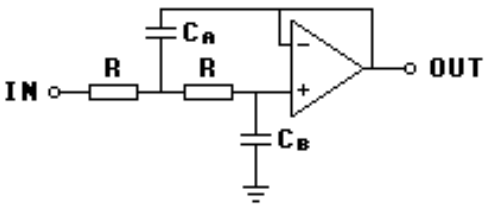


Figura 2-33.- Diseño del filtro activo pasa bajos de segundo orden [54]

En este circuito se tiene las siguientes ecuaciones:

$$Ca = \frac{1.414}{(2 * \pi * Fc * R)}$$

$$Cb = \frac{0{,}7071}{(2 * \pi * Fc * R)}$$

Tomando como consideraciones de diseño:

Debido a que el objetivo fue eliminar ruido de alta frecuencia, que es el que podría suscitar una falsa alarma, se consideró como frecuencia de filtrado $Fc = 80Hz$. Este valor de frecuencia se consideró adecuado ya que el ruido de equipos electrónicos, tales como fuentes de poder de computadores, generan un ruido con una frecuencia alrededor de 40KHz.[58] Además, en los circuitos integrados es frecuente que se presente mezclado el ruido blanco con el ruido rosa, siendo este último el predominante a bajas frecuencias. El ruido rosa tiene mayor intensidad a la frecuencia de 100Hz. [59]

Sea:

$$R = 5.6K\Omega$$

Entonces:

$$Ca = \frac{1.414}{(2 * \pi * 80 * 5.6K)} F$$

$$Ca = 502.33nF$$

$$Cb = \frac{0.7071}{(2 * \pi * 80 * 5.6K)} F$$

$$Cb = 251.3nF$$

Debido a la disponibilidad de elementos estandarizados en el mercado se emplean los siguientes elementos:

$$Ca = 470nF$$

$$Cb = 235nF;$$

Obteniéndose el valor de Cb poniendo dos capacitores de 470nF en serie. Recalculando la frecuencia de filtrado o de corte para estos valores de capacitancia, se tiene que:

$$Fc = 85.5Hz$$

De esta forma, el filtro dejará pasar las señales con frecuencias menores a 85,5 Hz y eliminará las de valores superiores.

A continuación se muestra las conexiones realizadas para acoplar los electrodos para detección de agua al circuito de filtrado:

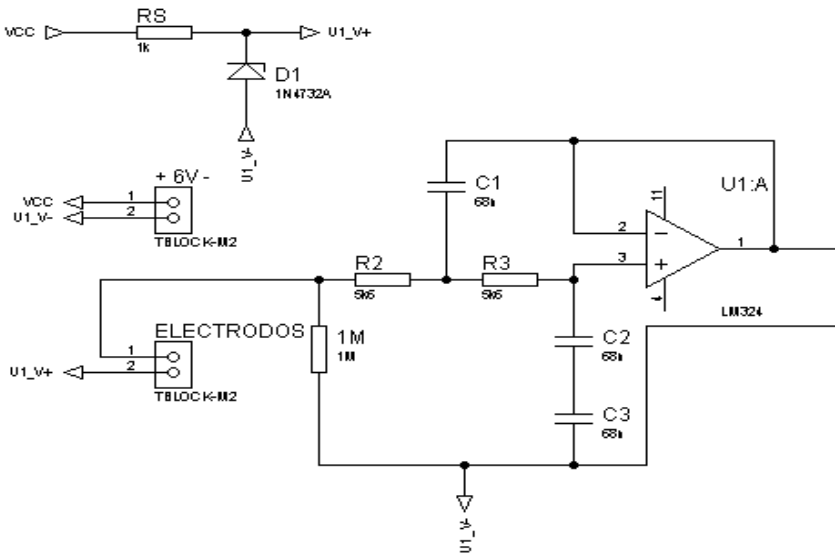


Figura 2-34.- Implementación del filtro activo de Butterworth pasa bajos de segundo orden

2.4.1.1.2 Circuito de amplificación de señal.

Debido a la poca conductividad del agua, el nivel de voltaje en la entrada y salida del filtro activo puede llegar a ser muy bajo en caso de que no existan suficientes impurezas en el agua para garantizar una buena conductividad. Para minimizar

los inconvenientes causados por este hecho, se emplea un amplificador diferencial a la salida del filtro activo.

Para el diseño de este circuito se emplea la configuración que se presenta a continuación:

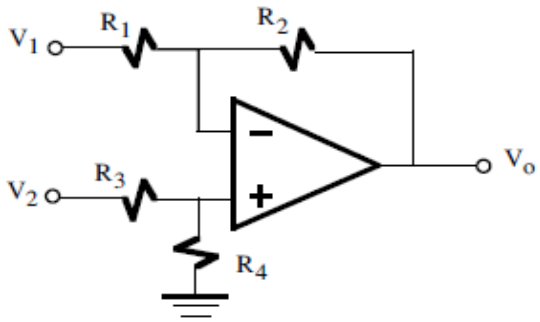


Figura 2-35.- Circuito amplificador diferencial [55]

Si las resistencias R2 = R4 y R1 = R3, se tiene la siguiente ecuación:

$$\mathrm{Vo} = (\mathrm{V_2} - \mathrm{V_1})\frac{\mathrm{R_2}}{\mathrm{R_1}}$$

En donde R2/R1 representa la ganancia del amplificador diferencial, denominada G.

Sea G=5 y R1 = 15kΩ se tiene que:

$$R_2 = 75K\Omega$$

Este valor de ganancia se tomó para garantizar que el nivel de voltaje que aparezca en la entrada del comparador de voltaje, que se muestra más adelante, sea lo suficientemente alto para activar la salida del mismo.
Como resultado se implemento el siguiente circuito:

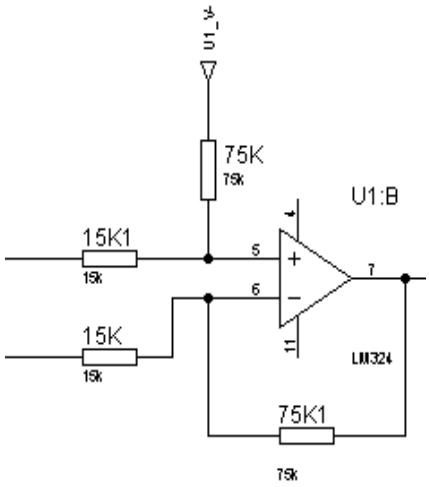


Figura 2-36.- Diseño del amplificador diferencial para el nodo de presencia de agua

2.4.1.1.3 Circuito de comparación de voltaje con histéresis

Finalmente, para indicar la presencia de agua se emplea un circuito comprador de voltaje con histéresis el cual permite determinar si el nivel de voltaje detectado corresponde a la continuidad entre los electrodos debido a la presencia de agua. Tomando en cuenta que la fuente de alimentación (Vcc) de este circuito es una fuente switching con salida de 6Vdc y después de diodo zener se obtiene un voltaje de 5V (U1_V+), se propone la siguiente ventana de histéresis para el diseño del comprador de voltaje:

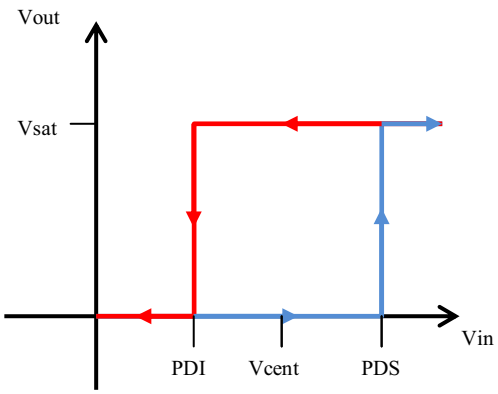


Figura 2-37.- Ventana de histéresis del circuito de comparación de voltaje

Donde:

PDI = 1.5Vdc ; punto de disparo inferior.

PDS =2.5Vdc ; punto de disparo superior.

Vcent = 2 Vdc = (PDI + PDS)/2.

Vsat = 5 Vdc ; Voltaje de saturación.

Esta ventana corresponde al circuito de comparación con histéresis que se presenta a continuación:

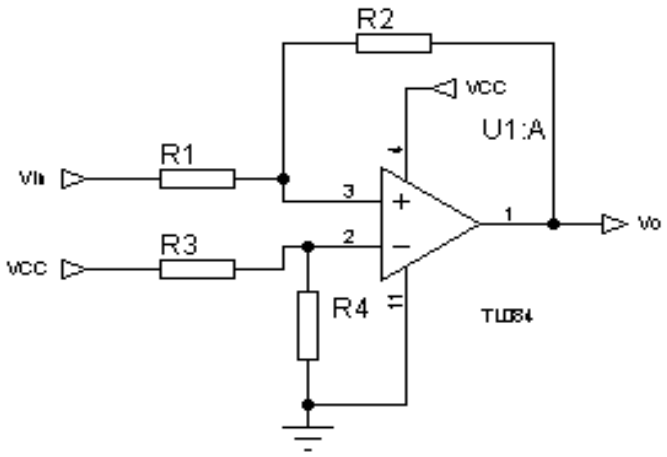


Figura 2-38.- Circuito comparador de voltaje con histéresis no inversor con desplazamiento de ventana

El comportamiento de este circuito obedece las siguientes ecuaciones:

$$\mathrm{Vcent} = \mathrm{Vref}\left(1 + \frac{\mathrm{R}_1}{\mathrm{R}_2}\right) \quad (1)$$

$$Vref = Vcc\left(\frac{R_4}{R_3 + R_4}\right) \quad (2)$$

$$PDS = Vcent + Vsat\frac{R_1}{R_2} \quad (3)$$

$$PDI = Vcent - Vsat\frac{R_1}{R_2} \quad (4)$$

Vsat: es el voltaje máximo que aparece a la salida del amplificador operacional el cual tiene un valor aproximado a U1_V+.

$$Vsat = 5Vdc$$

Reemplazando las consideraciones de diseño en la ecuación número 3 se tiene:

$$2.5 = 2 + 5\frac{R_1}{R_2}$$

$$\frac{R_2}{R_1} = 10$$

De acuerdo a este resultado se toman los siguientes valores:

$$R_1 = 10K\Omega \ \wedge \ R_2 = 100K\Omega$$

Reemplazando las consideraciones de diseño en la ecuación número 1 se tiene:

$$2 = Vref\left(1 + \frac{1}{10}\right)$$

$$Vref = 1.818Vdc$$

Reemplazando este resultado en la ecuación 2:

$$1.818 = 5\frac{R_4}{R_3 + R_4}$$

$$R_3 + R_4 = 2.75R_4$$

$$R_3 = 1.75R_4$$

De acuerdo a este resultado se toman los siguientes valores:

$$R_4 = 27K\Omega \wedge R_3 = 47K\Omega$$

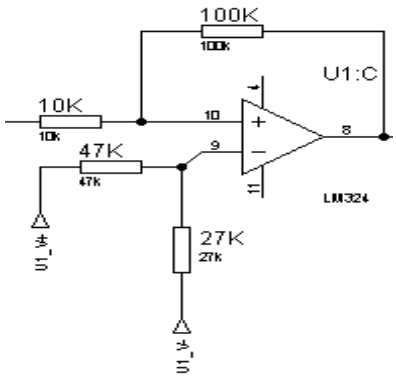


Figura 2-39.- Implementación del circuito de comparación de voltaje con histéresis

2.4.1.2 Circuito de Adquisición de Datos y Enlace a la red RS-485

Para enviar el estado de detección de agua al nodo maestro, se emplea un circuito de enlace a la red RS-485. Este circuito consta de las siguientes secciones y se muestra en la figura 2-40:

- Comunicación con la red RS-485.
- Fuente de alimentación.
- Microcontrolador.
- Señalización.
- Pines de programación del microcontrolador.
- Entrada optoacoplada para obtener información del estado del detector de agua.
- Medición de humedad ambiental.

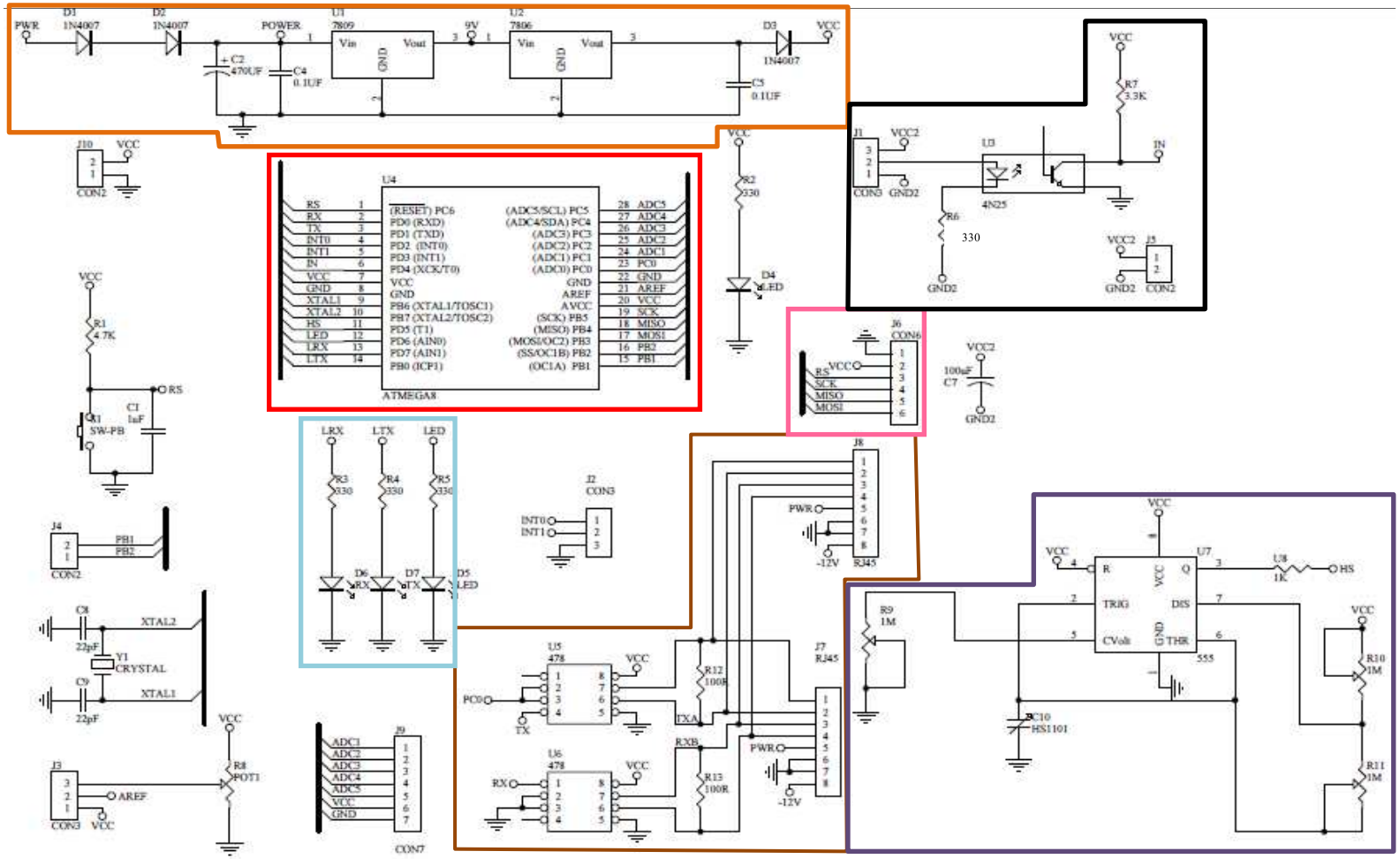


Figura 2-40.- Esquema general del circuito de adquisición de datos y procesamiento del nodo de presencia de agua

2.4.1.2.1 Selección del microcontrolador

Los requerimientos que se toman en cuenta para la selección del microcontrolador son los siguientes:

- 1 puerto de comunicación serial.
- 4 salidas para señalización.
- 1 pin detección de esta de detector de agua.
- Capacidad de memoria para realizar operaciones matemáticas y para completar el proceso de comunicación.

Considerando los requerimientos expuestos anteriormente se han determinado como candidatos a los siguientes microcontroladores:

Tabla 2-10. Especificaciones requeridas para el microcontrolador

	# de Pines	Pines I/O	Interrupciones Externas	Canales ADC	Puertos USART
ATmega 8	28	23	2	8	1
ATmega 48	28	23	24	8	1

De acuerdo con estas consideraciones se determina como la mejor opción al microcontrolador ATmega 48.

2.4.1.2.2 Controlador del circuito.

El microcontrolador del sistema tiene la tarea de verificar el estado del detector de agua que va conectado a una entrada opto-acoplada. También cumple con la función de enviar esta información al nodo maestro a través de la red RS-485.

El microcontrolador cuenta con un circuito de reset el cual se emplea para reiniciar el programa del microcontrolador, en caso de que se lo requiera.

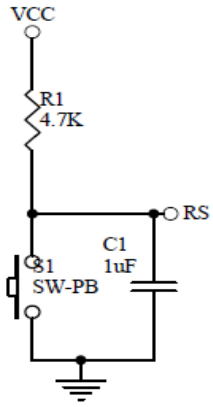


Figura 2-41.- Circuito de reset del microcontrolador

2.4.1.2.3 Comunicación con la red RS-485.

El nodo de presencia de agua emplea, para la comunicación con el nodo maestro, un circuito de conversión de comunicación en niveles de voltaje TTL a niveles de voltaje del estándar RS-485.

Para esto se emplean dos circuitos integrados MAX487, uno para recepción y otro para el envío de datos. La configuración de uno de estos circuitos integrados esta comandada por el microcontrolador de este circuito que conmuta la configuración entre estado de alta impedancia y configuración de emisor. El primer estado se mantiene mientras el nodo maestro no realice ninguna petición de información a este nodo en particular. Al darse el caso de petición de información por parte del nodo maestro, el microcontrolador cambia la configuración del circuito integrado a emisor, para que se pueda responder a la petición. El otro circuito integrado MAX487 siempre se encuentra en configuración de receptor, con el fin de captar todas las peticiones del nodo maestro.

2.4.1.2.4 Fuente de alimentación.

Para dar energía a los diferentes elementos del circuito de adquisición de datos se emplea el circuito siguiente:

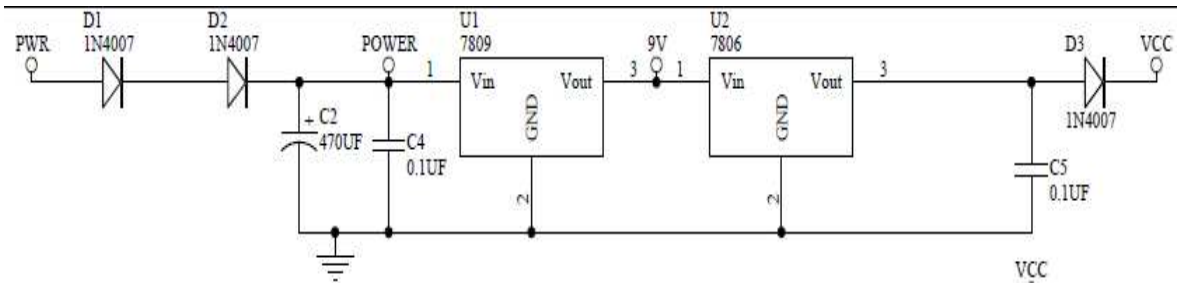


Figura 2-42.- Fuente de alimentación del nodo de presencia de agua

Para el diseño de esta fuente de alimentación se tomaron las mismas consideraciones que se describieron en el nodo maestro. Como se observa, en este caso se emplean dos diodos rectificadores a la entrada los cuales cumplen con dos funciones: proteger la circuitería que está por detrás de PWR y para bajar el este voltaje de entrada en 1.4Vdc para aliviar el trabajo del regulador LM7809.

2.4.1.2.5 Señalización.

Para indicar el estado del nodo se emplean leds como señalización; lo cuales se encuentran conectados al microcontrolador. El microcontrolador comanda el encendido y apagado de los leds dependiendo de si se ha recibido un dato o enviado un dato.

Debido a que no es aconsejable que la corriente por los diodos led sobrepase los 20mA, se emplearon resistencias de 330Ω para limitar la corriente a 15mA, lo cual está dentro del límite máximo de corriente que pueden soportar los pines del microcontrolador, que como se vio anteriormente, este límite es de 40mA.

2.4.1.2.6 Pines para programación del microcontrolador.

Para facilitar la actualización del software del sistema microcontrolado en caso de una ampliación, se ha puesto a disponibilidad del usuario los pines para la programación de microcontrolador.

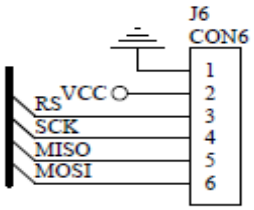


Figura 2-43.- Pines de programación del microcontrolador

2.4.1.2.7 Adquisición del estado del detector de agua.

El monitoreo de presencia de agua se realiza a través de una entrada opto-acoplada; para evitar que fallas en el detector de agua afecten el funcionamiento del circuito de adquisición de datos.

Para garantizar un funcionamiento adecuado del opto-acoplador se emplea una resistencia de 330Ω en el led de entrada haciendo circular una corriente de alrededor de 15mA por el mismo. La luminosidad obtenida por la circulación de esta corriente produce una saturación adecuada del transistor del opto-acoplador; enviando de esta manera una señal de nivel TTL 1 lógico al pin del microcontrolador que se encuentra conectado al opto-acoplador.

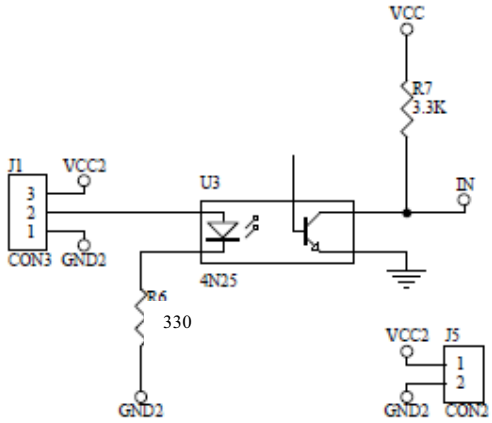


Figura 2-44.- Entrada opto-acoplada del circuito de adquisición de datos del nodo de presencia de agua

2.4.1.2.8 Medición de humedad en el ambiente.

Como una opción adicional, se incluye en el circuito de adquisición de datos los elementos necesarios para implementar la medición de humedad en el ambiente a través del sensor de humedad HS1101.

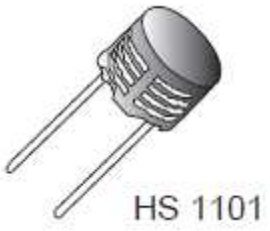


Figura 2-45.- Sensor capacitivo de humedad en el ambiente [56]

El sensor se implementa con un circuito integrado oscilador NE555 con el que se arma un generador de pulsos, cuya frecuencia está determinada por la variación en la capacitancia del sensor HS1101 debido a la variación del nivel humedad en el ambiente.

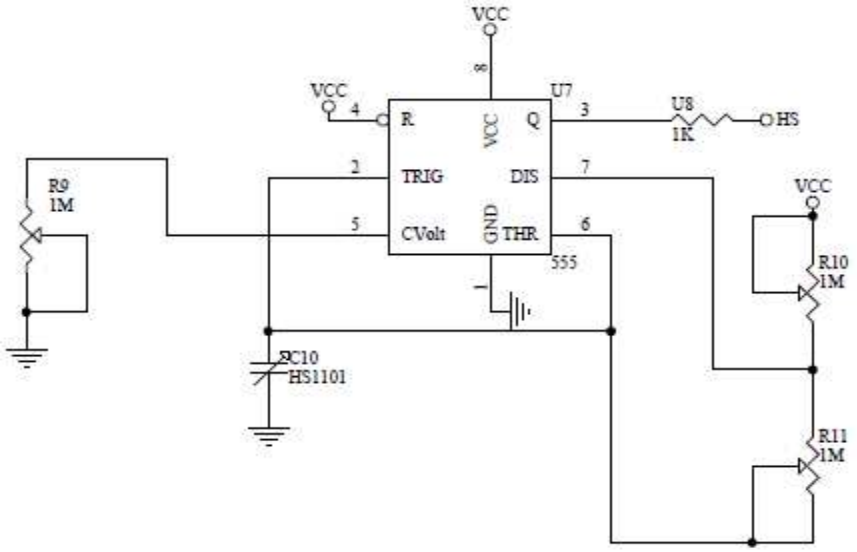


Figura 2-46.- Circuito oscilador para la mediación de humedad ambiental

Los valores de los elementos necesarios para que este circuito funcione correctamente se encuentran en el Anexo C, en el correspondiente datasheet del elemento.

2.4.2 NODO DE CORRIENTE

El nodo de corriente se encarga de la adquisición del valor de corriente que alimenta a las instalaciones y de enviar este valor al nodo maestro a través de la red RS-485. Para cumplir esta tarea se diseñó y construyó la circuitería de este nodo basándose en el siguiente esquema:

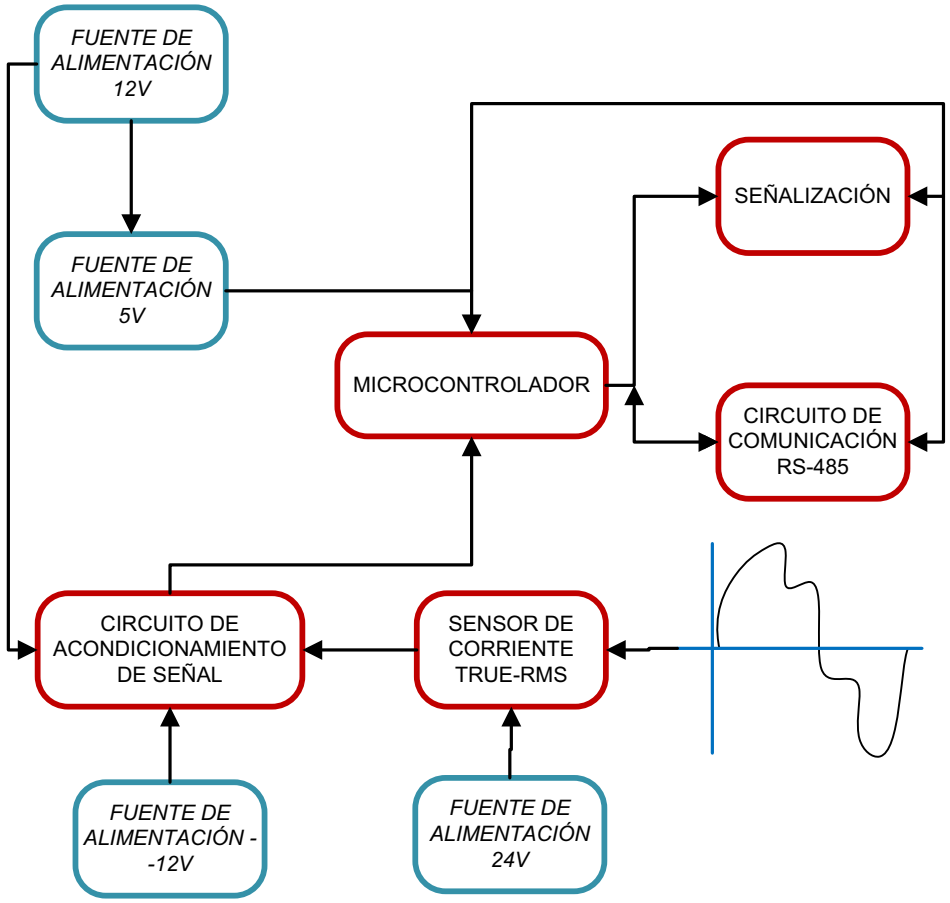


Figura 2-47.- Esquema general del nodo de corriente

2.4.2.1 Sensado de corriente

La adquisición del valor de corriente se realiza a través del transductor de núcleo abierto AKR 50 B420L, cuya salida varía entre 4 a 20 mA. El valor de corriente a la salida es directamente proporcional al valor TrueRMS de la corriente que circula por el conductor que se coloque en el núcleo del transductor. Este sensor dispone de un jumper para seleccionar el rango de corriente que se estima medir, y ofrece tres opciones: 0 a 10A, 0 a 20A ó 0 a 50A.

Este sensor se muestra en la figura siguiente:

Figura 2-48.- Transductor de Corriente AKR 50 B420L [51]

Este transductor de corriente requiere de una fuente de alimentación de 24Vdc para su correcto funcionamiento. El circuito diseñado para cumplir con este requerimiento se presenta a continuación:

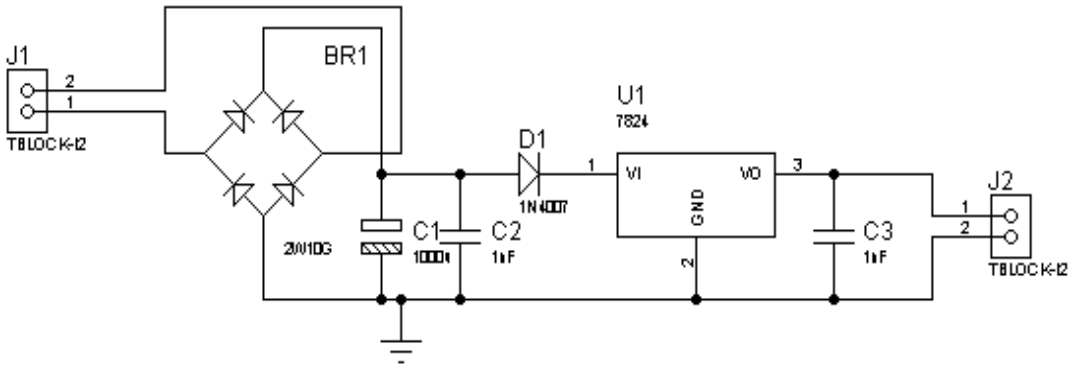


Figura 2-49.- Fuente de 24Vdc para sensor de corriente

Esta fuente toma energía de la red de corriente alterna, emplea un puente de diodos para rectificar la señal de voltaje proveniente de un transformador con relación de transformación de 5/1 (120Vac / 24Vac), dejando pasar solamente la parte positiva del voltaje de alimentación. La salida del puente rectificador es conectada a un condensador electrolítico de 1000uF a 35Vdc, que tiene la finalidad de disminuir el rizado en la señal de voltaje. A continuación se coloca un capacitor cerámico de 1nF para eliminar el ruido eléctrico de alta frecuencia. Para obtener un voltaje DC[27] constante de 24V se emplea el circuito integrado LM7824. Adicionalmente se emplea otro capacitor cerámico para filtrar el ruido eléctrico a la salida del la fuente de voltaje.

2.4.2.2 Acondicionamiento de la señal

El acondicionamiento se realiza poniendo una carga resistiva (load) de 250Ω en serie con la fuente de alimentación y el transductor de corriente, como se observa en la Figura 2-50, esto permite obtener un valor de voltaje proporcional a la corriente de alimentación de las instalaciones.

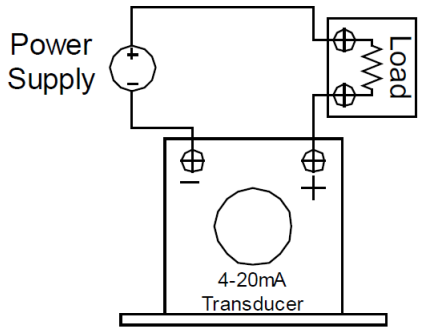


Figura 2-50.- Diagrama de conexión del transductor de corriente AKR 50 B420L [52]

Tras obtener una señal de voltaje proporcional al valor de corriente de alimentación de las instalaciones se debe transformar esta señal para que sea

[27] Direct Current (Corriente continua)

posible transmitir su valor al nodo maestro a través de la red RS-485. Esta señal de voltaje se encontrará a 1V cuando el valor de la corriente de alimentación sea cero y a 5Vdc cuando la corriente medida sea el valor máximo del rango seleccionado, por esta razón, con el objetivo de obtener un voltaje entre 0 a 4Vdc, se empleó un circuito restador. Para esto se emplean en primer lugar un circuito emisor-seguidor tanto en el voltaje resultante de la medición de corriente como en la referencia de voltaje que se resta de la anterior. La referencia de voltaje a restar se obtiene de un potenciómetro calibrado para tener en el terminal variable el voltaje de 1Vdc.

El circuito restador (Figura 2-51) se diseña a partir de la siguiente ecuación:

$$Vo = \frac{R_4}{R_1}(V_1) - \frac{R_3}{R_2}(V_2)$$

Donde V_1 y V_2 son los niveles de voltaje a restar.

Considerando que no se requiere amplificación en ninguno de los valores a restar, las relaciones R_4/R_1 y R_3/R_2 se establecen como unitarias, y se toma el valor de estas resistencias como de 10KΩ.

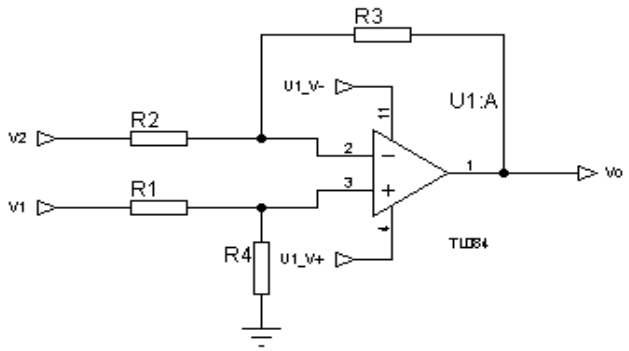


Figura 2-51.- Circuito restador a base de un amplificador operacional

El circuito de acondicionamiento completo se presenta en la figura siguiente:

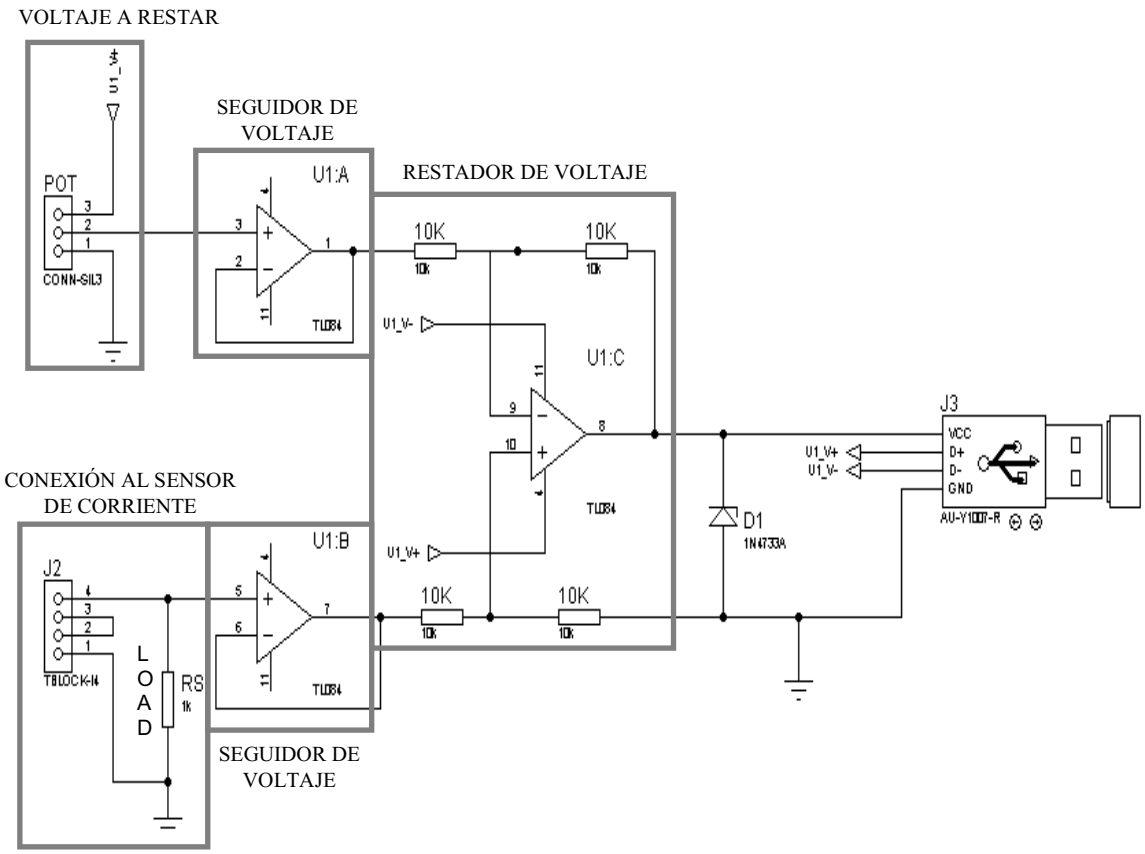


Figura 2-52.- Circuito de acondicionamiento de señal para el transductor de corriente

2.4.2.3 Circuito de Adquisición de Datos y Enlace a la red RS-485

Este circuito tiene como finalidad la adquisición de los datos y su transmisión al nodo maestro a través de la red RS-485. Para cumplir esta función se emplea el hardware de la figura 2-53, en la cual se pueden distinguir las siguientes secciones:

- Comunicación con la red RS-485.
- Adquisición de valor de corriente.
- Fuente de alimentación.
- Microcontrolador.
- Señalización.
- Pines para programación del microcontrolador

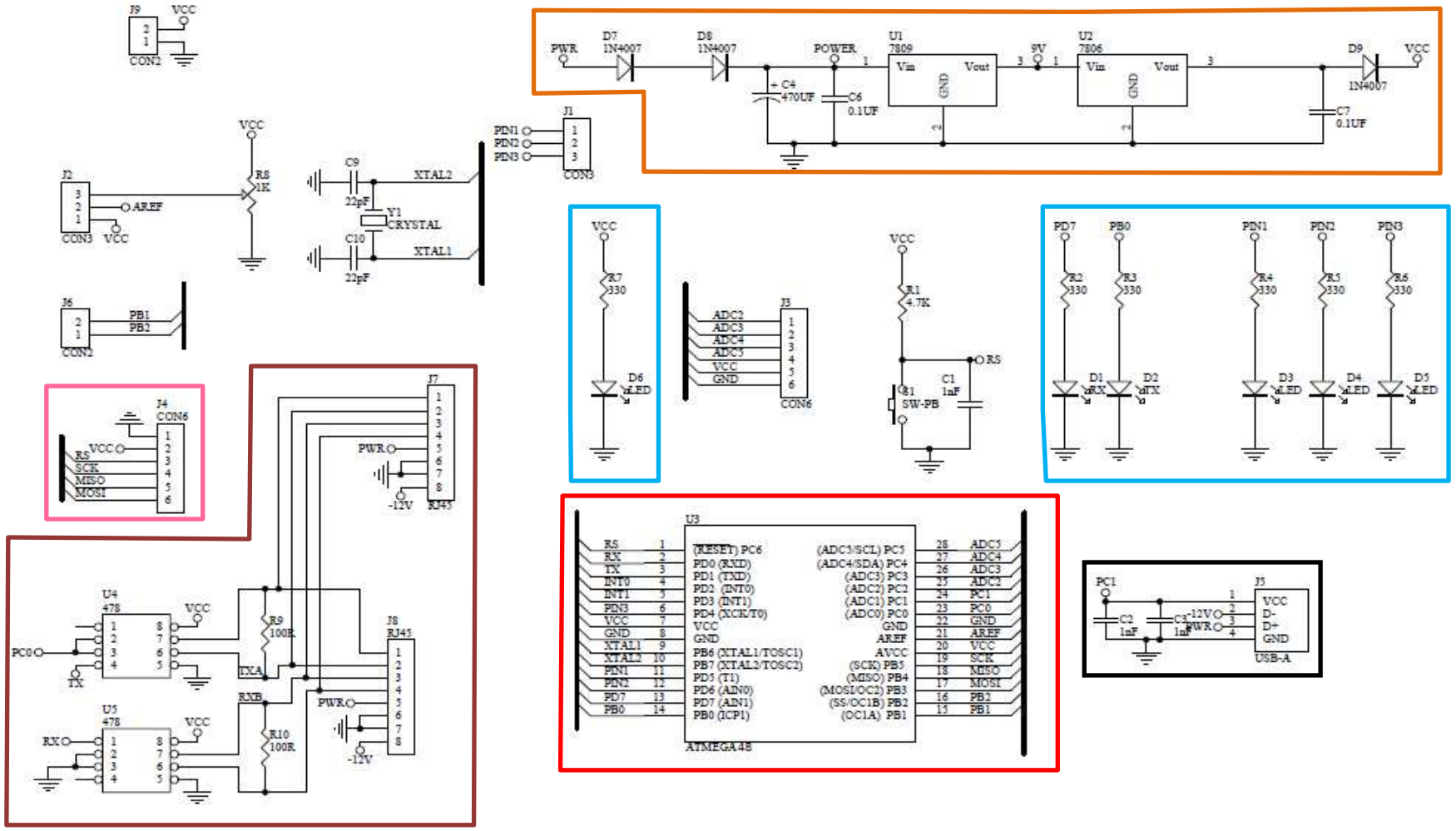


Figura 2-53.- Circuito de adquisición de datos y enlace a la red RS-485 del nodo de corriente

2.4.2.3.1 Selección del microcontrolador

Para seleccionar un microcontrolador adecuado a los requerimientos de este nodo se toman en cuenta los elementos a emplear y el número de pines requeridos.

Requerimientos:

- 1 puerto de comunicación serial.
- 3 salidas para señalización.
- 1 conversor análogo digital (1 pin).
- Capacidad de memoria para realizar operaciones matemáticas y para completar el proceso de comunicación.

Considerando los requerimientos expuestos anteriormente se han determinado como candidatos a los siguientes microcontroladores:

Tabla 2-11. Especificaciones requeridas para el microcontrolador del nodo de corriente

	# de Pines	Pines I/O	Interrupciones Externas	Canales ADC	Puertos USART
ATmega 8	28	23	2	8	1
ATmega 48	28	23	24	8	1

De acuerdo con estas consideraciones se determina como la mejor opción al microcontrolador ATmega 48.

2.4.2.3.2 Controlador

El microcontrolador cumple las funciones de realizar la adquisición del valor de corriente a través de uno de los puertos del conversor análogo/digital. Este valor

de nivel de voltaje es interpretado como un valor de corriente y enviado al nodo maestro utilizando el puerto USART del microcontrolador que se conecta al circuito de comunicación con la red RS-485.

Para la adquisición del valor de corriente se emplea un conector de tipo USB-A cuya conexión se muestra en la figura siguiente.

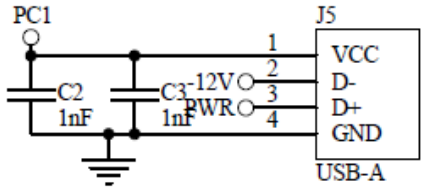


Figura 2-54.- Conector USB-A, empleado para adquirir la señal de voltaje proporcional a la corriente de alimentación

El microcontrolador también se encarga de señalizar la recepción y envío de datos, así como la alimentación de energía.

2.4.2.3.3 Fuente de alimentación

La fuente de alimentación para el nodo de corriente fue diseñada de igual forma que como se describió en los anteriores nodos. Esta sirve de alimentación para el circuito de adquisición de datos y comunicación con la red RS-485 y para señalización en este nodo.

El circuito de alimentación implementado se muestra a continuación:

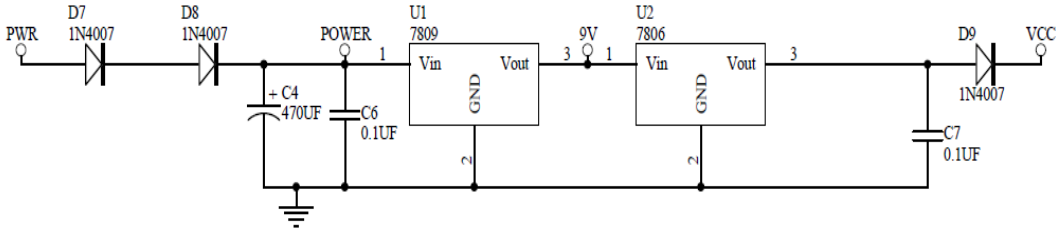


Figura 2-55.- Fuente de alimentación del nodo de corriente

2.4.2.3.4 Comunicación con la red RS-485

Con el fin de enlazar el sistema microprocesado a la red RS-485 se emplearon dos circuitos integrados SN75176, los cuales reciben la señal de datos seriales del microcontrolador y la convierten a señales de voltaje diferencial correspondiente al estándar de comunicación RS-485. La comunicación es sincronizada por el nodo maestro, por lo que uno de los SN75176 se encontrará siempre en configuración de receptor mientras que el otro se encontrará en alta impedancia hasta que se recibe un comando determinado del nodo maestro, con el cual se activa la transmisión.

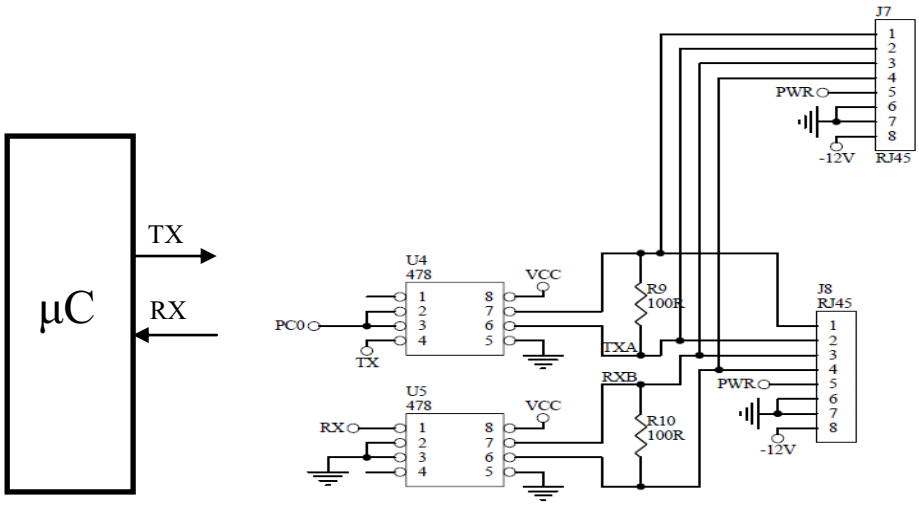


Figura 2-56.- Circuito de enlace a la red RS-485

2.4.2.3.5 Programación del microcontrolador.

Para facilitar la actualización del software del sistema microprocesado, se colocó un zócalo de programación.

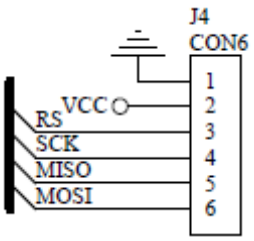


Figura 2-57.- Pines de programación del microcontrolador

Se debe tomar en cuenta que para la programación del microcontrolador del circuito de adquisición de datos este debe estar desconectado de la red RS-485 y del circuito de acondicionamiento de la señal de corriente.

El circuito de adquisición de datos cuenta con elementos adicionales no esenciales para el funcionamiento del sistema SECURE SYS, que se han dejado disponibles para futuras ampliaciones al sistema. Entre los elementos que se dejaron disponibles se encuentran: conversores análogo digital, diodos led adicionales para señalización, interrupciones externas y pines de entrada/salida digital.

2.4.3 NODOS DE TEMPERATURA

Cada nodo de temperatura tiene la función de adquirir el valor de temperatura dentro del área en la que se encuentra instalado y transmitir esta información al nodo maestro. Para cumplir con esta función se diseñó un circuito siguiendo el siguiente esquema:

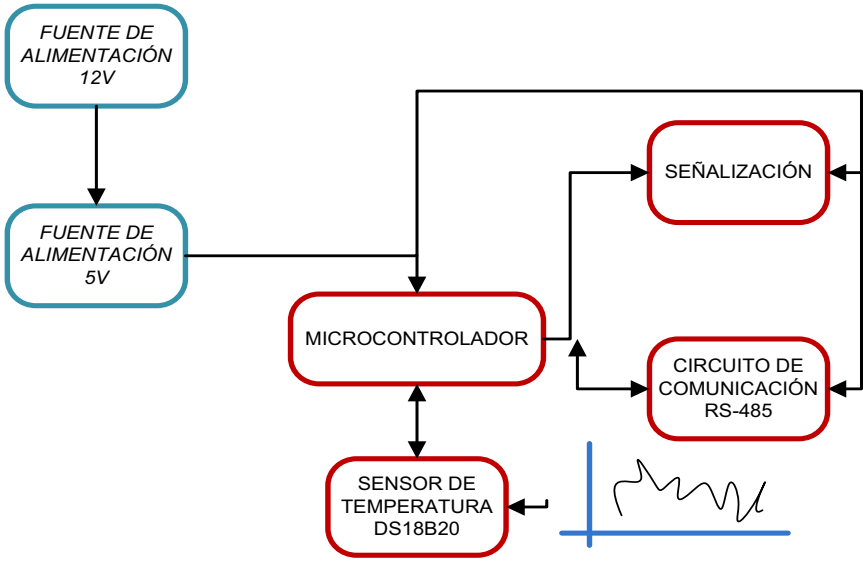


Figura 2-58.- Esquema general del nodo de temperatura

2.4.3.1 Sensado de temperatura

Para realizar la medición de temperatura se emplea un el sensor DS18B20, el cual se comunica con el sistema microcontrolado del nodo a través de un solo hilo (tecnología 1wire) (Figura 2-18).

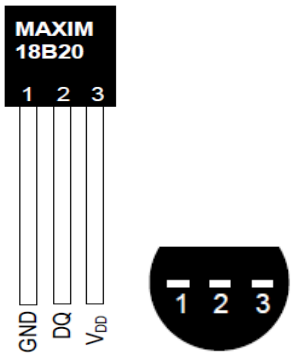


Figura 2-59.- Sensor de temperatura 1wire DS18B20 [53]

Este sensor de temperatura entrega un valor digital del valor de temperatura con resolución de 9 o 12 bits, que puede ser seleccionada por el usuario. Puede o no

emplear una fuente de alimentación externa de 5Vdc. A continuación se exponen algunas de las características técnicas de este sensor.

Tabla 2-12. Características del sensor de temperatura DS18B20

Característica	Valor
Voltaje de alimentación máximo	6Vdc
Formato de salida de datos	°C
Rango de temperatura de operación	-55°C a +125°C
Máximo consumo de corriente	9mA

Este tipo de sensor puede conectarse en paralelo a varios sensores del mismo tipo, gracias a que cada sensor tiene asociado una dirección única, permitiendo que el microcontrolador identifique la información de cada sensor. Así se puede medir la temperatura de varios lugares, empleando únicamente un pin del microcontrolador.

Para realizar la comunicación con el sensor DS18B20 el microcontrolador debe enviar una serie de comandos. Los comandos necesarios para realizar la adquisición de datos del sensor de temperatura se describen a continuación:

Tabla 2-13. Comandos para realizar la transferencia de datos

Comandos (Hexadecimales)		Acción
F0h	Search ROM	Cuando este comando es enviado los sensores conectados al pin de comunicación envían sus respectivos códigos únicos de 64 bits. Para

		determinar cuántos y cuáles son los sensores conectados al pin el, dispositivo maestro debe realizar un proceso de peticiones y eliminaciones sucesivas.
55h	Match ROM	Este comando permite al dispositivo maestro dirigir los comandos que se muestran más adelante a un sensor en específico, el cual es seleccionado al enviar un código de 64bits, único de cada sensor, a continuación del comando 55h.
CCh	Skip ROM	Este comando le permite al dispositivo maestro comandar a todos los sensores, sin enviar códigos únicos de identificación.
44h	Convert T	Tras recibir este comando el sensor empieza el proceso de medición y transformación de temperatura a formato digital. El dispositivo maestro debe esperar al menos 10ms antes de leer los datos.
4Eh	Write Scratchpad	El dispositivo maestro puede emplear este comando para escribir en los registros T_H y T_L, los cuales se emplean como base de comparación para la determinación interna de alarmas.
48h	Copy Scratchpad	Tras este comando el sensor de temperatura realiza una copia en la memoria EEPROM de los valores en los registros T_H y T_L.
BEh	Read Scratchpad	Este comando inicia la transmisión de datos desde el sensor hacia el dispositivo maestro, la cual termina al transmitirse el byte de CRC[28] que corresponde al noveno byte.
B8h	Recall E	Este comando se emplea para cargar los valores de T_H y T_L guardados en la EEPROM.

[28] Cyclic Redundancy Check (Comprobación de redundancia cíclica)[57]

La cadena de datos enviados por este sensor no sólo contiene los bytes en los que se encuentra la información de temperatura, sino que envía información adicional, como se observa en la siguiente tabla:

Tabla 2-14. Cadena de Bytes recibidos del sensor de temperatura [53]

	Descripción
Byte 0	LSB[29] de Temperatura
Byte 1	MSB[30] de Temperatura
Byte 2	Registro T_H
Byte 3	Registro T_L
Byte 4	Reservado
Byte 5	Reservado
Byte 6	COUNT REMAIN (C0h)
Byte 7	COUNT PER °C (10h)
Byte 8	CRC

Los bytes 0 y 1 son los empleados para obtener el valor de temperatura. Los bytes 6 y 7 se emplean para obtener un valor de temperatura con resolución extendida.

2.4.3.2 Circuito de Adquisición de Datos y Enlace a la red RS-485

El circuito de adquisición de datos se encarga de comunicarse con el par de sensores de temperatura conectados en paralelo y de recibir la información de temperatura de estos. Posteriormente estos valores son enviados al nodo maestro por medio de la red RS-485.

[29] Less Significant Bits (Bits menos segnificantes)
[30] Most Significant Bits (Bits más significantes)

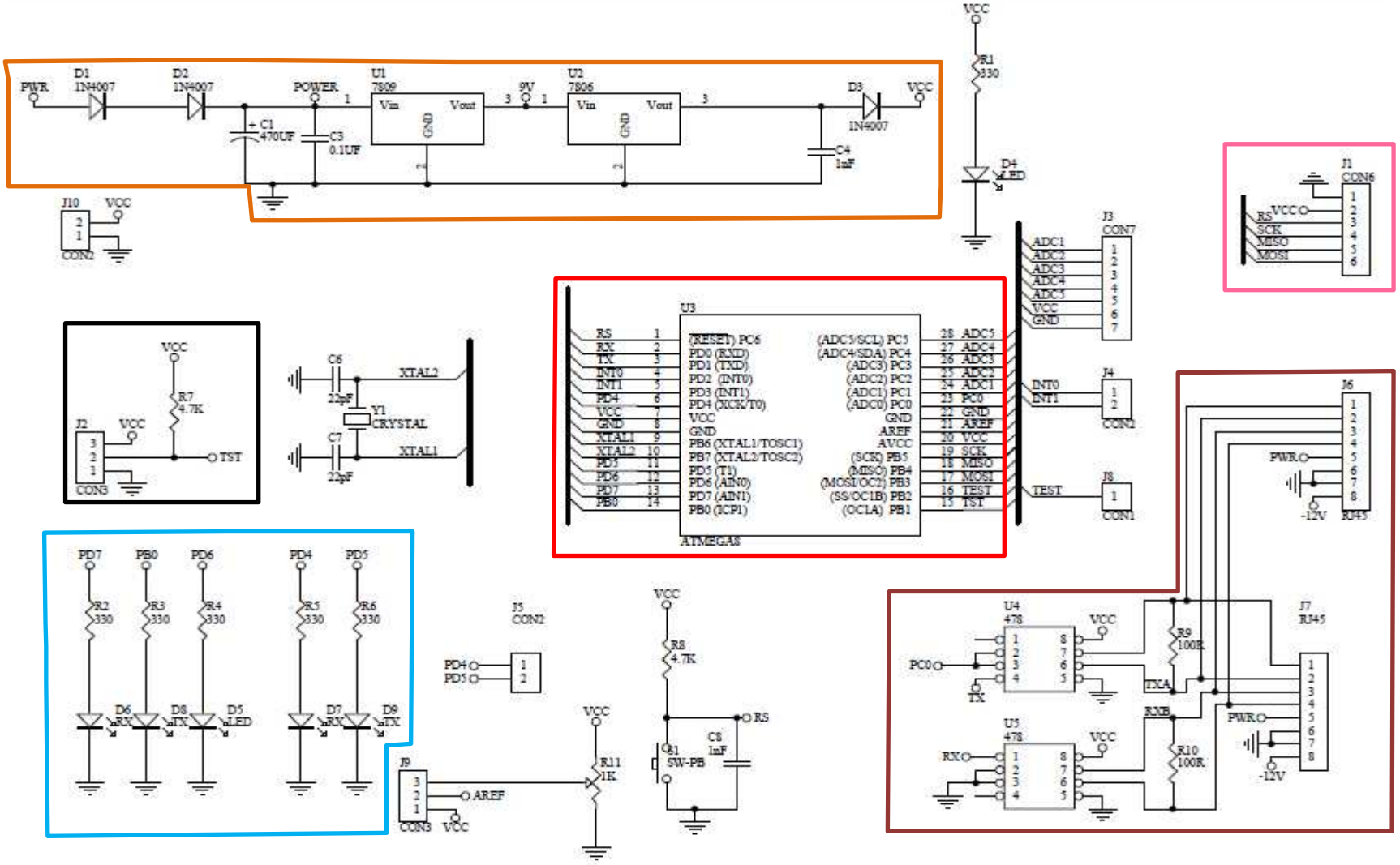


Figura 2-60.- Circuito de adquisición de datos y enlace a la red RS-485 del nodo de Temperatura

Observando la figura anterior se pueden distinguir las siguientes secciones:

- Comunicación con la red RS-485.
- Fuente de alimentación.
- Controlador.
- Señalización.
- Pines para programación del microcontrolador
- Adquisición de valor de temperatura.

2.4.3.2.1 Selección del microcontrolador

Como se describió anteriormente en el nodo de corriente, para la selección del microcontrolador se toma en consideración el número y tipo de periféricos que se requieren, y también el número de pines que estos ocupan. Para este nodo se hacen las siguientes consideraciones:

- 1 puerto de comunicación serial.
- 4 salidas para señalización.
- 1 pin para comunicación 1-Wire
- Capacidad de memoria para realizar operaciones matemáticas y para completar el proceso de comunicación.

Considerando los requerimientos expuestos anteriormente se han determinado como candidatos a los siguientes microcontroladores:

Tabla 2-15. Especificaciones requeridas para el microcontrolador

	# de Pines	Pines I/O	Interrupciones Externas	Canales ADC	Puertos USART
ATmega 8	28	23	2	8	1
ATmega 48	28	23	24	8	1

De acuerdo con estas consideraciones se determina como la mejor opción al microcontrolador ATmega 48.

2.4.3.2.2 Controlador

En este circuito, el microcontrolador se encarga de enviar los comandos necesarios a los sensores DS18B20, como se verá en el siguiente capítulo, para adquirir los valores temperatura.

El microcontrolador también cumple con la tarea de señalizar el funcionamiento del nodo, es decir, la recepción y transmisión de datos.

Finalmente, el microcontrolador tiene la función de enviar los datos de temperatura al nodo maestro, para lo cual emplea el puerto de comunicación serial USART, el mismo que está conectado al circuito de enlace a la red RS-485.

2.4.3.2.3 Fuente de alimentación

El circuito de la siguiente figura se emplea como fuente de alimentación de los elementos del circuito del nodo de temperatura: microcontrolador, elementos de señalización, dos circuitos integrados SN75176 para el enlace con la red RS-485 y dos sensores de temperatura DS18B20.

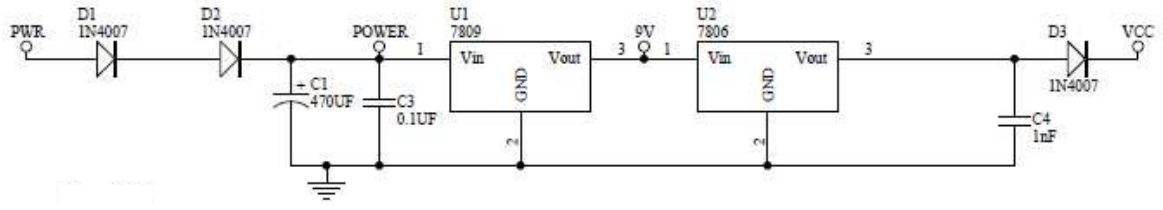


Figura 2-61.- Fuente de alimentación del nodo de temperatura.

Las consideraciones tomadas para el diseño de este circuito son las ya detalladas en los anteriores nodos.

2.4.3.2.4 Comunicación con la red RS-485

El siguiente circuito se emplea para transmitir los datos de temperatura obtenidos de los sensores DS18B20.

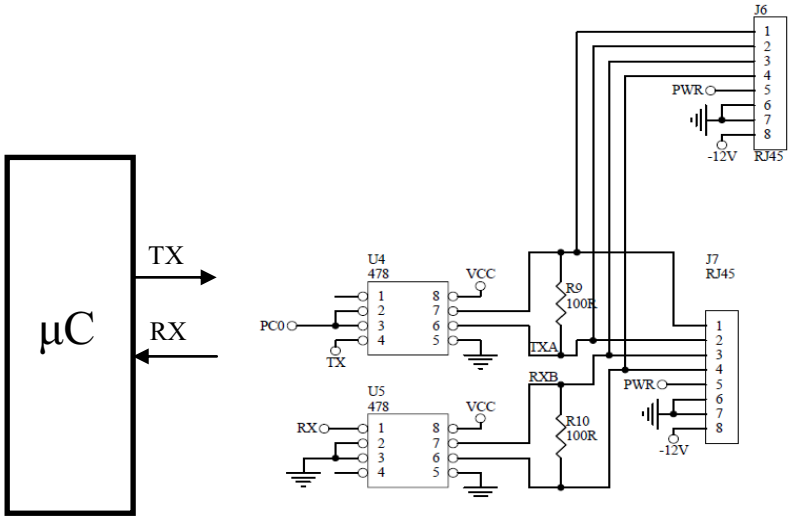


Figura 2-62.- Circuito de enlace a la red RS-485

Los elementos U4 y U5 son los circuitos integrados SN75176, los cuales se emplean para unir el nodo de temperatura a la red RS-485 y comunicarse con el nodo maestro. Como se menciono antes, el microcontrolador del circuito de comunicación con la red RS-485 se encarga de determinar si la transmisión de datos se habilita o se mantiene en alta impedancia.

2.4.3.2.5 Pines para programación del microcontrolador.

De igual forma que en los anteriores nodos, se dispone a este nodo de un zócalo para programación del microcontrolador. El tipo de programación que se puede realizar el del tipo SPI.

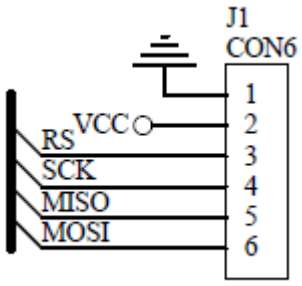


Figura 2-63.- Pines de programación del microcontrolador

El circuito del nodo de temperatura cuenta con entradas adicionales que se dejan a disposición del usuario. Estas entradas pueden ser empleadas en caso que se que requiera hacer ampliaciones al sistema SECURE SYS. Entre las funciones disponibles se tiene: entradas al conversor análogo digital, interrupciones externas, pines de entrada y salida y elementos para señalización.

2.5 NODO DE HUMO

El sistema SECURE SYS consta de dos nodos de humo. Cada nodo está conformado por un sensor de humo y su circuito de acondicionamiento, cuya salida va conectada directamente a un pin del microcontrolador, como se pudo ver en el apartado 2.3.2. Debido a que esta salida es de tipo ON/OFF y por tanto no requiere un procesamiento dedicado, no se diseño un nodo esclavo que envíe por la red RS-485 si hay o no presencia de humo, sino que se programó al microcontrolador para que verifique periódicamente si se ha suscitado una alarma.

2.5.1.1 Sensor implementado

Figura 2-64.- Detector de humo

En el sistema SECURE SYS se emplearon dos detectores de humo de ionización (BRK 9120B), como el de la figura, alimentados por corriente alterna y con batería de respaldo. Estos sensores poseen un botón de prueba/silencio, que al ser oprimido permite diferentes funciones tales como prueba del detector, silenciamiento de la alarma y reposicionamiento de la función de enclavamiento. También disponen de un indicador de enclavamiento, el cual indica cual fue el detector que inició la alarma aún después de haber desaparecido la condición de alarma.

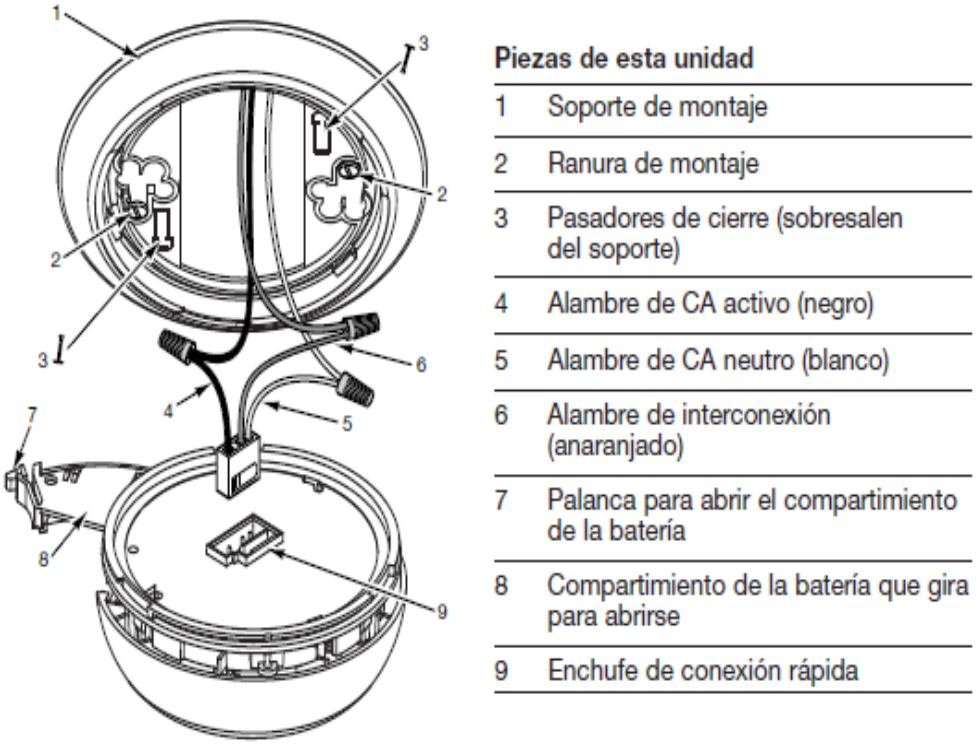
Se escogieron detectores de ionización debido a que estos son generalmente más sensibles que los sensores fotoeléctricos en la detección de partículas pequeñas, que tienden a ser producidas en mayores cantidades por fuegos llameantes que consumen los materiales combustibles rápidamente. Entre las fuentes que originan estos incendios se incluyen papeles quemados en cestos.

Las principales características del sensor de humo BRK 9120B son:

- Alimentación: 120Vac, 60Hz.
- Respaldo de alimentación: batería de 9Vdc

- Interconexión: posee un terminal de salida de interconexión, en el que se puede conectar otro sensor compatible, permitiendo así formar una red de sensores de humo que se accionarán al mismo tiempo si uno de ellos detecta una alarma. Se pueden interconectar hasta 12 sensores de humo.

En la siguiente figura se muestra las partes que conforman este sensor de humo:



Piezas de esta unidad	
1	Soporte de montaje
2	Ranura de montaje
3	Pasadores de cierre (sobresalen del soporte)
4	Alambre de CA activo (negro)
5	Alambre de CA neutro (blanco)
6	Alambre de interconexión (anaranjado)
7	Palanca para abrir el compartimiento de la batería
8	Compartimiento de la batería que gira para abrirse
9	Enchufe de conexión rápida

Figura 2-65.- Partes principales del sensor BRK 9120B [50]

De acuerdo a los eventos que se susciten o a las condiciones en las que se encuentre el sensor BRK 9120B, este señalizará como se muestra en la siguiente tabla:

COMPRENSIÓN DE LOS PATRONES DE LUZ Y BOCINA

Condición	Indicadores LED (rojo o verde)	Bocina
Funcionamiento normal (Alimentación de CA)	Indicador LED verde encendida. El indicador LED rojo destella una vez por minuto	La alarma audible no suena.
Funcionamiento normal (Alimentación de CC – Modelo 9120B)	Indicador LED verde apagado. La luz rojo LED destella una vez por minuto	La alarma audible no suena.
DURANTE LA PRUEBA	La luz rojo LED destella una vez cada segundo	Patrón de alarma: 3 sonidos, pausa, 3 sonidos, pausa
BAJA POTENCIA O FALTA DE BATERÍA (Modelo 9120B)	La luz rojo LED destella una vez por minuto	La alarma emite un "chirrido" una vez por minuto
CONDICIÓN DE ALARMA Dispositivo iniciador de la alarma de humo	El indicador LED rojo parpadea rápidamente en la unidad que activó la alarma.	Patrón de alarma: 3 sonidos, pausa, 3 sonidos, pausa, se repiten en todos los detectores
CONDICIÓN DE ALARMA Interconexión de Alarma	El LED rojo de las demás unidades de una serie interconectada estará apagado.	Patrón de alarma: 3 sonidos, pausa, 3 sonidos, pausa, se repiten en todos los detectores
MODALIDAD EN SILENCIO	El LED rojo destella una vez cada 10 segundos	La alarma permanece silenciosa durante 10 minutos. La alarma sonará si los niveles de humo aumentan.
INDICADOR DE ENCLAVAMIENTO DEL DETECTOR	El LED verde está encendido 2 segundos y apagado 2 segundos, repetidamente hasta la reinicialización, en la(s) unidad(es) iniciadora(s).	La alarma permanece silenciosa

Figura 2-66.- Señalización del sensor BRK 9120B [50]

2.5.1.2 Acondicionamiento de la señal

Para determinar si se ha generado una alarma de humo, se utilizó la señal de salida del pin de interconexión del sensor humo. Esta señal en caso de suscitarse una alarma se muestra en la figura siguiente:

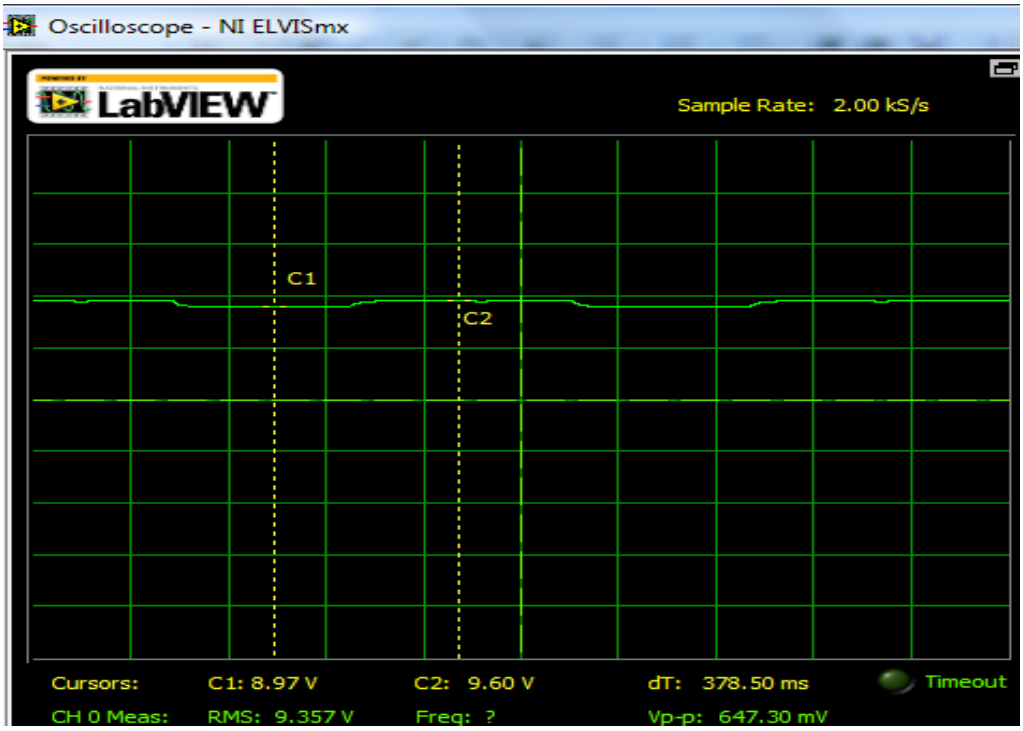


Figura 2-67.- Señal de salida del pin de interconexión del sensor de humo

Como se observa en la figura, la señal de salida del pin de interconexión oscila entre 8,97Vdc y 9,60Vdc, por esta razón se empleó el siguiente circuito de acondicionamiento para obtener una señal continua de 5V a la salida.

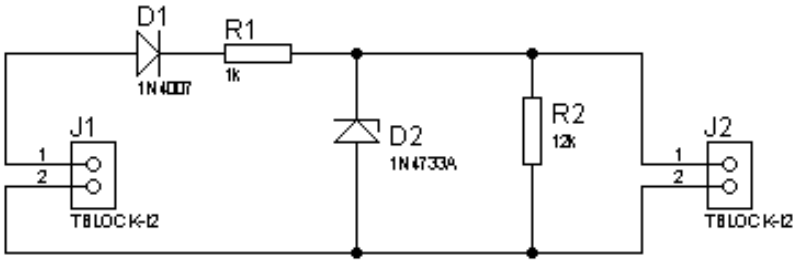


Figura 2-68.- Circuito de acondicionamiento para el sensor de humo

De acuerdo a la figura, se empleó un diodo 1N4007 como protección para que no haya circulación de corriente en sentido inverso y se elimine picos negativos de voltaje en la entrada que pueden ser causados por agentes externos. Además, se colocó una resistencia de 1kΩ en serie al diodo rectificador con el objetivo de que una vez que el diodo zener 1N4733A regule la salida a 5Vdc, el voltaje restante caiga en dicha resistencia. El voltaje en esta resistencia (R1) se calculó mediante la siguiente ecuación, que se obtiene del circuito anterior:

$$V_E = V_D + V_{R1} + V_Z$$

Donde:
V_E = Voltaje de entrada = 8,97Vdc ~ 9,60Vdc.
V_D = Voltaje en el diodo en conducción = 0,7Vdc.
V_{R1} = Voltaje en la resistencia R1
V_Z = Voltaje en el diodo zener en regulación = 5,1Vdc

Por tanto, el voltaje en la resistencia R1 cuando el diodo zener está en regulación es:

$$(8{,}97V - 5{,}1V - 0{,}7V) \leq V_{R1} \leq (9{,}60V - 5{,}1V - 0{,}7V)$$

$$3{,}17Vdc \leq V_{R1} \leq 3{,}8Vdc$$

Las resistencias R1, R2 y el diodo zener fueron dimensionadas de la siguiente manera:

Caso 1 (Sin diodo zener):

- Sea V_E = 6V. Se toma este valor para que aún en voltajes cercanos a lo que se desea a la salida (5.1V), el diodo zener pueda regular.
- Sea V_{R2} = 5,5V (para asegurar que cuando se coloque el diodo zener este regule)
- Sea R1 = 1kΩ

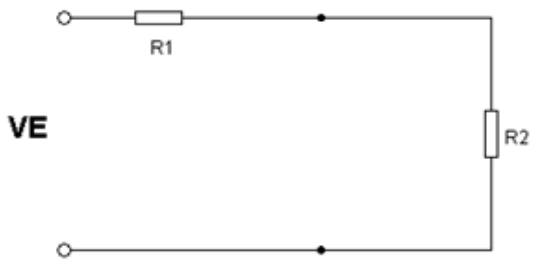


Figura 2-69.- Circuito equivalente del acondicionamiento del sensor de humo sin diodo zener

Basándose en la figura anterior se tiene:

$$V_E - V_{R2} = I_{R1} * R1$$

$$0.5Vdc = I_{R1} * 1k\Omega$$

$$I_{R1} = 0.5mA$$

Debido a que no hay zener $I_{R1} = I_{R2}$

$$I_{R2} = 0.5mA$$

Por tanto:

$$R2 = \frac{5.5Vdc}{0.5mA}$$

$$R2 = 11k\Omega, \text{por disponibilidad de elementos se empleó una de } 12\text{k}\Omega$$

Caso 2 (con zener):

- Voltaje de salida Vo = 5.1V
- Sea V_E = 9.6Vdc. En este caso se toma este valor ya que los elementos a utilizar disiparán una potencia máxima cuando V_E es máxima.

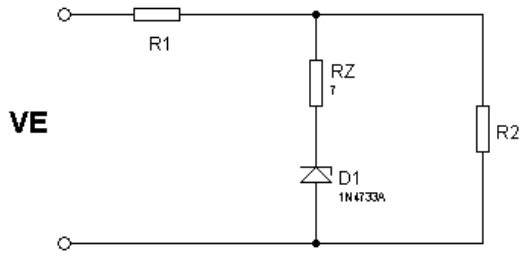


Figura 2-70.- Circuito equivalente del acondicionamiento del sensor de humo con diodo zener

La figura anterior muestra el circuito equivalente cuando se coloca el diodo zener. Basándose en este circuito se realizaron los siguientes cálculos:

Potencia de R1:

$$P_{R1} = \frac{(V_{R1})^2}{R1}$$

$$P_{R1} = \frac{(3.8Vdc)^2}{1k\Omega}$$

$$P_{R1} = 14.4mW$$

Potencia de R2:

$$P_{R2} = \frac{(V_{R2})^2}{R2}$$

$$P_{R1} = \frac{(5.1Vdc)^2}{12k\Omega}$$

$$P_{R1} = 2.16mW$$

Potencia del diodo zener:

$$P_Z = V_Z * I_Z$$

$$I_Z = I_{R1} - I_{R2}$$

$$I_Z = \frac{3.8Vdc}{1k\Omega} - \frac{5.1Vdc}{12k\Omega} = 3.38mA$$

$$P_Z = 5.1Vdc * 3.38mA = 17.2mW$$

Por tanto, se necesitaron resistencias R1 y R2 y el diodo zener de ¼ de vatio.

A continuación se muestra la señal de salida del circuito descrito:

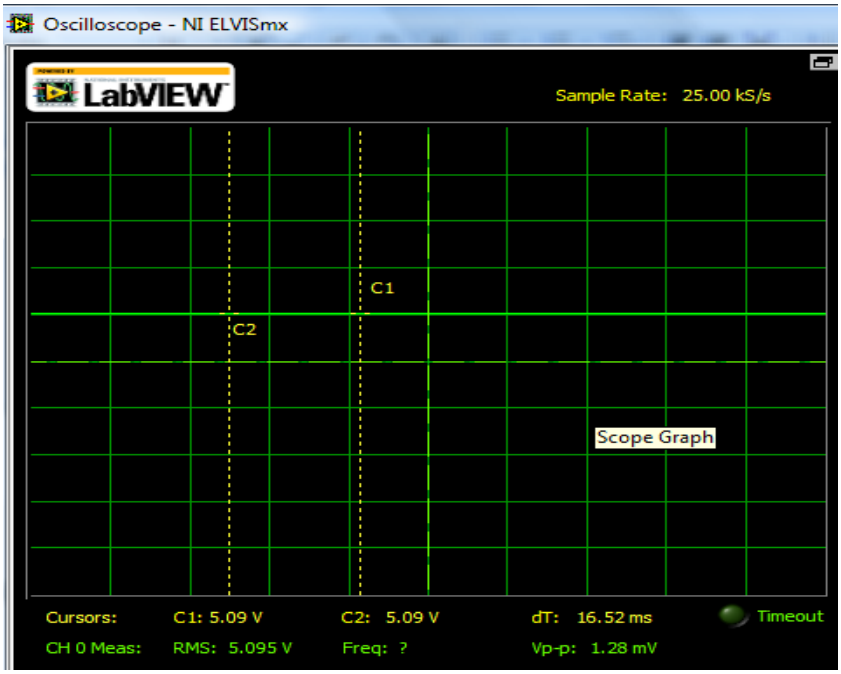


Figura 2-71.- Señal acondicionada del sensor de humo

2.6 DIAGRAMA UNIFICADO DEL SISTEMA

La figura siguiente muestra la estructura de conexión global del sistema:

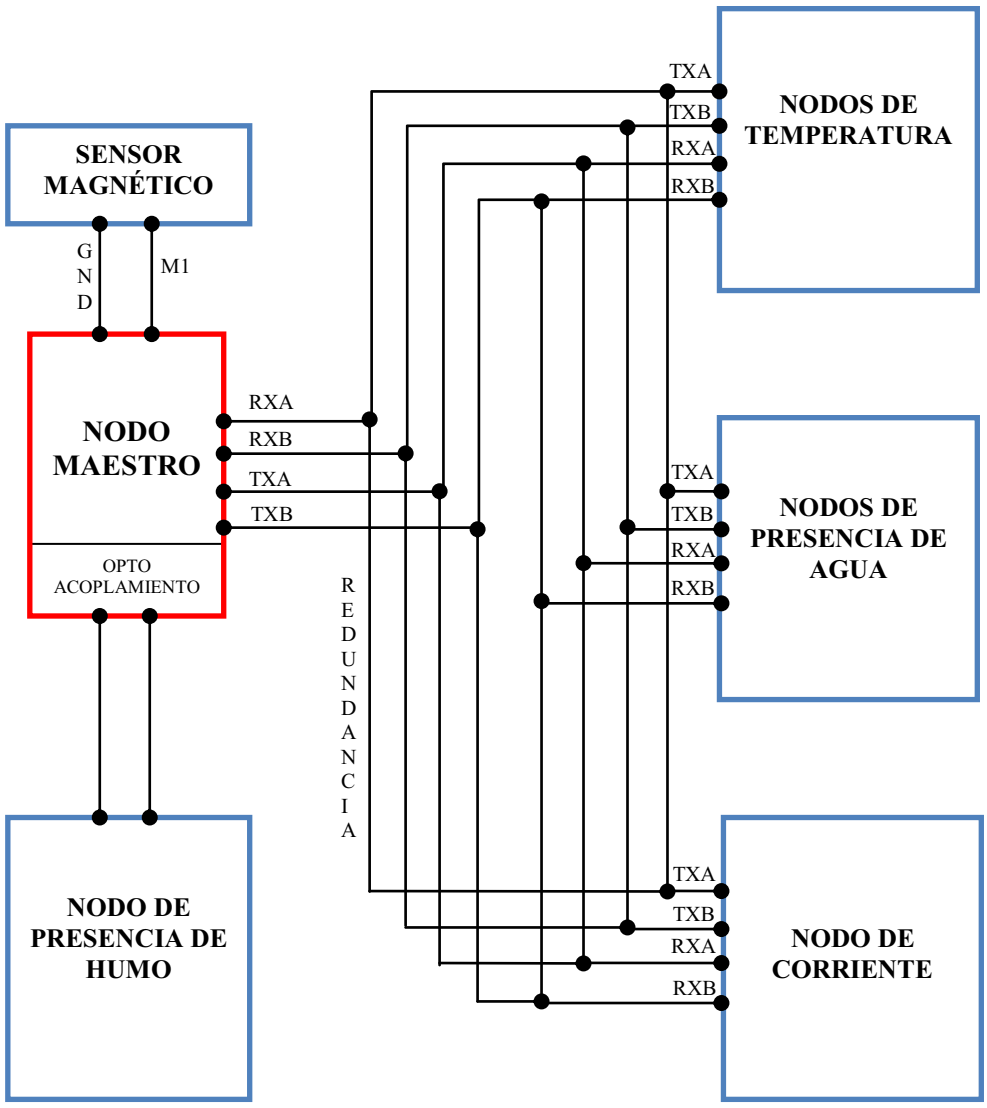


Figura 2-72.- Diagrama unificado del hardware del sistema SECURE SYS

2.7 IMPLEMENTACIÓN

De acuerdo a las actividades que se realizan en la empresa UPDATECOM Cia. Ltda. y al lugar en donde se centran estas, se dispuso el monitoreo de temperatura, presencia de agua y humo en dos departamentos de la misma: Departamento Técnico y Taller. Mientras tanto, el sensado de corriente se realiza en el tablero de breakers, sensado de apertura de puerta se lleva a cabo mediante un sensor magnético colocado sobre la misma y el accionamiento de la cerradura eléctrica se comandada a distancia.

La configuración de la red implementada es de tipo anillo. Se eligió esta configuración debido a que permite la utilización de una menor cantidad de conductores en comparación con una configuración tipo estrella. También se añadió cableado de redundancia, lo cual brinda la seguridad de que el sistema SECURE SYS siga siendo funcional en caso de existir daños en uno de los conductores que conecta cada nodo.

A continuación se describe cómo se dispusieron cada uno de los nodos y sensores dentro de las instalaciones de UPDATECOM:

2.7.1 DEPARTAMENTO TÉCNICO

Dentro del perímetro del departamento técnico se colocaron dos sensores de temperatura, dos detectores de presencia de agua y uno de humo. La ubicación de los diferentes nodos se determinó en base al espacio físico disponible y a las condiciones necesarias para el correcto funcionamiento de los mismos.

2.7.1.1 Ubicación del nodo de temperatura.

El nodo de temperatura se colocó en una de las esquinas superiores del departamento técnico y los sensores de temperatura DS18B20 se colocaron en esquinas opuestas de la habitación.

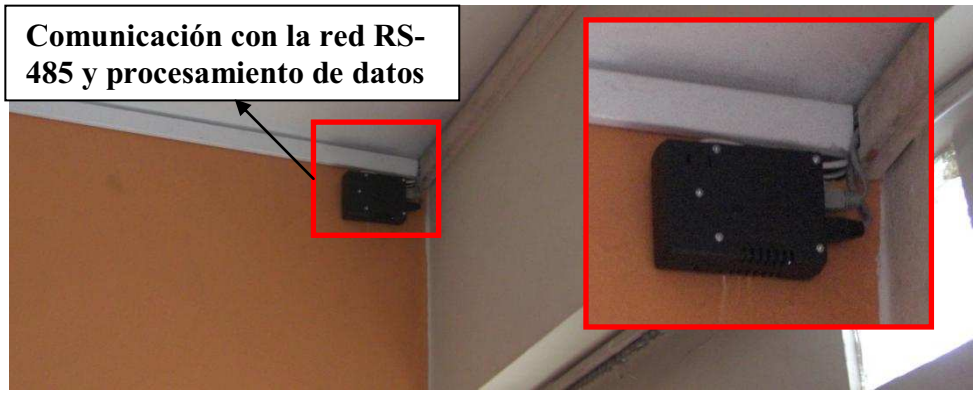


Figura 2-73.- Ubicación del nodo de temperatura en el departamento técnico

Se dispuso esta ubicación de los sensores de temperatura para poder obtener un promedio más aproximado del valor de temperatura dentro de este departamento.

En las siguientes figuras se muestran los lugares dentro del departamento técnico en donde se colocaron los sensores de temperatura, a los cuales de aquí en adelante se llamarán T1 y T2.

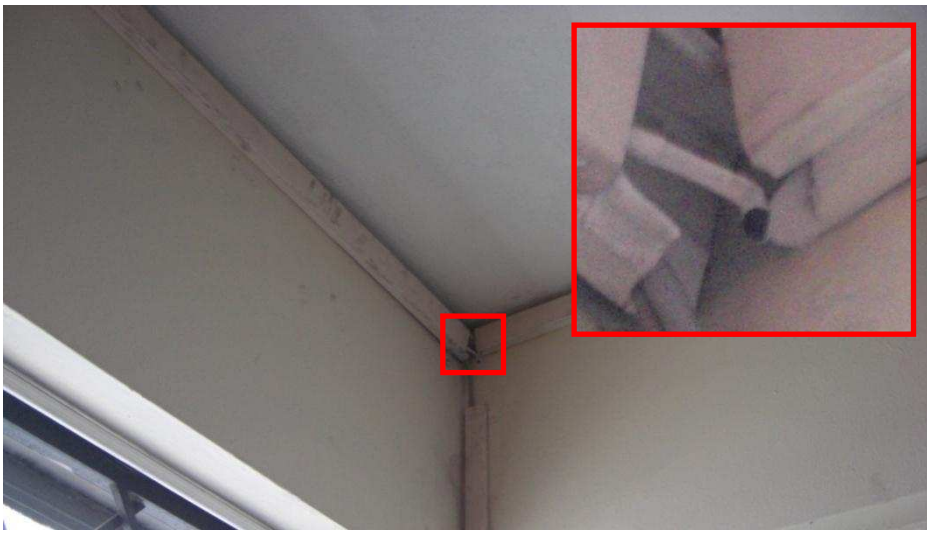


Figura 2-74.- Ubicación del sensor de temperatura T1 para el departamento técnico

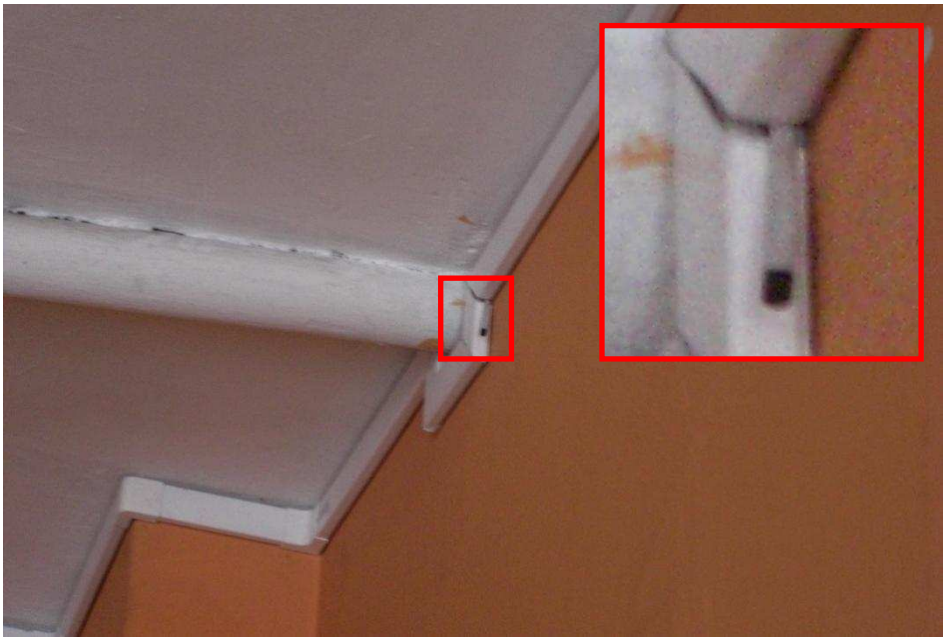


Figura 2-75.- Ubicación sensor de temperatura T2 para el departamento técnico

2.7.1.2 Ubicación del nodo de presencia de agua.

Tanto el circuito de adquisición de datos, como el circuito de comparación de voltaje y los electrodos de detección de presencia de agua se ubicaron cerca de la fuente de agua más próxima al departamento técnico.

Para evitar que la circuitería de este nodo se vea afectada por el contacto con el agua, esta fue colocada en la parte superior de la habitación, cerca al tumbado de la misma, como se observa en la figura 2-76.

Los electrodos para la detección de presencia de agua su ubicaron sobre el piso, bajo la puerta que conecta el baño con el área del departamento técnico. Esto se realizó con el fin de salvaguardar la integridad de los equipos que se encuentran dentro de este departamento.

Se emplearon dos pares de electrodos, colocados en dos puntos diferentes, de acuerdo a las observaciones que se realizaron para determinar la dirección de inclinación del piso, como en la figura 2-77.

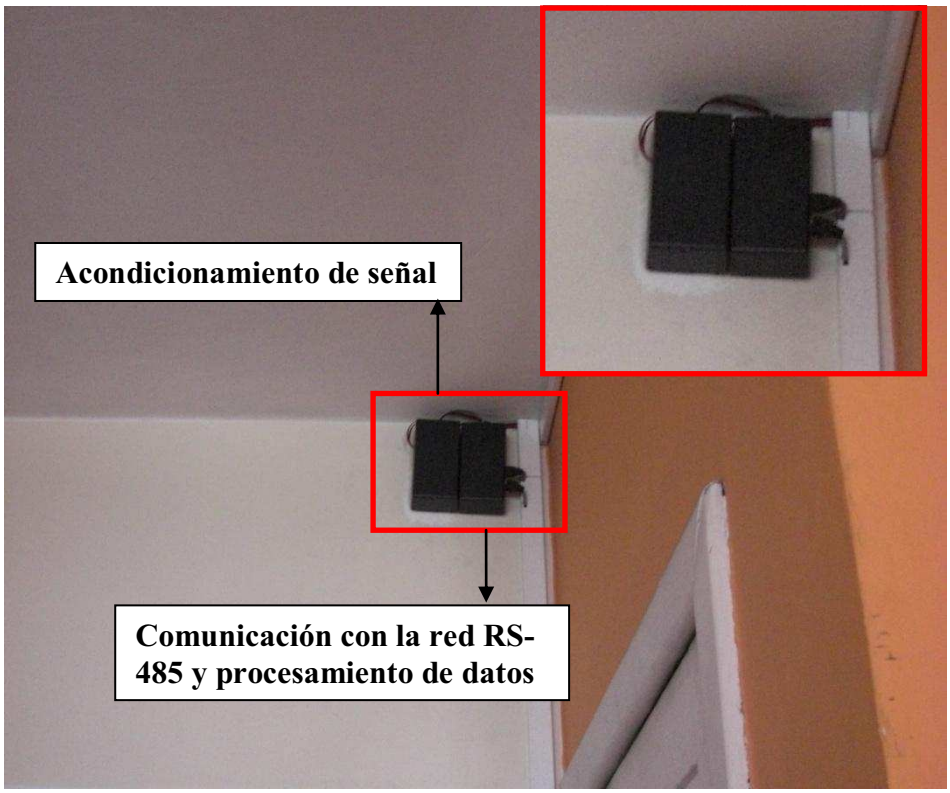


Figura 2-76.- Ubicación del nodo de presencia de agua en el departamento técnico

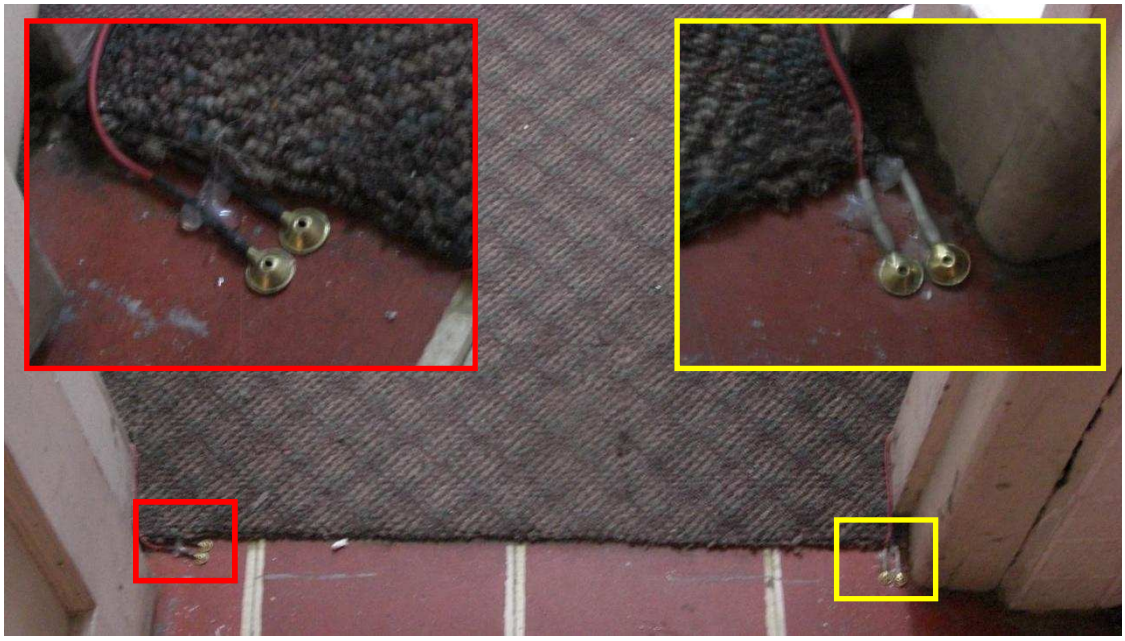


Figura 2-77.- Ubicación de electrodos en el departamento técnico

La separación entre los electrodos de cada par es mínima para garantizar la activación de la salida del comparador de voltaje cuando exista presencia de agua.

2.7.1.3 Ubicación del nodo de presencia de humo.

El detector de humo está colocado en el tumbado, en una ubicación central dentro del área del departamento técnico, para detectar fácilmente cualquier emisión de humo presente en el mismo.

Figura 2-78.- Ubicación del detector de humo en el departamento técnico

El circuito de acondicionamiento de la señal de salida del detector de humo se colocó cerca del nodo maestro con el fin de evitar que la caída de voltaje en el conductor que lleva la señal desde el detector de humo hasta el nodo maestro atenúe demasiado la misma.

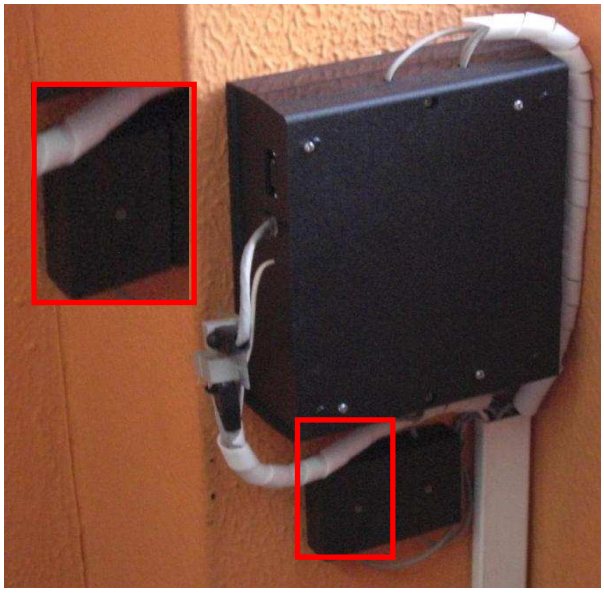


Figura 2-79.- Ubicación del circuito de acondicionamiento del detector de humo para el departamento técnico

2.7.2 TALLER

De igual forma en el taller se colocaron dos sensores de temperatura, dos detectores de presencia de agua y uno de humo. Las consideraciones que se tomaron en cuenta para la colocación de los sensores fueron las mismas que en el departamento técnico. Así:

- Los dos sensores de temperatura se dispusieron en dos equinas opuestas del taller. A estos sensores se los nombró como T3 y T4, nomenclatura que se empleará de aquí en adelante.
- Se ubicaron dos pares de electrodos en puntos opuestos de acuerdo a la inclinación del piso y lo más cercanos a los lugares de donde puede provenir agua.
- El sensor de humo se ubicó en la mitad del tumbado para lograr una mejor captación de emisiones de humo en todo el taller.

A continuación se muestran imágenes en donde se puede observar los lugares en donde se ubicaron los sensores y nodos.

Figura 2-80.- Ubicación del nodo de temperatura en el taller

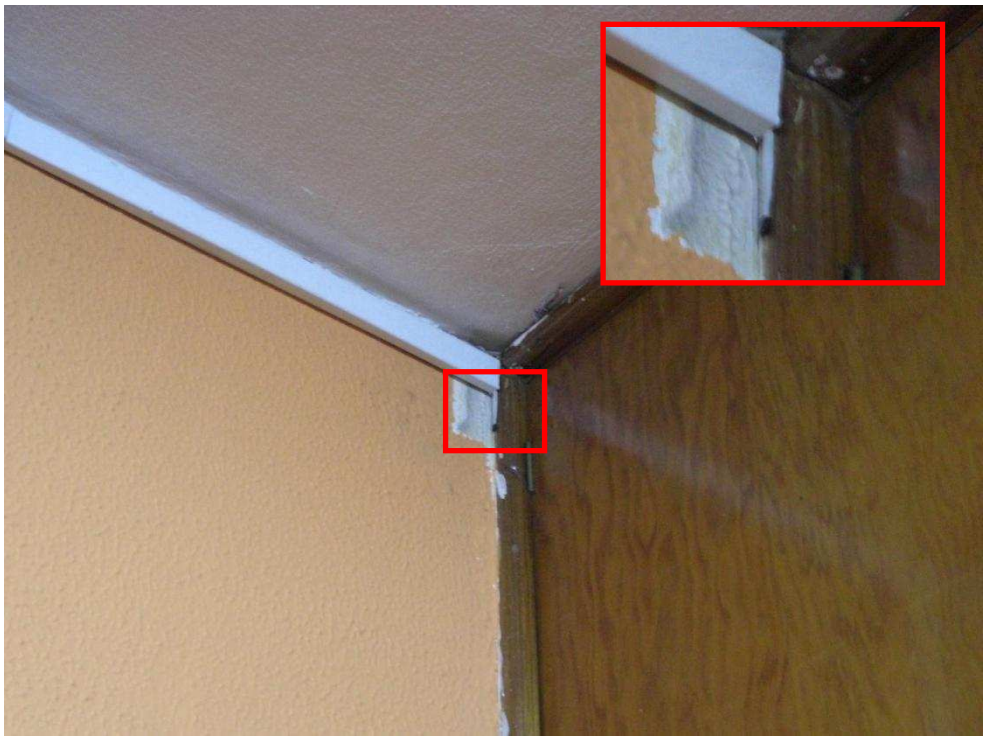


Figura 2-81.- Ubicación sensor de temperatura T3 en el taller

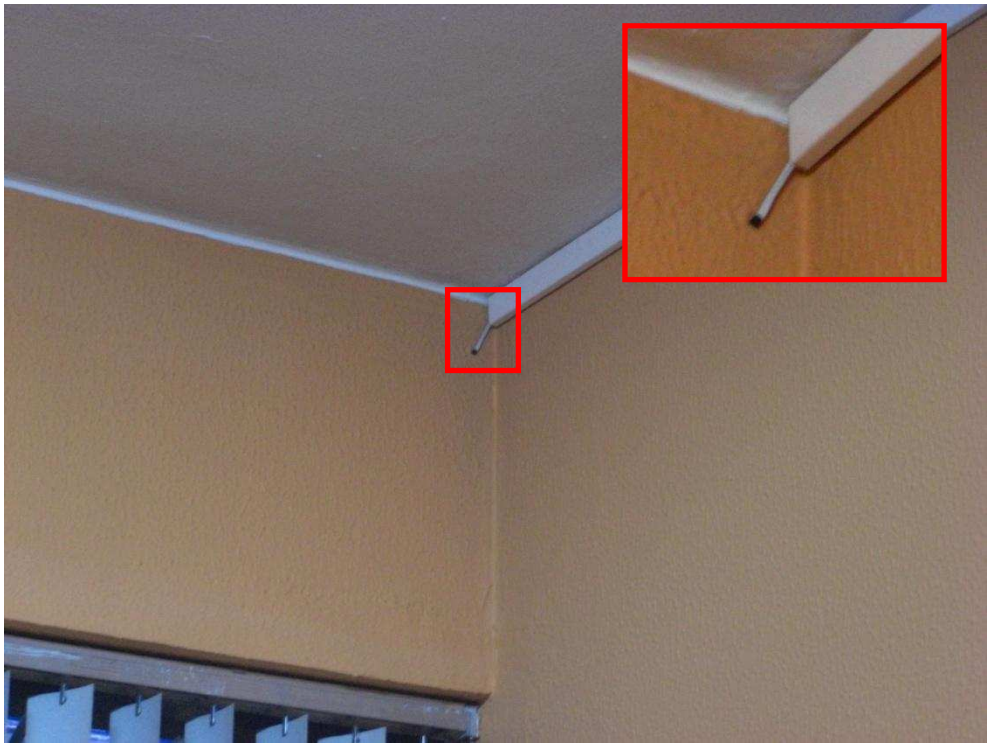


Figura 2-82.- Ubicación sensor de temperatura T4 en el taller

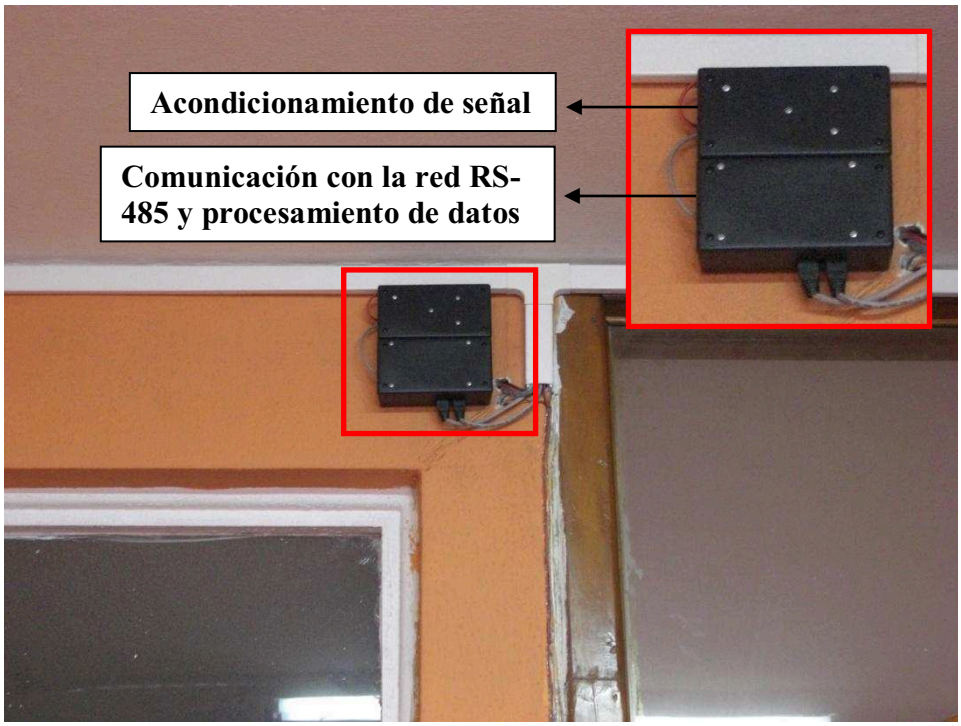


Figura 2-83.- Ubicación del nodo de presencia de agua en el taller

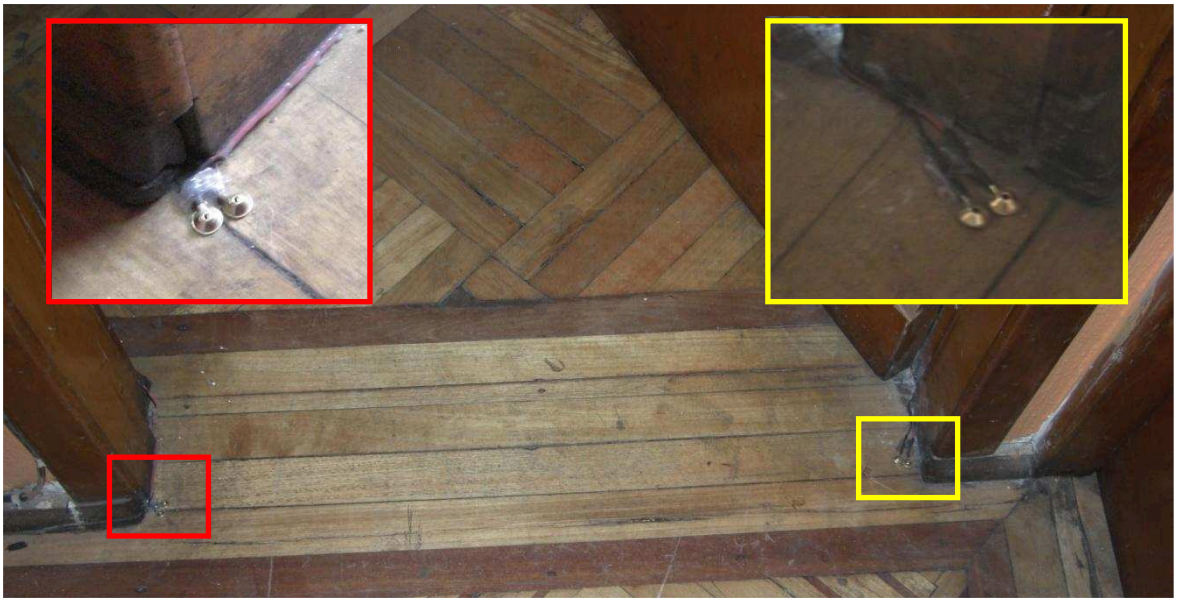


Figura 2-84.- Ubicación de electrodos en el área del taller

Figura 2-85.- Ubicación del detector de humo en el taller

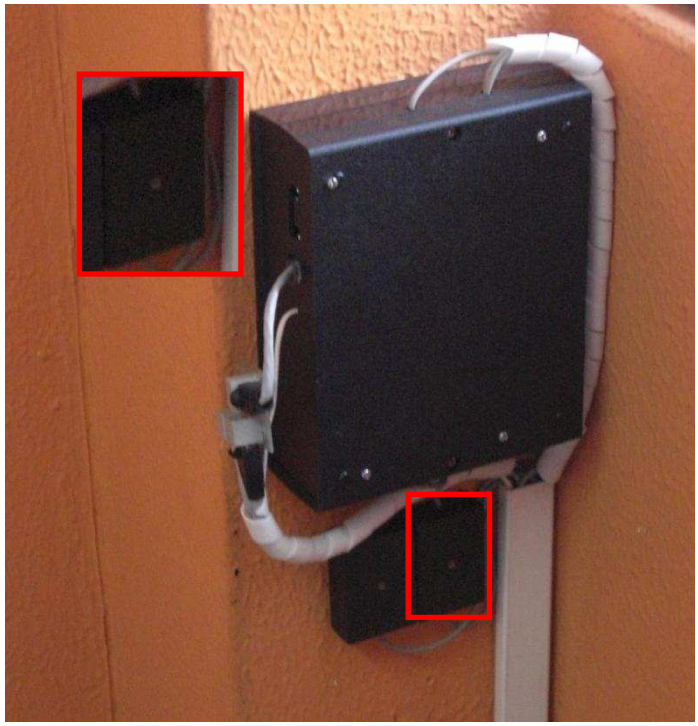


Figura 2-86.- Ubicación del circuito de acondicionamiento del detector de humo para el taller

2.7.3 UBICACIÓN DEL NODO DE CORRIENTE

El circuito de adquisición de datos y comunicación con la red RS-485, el circuito de acondicionamiento de señal y la fuente de alimentación para el sensor de corriente se ubicaron en la bodega de las instalaciones ya que esta se encuentra cerca del tablero de breakers donde se ubicó el sensor de corriente.

A continuación se muestra una imagen en donde se puede observar la ubicación de los circuitos antes mencionados, en la bodega.

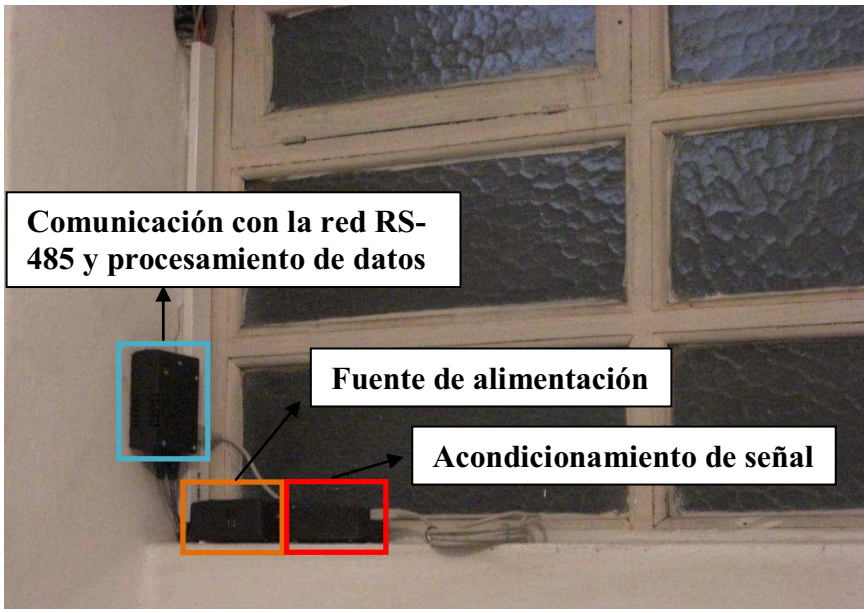


Figura 2-87.- Ubicación del nodo de corriente

En la siguiente imagen se observa la ubicación del sensor de corriente dentro del tablero de breakers:

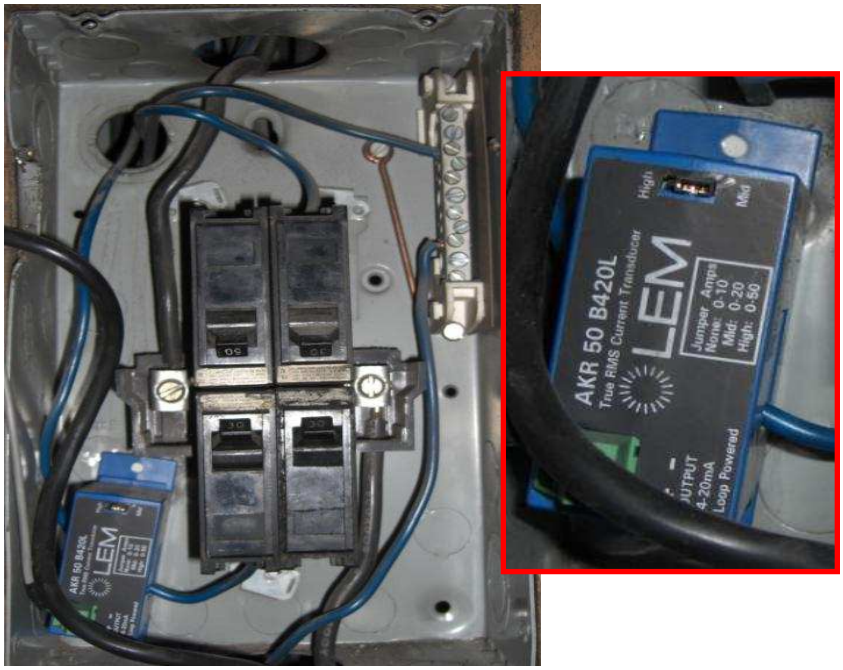


Figura 2-88.- Ubicación del sensor de corriente

2.7.4 UBICACIÓN DEL SENSOR MAGNÉTICO

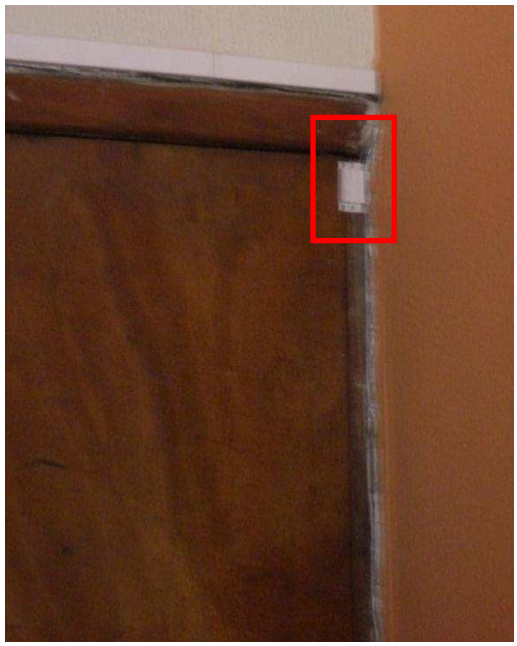


Figura 2-89.- Ubicación del sensor magnético

El sensor magnético se ubicó de tal forma que se pueda detectar desde que la puerta empieza a abrirse. Así, no será necesario que la puerta se abra en su totalidad para que el sistema comunique este estado.

2.7.5 ACTIVACIÓN DE CERRADURA ELÉCTRICA.

Para accionar la cerradura eléctrica de la puerta por medio del nodo maestro, sea por petición por medio del celular o del computador, se hizo una conexión en paralelo al switch de la misma como se muestra en la siguiente figura:

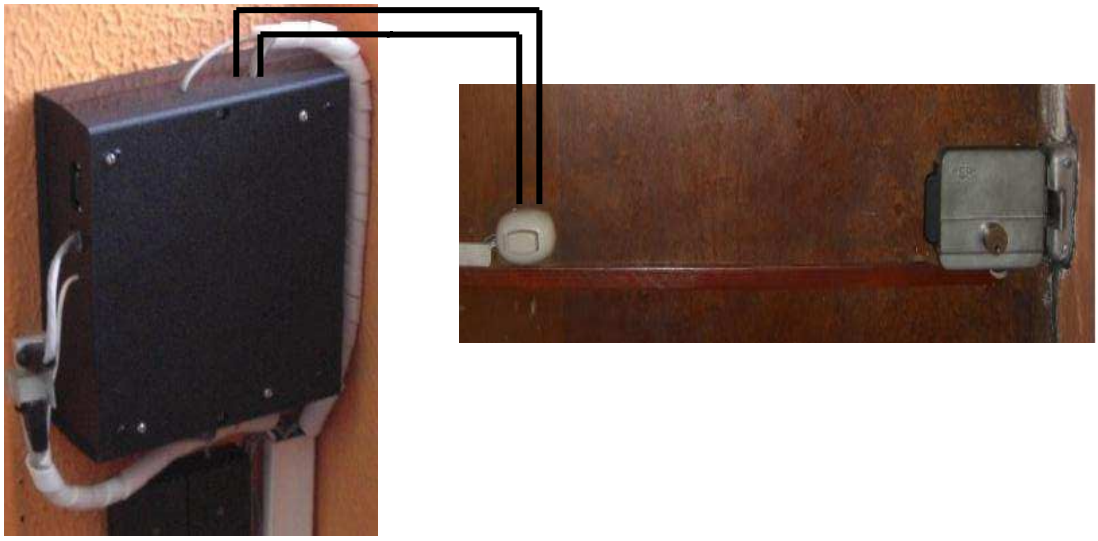


Figura 2-89.- Accionamiento de la cerradura eléctrica

CAPÍTULO 3
DISEÑO DE SORTWARE

3.1 INTRODUCCIÓN

Para lograr todos los objetivos propuestos en este proyecto se empleó: una aplicación para computador, una aplicación para celular y cuatro programas para los nodos: maestro, temperatura, corriente y de presencia de agua.

En los siguientes numerales se expone el funcionamiento de dichas aplicaciones y las diferentes tareas que se realizan con los programas de los nodos. Para ver todos los programas completos, referirse al Anexo B.

3.2 DESARROLLO DE LA APLICACIÓN DE LA INTERFAZ ESTACIONARIA

La aplicación para computador fue desarrollada mediante el lenguaje de programación Visual C#, la cual en resumen se encarga de las siguientes tareas:

- Comunicación con el nodo maestro para adquirir información acerca de los valores de temperatura y corriente; del estado de los detectores de humo, agua y apertura de puerta.
- Determinación de alertas, envío de mensajes de texto y mail.
- Permitir a los usuarios registrados configurar ciertos parámetros de acuerdo al privilegio que se les haya otorgado.
- Guardar en la base de datos todas las configuraciones realizadas.
- Comunicación con un celular vía Bluetooth, para envío de información de los sensores y/o comandar la apertura de la puerta.
- Señalización de alarmas.

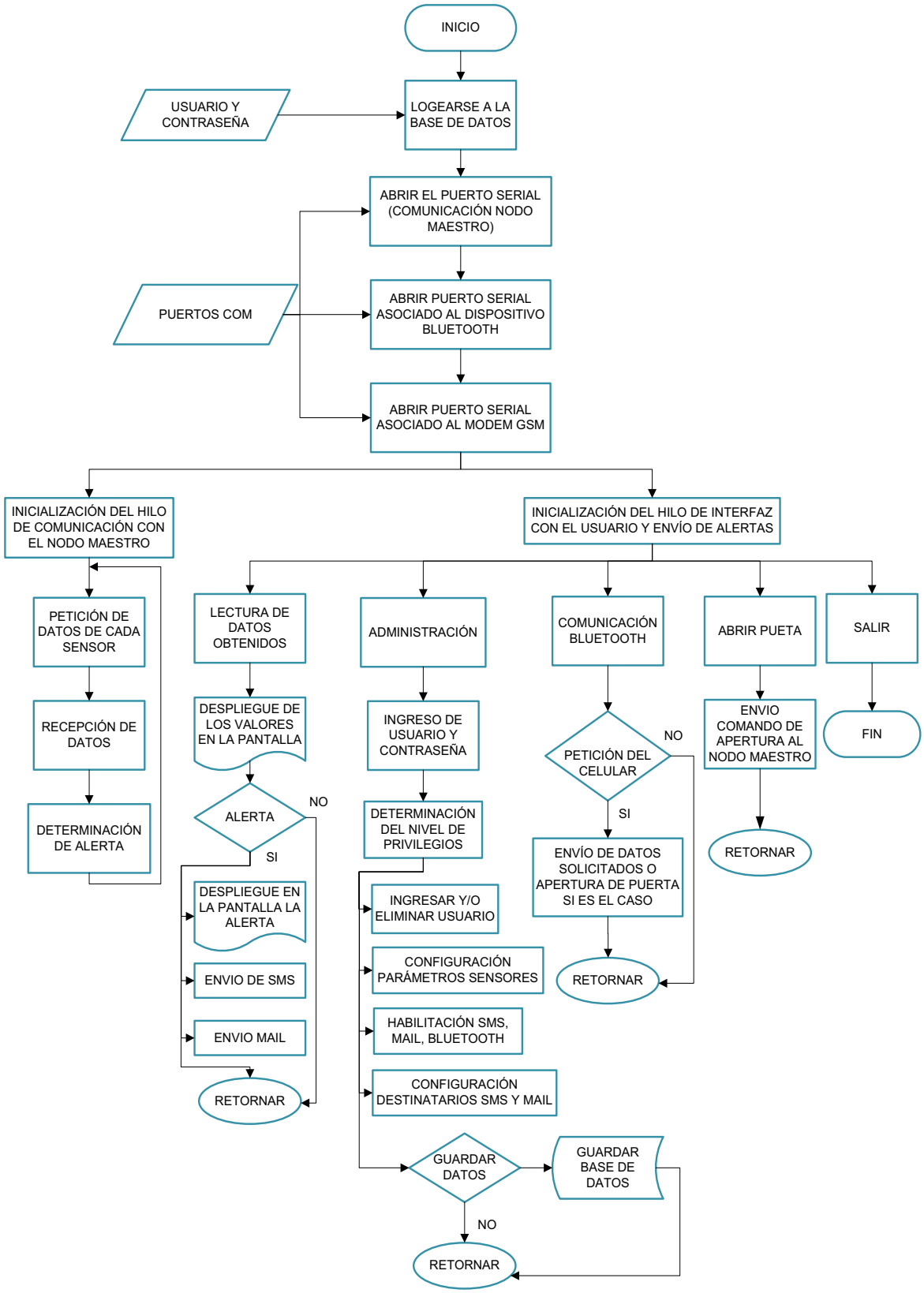


Figura 3-1.- Diagrama de flujo de la aplicación para computador

En el diagrama de flujo de la figura 3-1, se puede observar la lógica que sigue el programa de esta aplicación para realizar todas las tareas mencionadas.

A continuación se presenta en detalle cada una de las acciones que se llevan a cabo en la aplicación.

3.2.1 ACCESO A RECURSOS NECESARIOS

Para cumplir con todas las funciones establecidas, se requiere abrir un puerto de comunicación serial para enlazarse al nodo maestro, otro para comunicación vía Bluetooth con un celular y un tercero para comandar un modem GSM que se emplea para envío de mensajes de texto. La siguiente ventana "Configuración" permite seleccionar dichos puertos:

Figura 3-2.- Ventana de configuración de puertos

En caso de que no se haya determinado algún puerto en la ventana anterior, se muestra en la pantalla el siguiente mensaje de error:

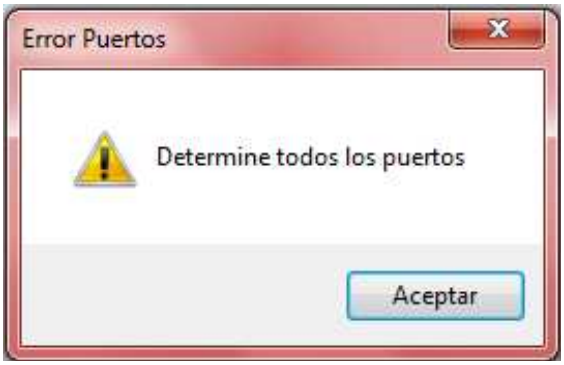


Figura 3-3.- Ventana de error de configuración de puertos

Determinados los puertos, la aplicación inicializa la base de datos realizando tres conexiones independientes, para cada una de las cuales se necesita un nombre se usuario y contraseña. Se emplean estas tres conexiones ya que se da el caso en el que la funciones a la que está dedicada cada una de ellas se realizan al mismo tiempo y de no haber una conexión independiente surgiría un error en la aplicación. La primera conexión se emplea para guardar en la base de datos los cambios realizados en cuanto a parámetros de sensores, usuarios, conectividad, entre otros, como se verá posteriormente. En caso de no poder hacer esta conexión se muestra la siguiente ventana de error en la pantalla:

Figura 3-4.- Ventana de error en la primera conexión a la base de datos

La segunda conexión se utiliza para acceder a la base de datos y tomar la información acerca de destinatarios para envío de mensajes de texto y mail. Si no se puede realizar la conexión, se muestra la siguiente ventana de error:

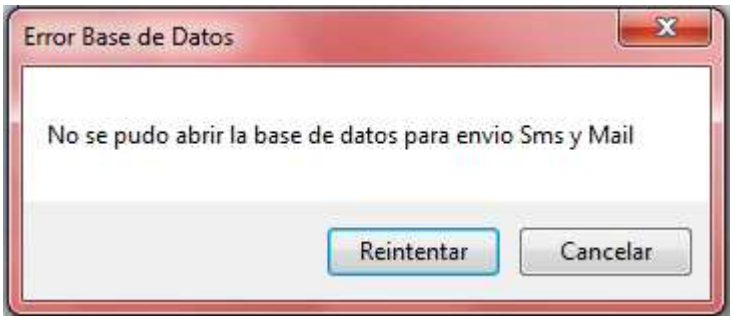


Figura 3-5.- Ventana de error en la segunda conexión a la base de datos

En la aplicación se creó un segundo hilo, el cual tiene la función de ejecutar líneas de programa al mismo tiempo que lo hace el hilo principal. Este segundo hilo se encarga de pedir información de las distintas variables sensadas al nodo maestro y levantar algunas banderas indicando si se ha suscitado una alerta. Así, la tercera conexión a la base de datos se emplea en este hilo para adquirir los datos de los límites de corriente y temperatura, los cuales serán comparados con la información actual para determinar la existencia de alertas. En caso de no poder realizar dicha conexión se muestra la siguiente ventana de error:

Figura 3-6.- Ventana de error en la tercera conexión a la base de datos

A continuación, en la aplicación se ejecutan comandos para abrir cada uno de los puertos antes determinados. Sí surge un error en la apertura del puerto para comunicación con el nodo maestro, se despliega la siguiente ventana:

Figura 3-7.- Ventana de error en la comunicación con nodo maestro

En caso de surgir un error al abrir el puerto para envío de mensajes de texto, se abre la ventana de error que se muestra a continuación:

Figura 3-8.- Ventana de error en la apertura del puerto para envío de SMS

Si el puerto para comunicación Bluetooth no se puede abrir, aparece la siguiente ventana:

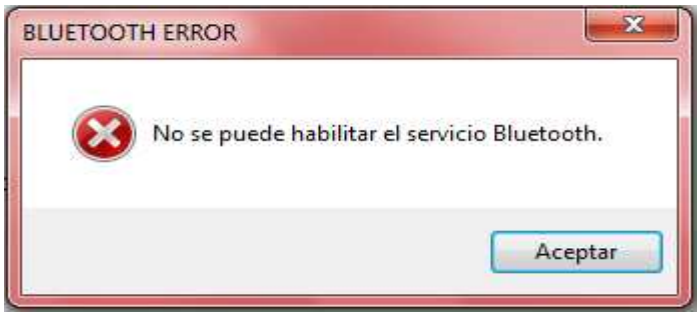


Figura 3-9.- Ventana de error en la apertura del puerto para comunicación vía Bluetooth

La siguiente figura muestra la ventana principal de la aplicación cuando no se ha podido establecer comunicación con el nodo maestro:

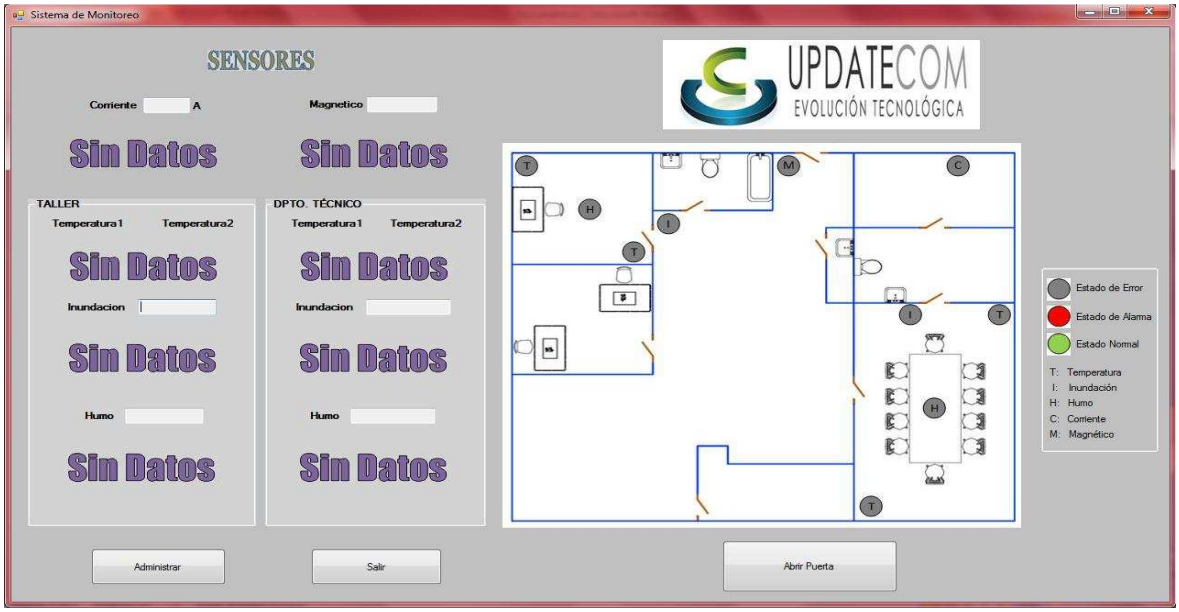


Figura 3-10.- Ventana principal sin comunicación con el nodo maestro

Si la comunicación con el nodo maestro ha sido exitosa, la ventana principal luce como la siguiente después de haber adquirido los datos de cada nodo esclavo y de los detectores de humo, agua y apertura de puerta.

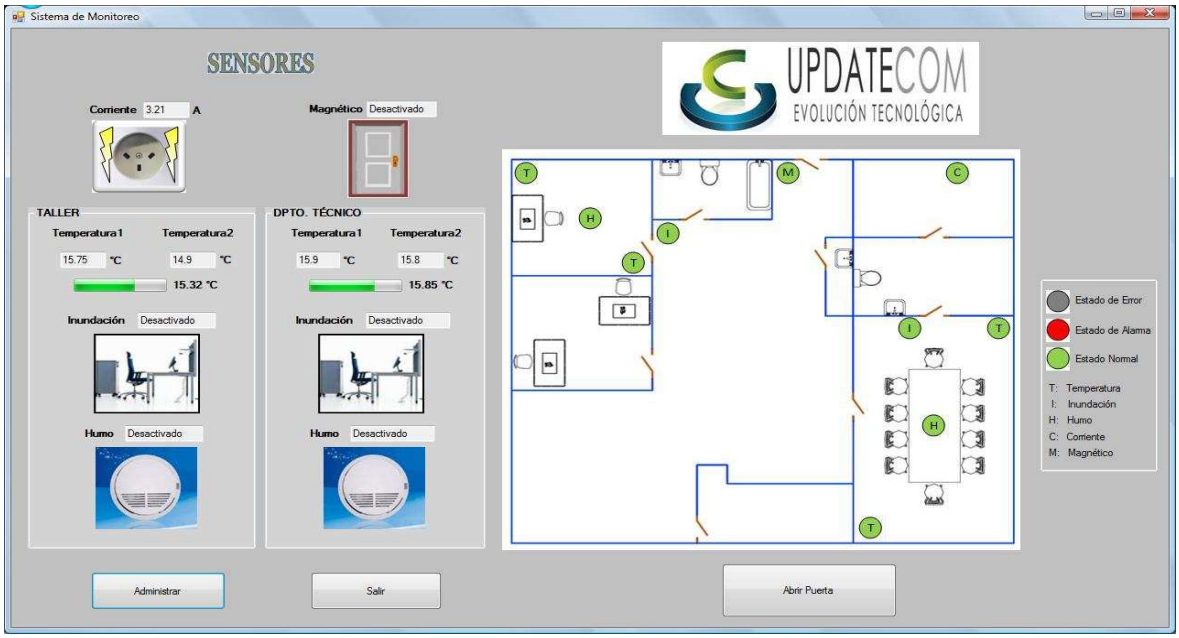


Figura 3-11.- Ventana principal una vez establecida la comunicación con el nodo maestro

Esta pantalla está conformada por:

- Cuadros de texto para visualizar los valores de temperatura en °C medidos por los distintos sensores ubicados en el departamento técnico y en el taller.
- Cuadros de texto que muestran el estado de los detectores de humo, presencia de agua y apertura de puerta.
- Iconos para señalización de alertas.
- Plano de las instalaciones en donde se observa la ubicación de cada sensor y en caso darse una alerta señaliza qué sensor emitió dicha alerta. También muestra los nodos que están fuera de la red y en caso de los sensores de temperatura indica que sensor está desconectado.
- Botón Administrar: Abre la ventana “Administrar” en la cual se pueden configurar diferentes aspectos de la aplicación.

- Botón Abrir Puerta: Comanda el accionamiento de un relé de la tarjeta del nodo maestro, el cual activa la cerradura eléctrica de la puerta de ingreso a las instalaciones de la empresa.
- Botón Salir: Culmina los procesos que está ejecutando la aplicación y cierra la ventana principal.

3.2.2 ADMINISTRACIÓN

Mediante el botón administrar de la ventana principal, los usuarios registrados pueden acceder a una serie de pestañas en las que se pueden hacer configuraciones, ingreso y eliminación de destinatarios y usuarios, como se verá en los siguientes numerales.

3.2.2.1 Login

Para acceder a la ventana “Administración” el usuario deberá ingresar utilizando su nombre de usuario y contraseña. La aplicación consulta la base de datos para verificar que la información ingresada corresponde a uno de los usuarios registrados, para así determinar su nivel de privilegio y permitir su acceso.

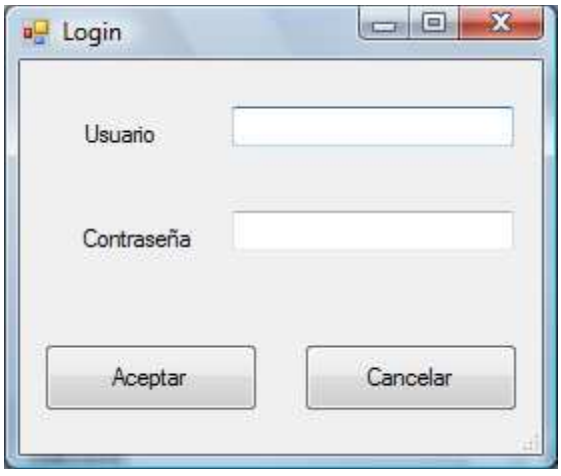


Figura 3-12.- Ventana Login para administración de la aplicación

Se dispusieron ayudas para el usuario como las que se muestran en la siguiente figura:

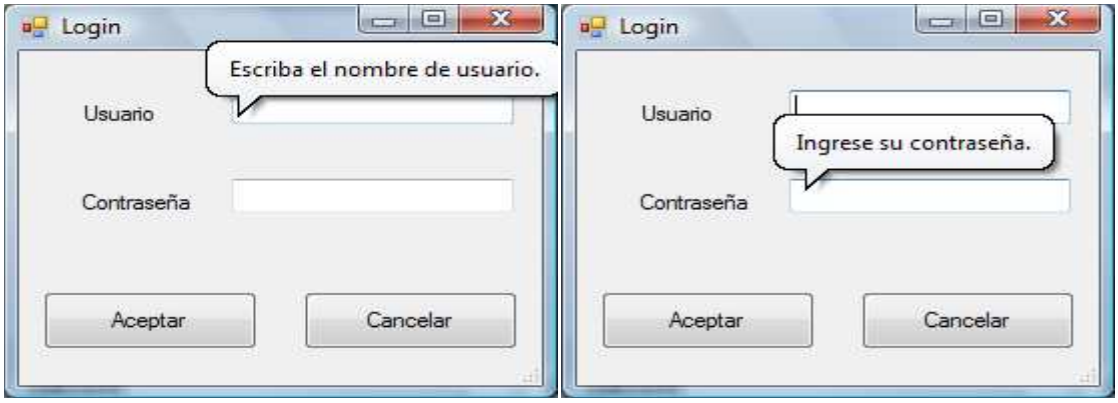


Figura 3-13.- Ayudas en la ventana Login

Si alguno de los campos pedidos no es completado o no es un usuario registrado, se despliega el siguiente mensaje de error:

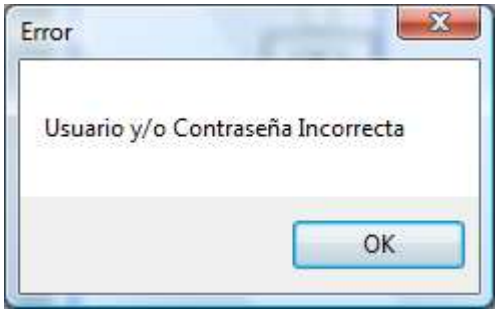


Figura 3-14.- Error de usuario y/o contraseña

3.2.2.2 Parámetros

Al ingresar las credenciales de identificación correctamente, se despliega la ventana “Administrar”, la cual está conformada por una serie de pestañas, cada

una con una función específica. La primera es la pestaña “Parámetros”, en la que se puede configurar los límites máximo y mínimo de corriente y los valores máximos de temperatura en el departamento técnico y taller. Estos parámetros servirán como referencia para determinar las diferentes alertas, por tanto es información muy importante que sólo un usuario con nivel de privilegio alto puede modificar. En dicha pestaña también se puede habilitar o deshabilitar el envío de alertas por apertura de la puerta de las instalaciones, ya que durante el tiempo en el que haya actividad en la empresa, el envío de esta alerta deberá ser deshabilitada. En la siguiente figura se puede visualizar el diseño para esta pantalla.

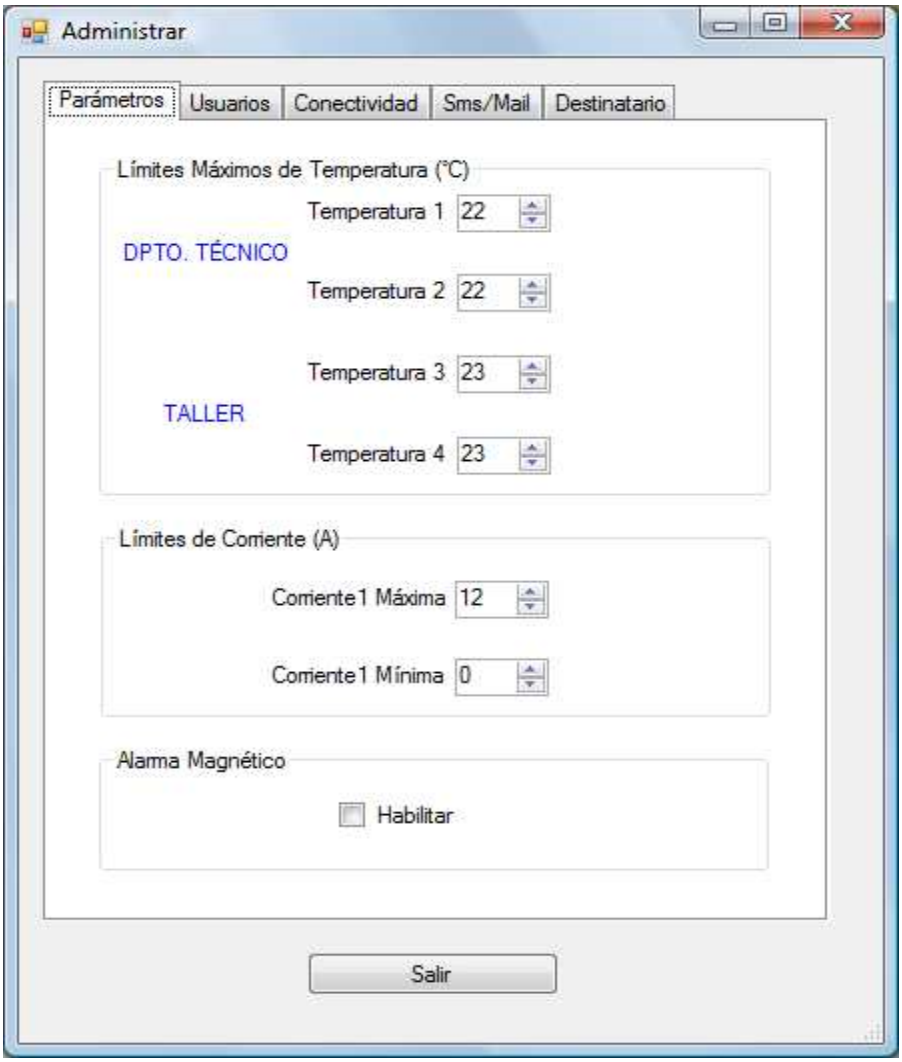


Figura 3-15.- Pestaña Parámetros de la ventana Administración

3.2.2.3 Usuarios

La pestaña “Usuarios” de la ventana “Administrar” luce como se muestra en la siguiente figura. En ella se puede registrar y eliminar usuarios, como se verá posteriormente.

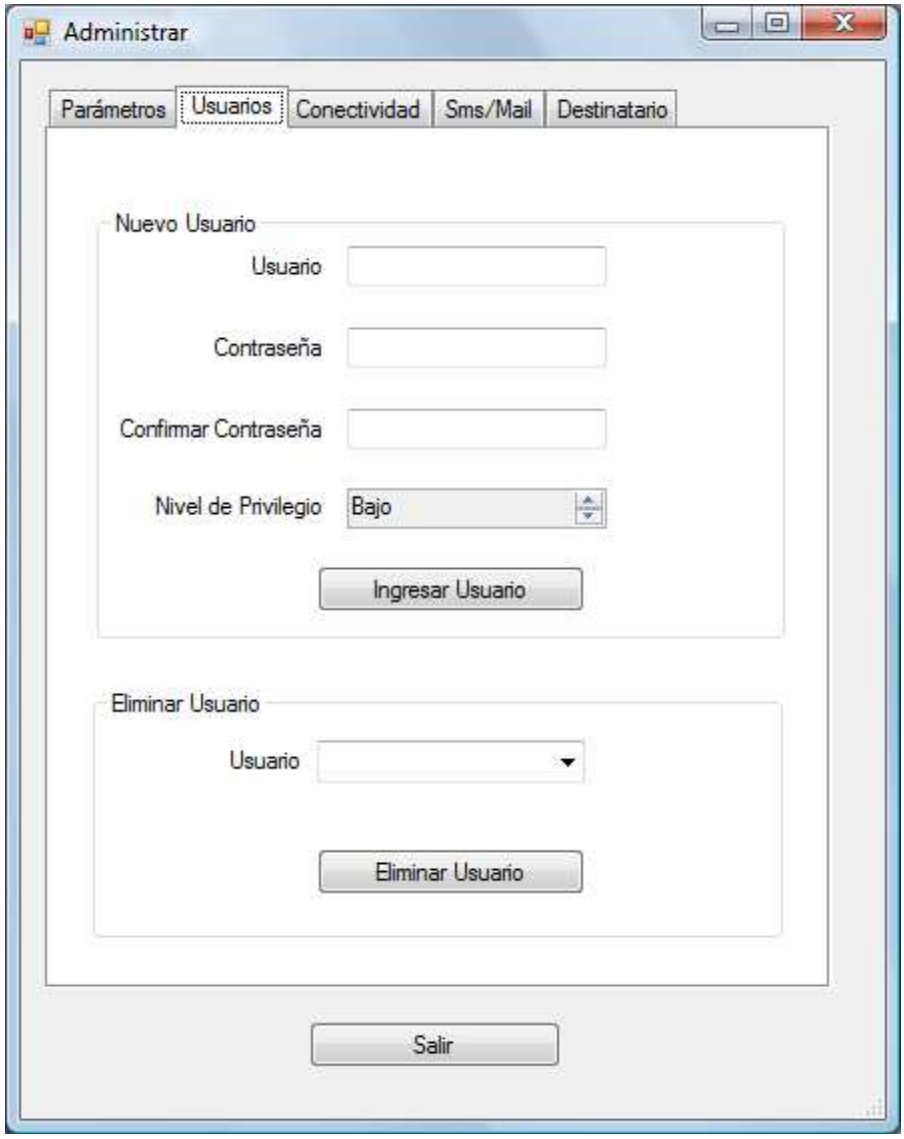


Figura 3-16.- Pestaña Usuarios de la ventana Administración

3.2.2.3.1 Registro de usuarios

Para registrar un nuevo usuario se deben completar los campos usuario, contraseña, confirmar contraseña y nivel de privilegio de la sección “Nuevo usuario” que se muestra en la figura 3-16. Para esto se disponen de ayudas como las que se muestra a continuación:

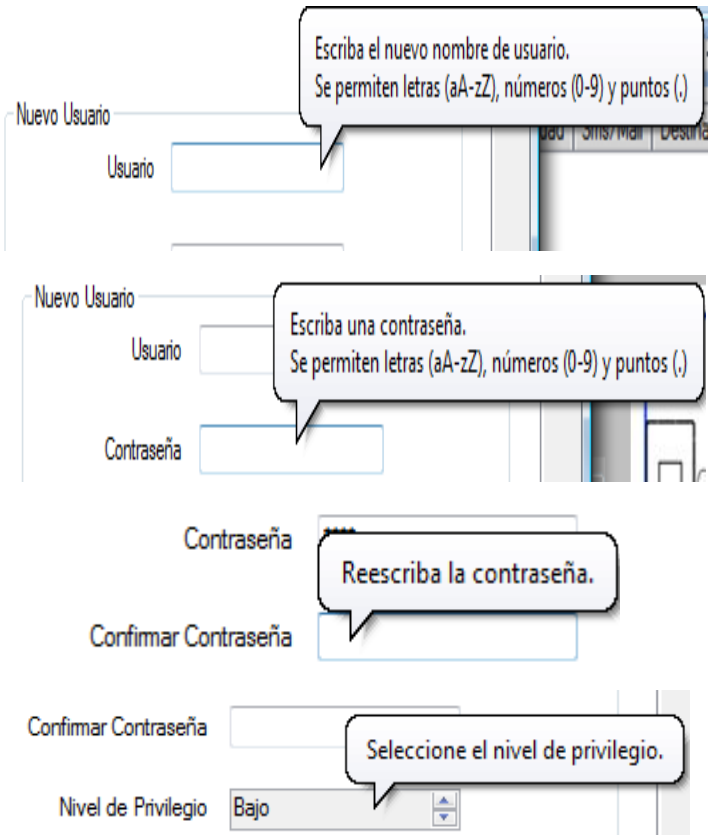


Figura 3-17.- Mensajes de ayuda para registro de usuarios

La aplicación verifica si los datos ingresados son válidos. En caso de no serlos, se muestra un signo de admiración con el correspondiente mensaje de error a la derecha del casillero que contiene el dato inválido, como se observa a continuación:

Figura 3-18.- Indicadores de datos inválidos en nuevo usuario

Al dar clic en el botón “Ingresar Usuario”, si la contraseña ingresada no coincide con su confirmación, se desplegará la siguiente ventana de error:

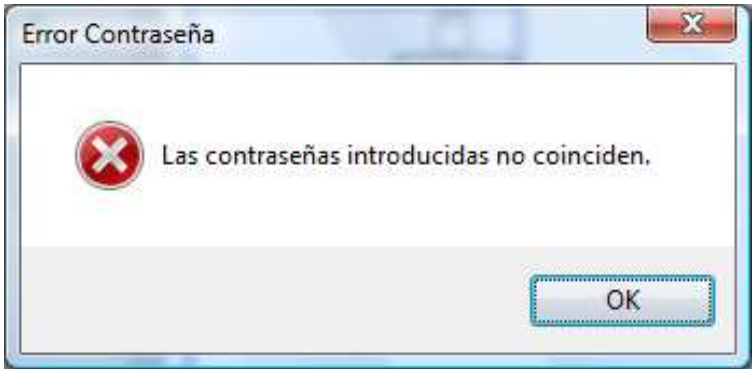


Figura 3-19.- Mensaje de error por no coincidencia de contraseñas

Si no se ingresaron todos los datos requeridos, se mostrará la ventana que sigue:

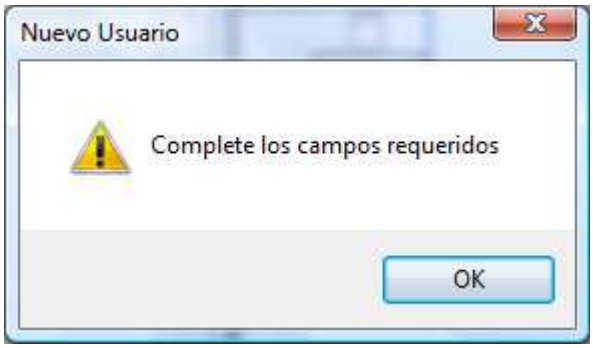


Figura 3-20.- Mensaje que alerta la ausencia de datos requeridos

En tanto, si toda la información fue ingresada correctamente, se verá en la pantalla la ventana que se muestra a continuación:

Figura 3-21.- Mensaje de ingreso exitoso de un nuevo usuario

Un usuario con nivel de privilegio alto tiene acceso a todas las configuraciones y acciones que pueden realizarse en la pantalla de administración. Mientras que, un usuario con nivel de privilegio bajo tiene acceso a habilitar o deshabilitar el envío de alertas por correo electrónico, modificar el texto que aparecerá en los mensajes de texto y correo electrónico, seleccionar los destinatarios mail y a habilitar o deshabilitar el envío de alerta cuando la puerta se abra, como se irá viendo en los siguientes apartados.

3.2.2.3.2 Eliminación de usuarios

De igual forma en la sección “Eliminar Usuario” se despliega un mensaje de ayuda que puede servir de guía para el usuario, como ve a continuación:

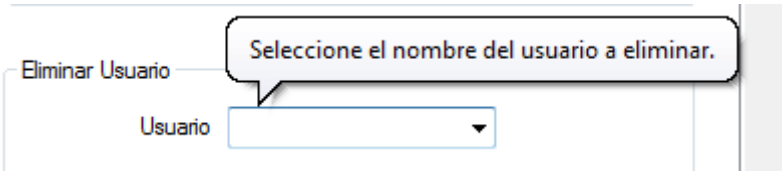


Figura 3-22.- Mensaje de ayuda para eliminación de usuarios

Para asegurarse que realmente se quiere eliminar un usuario se muestra la ventana que sigue:

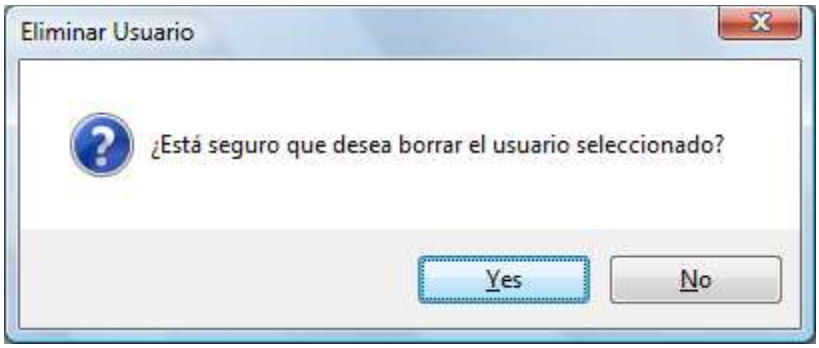


Figura 3-23.- Confirmación de eliminación de usuario

Si la respuesta es sí, el usuario se eliminará, pero si es no, no se hará ningún cambio y se volverá a la ventana “Administrar”.

3.2.2.4 Conectividad

En la pestaña conectividad se puede habilitar o deshabilitar los siguientes servicios:

- Envío de mensajes de texto
- Envío de correo electrónico
- Comunicación vía Bluetooth

El contenido de esta pestaña se muestra a continuación:

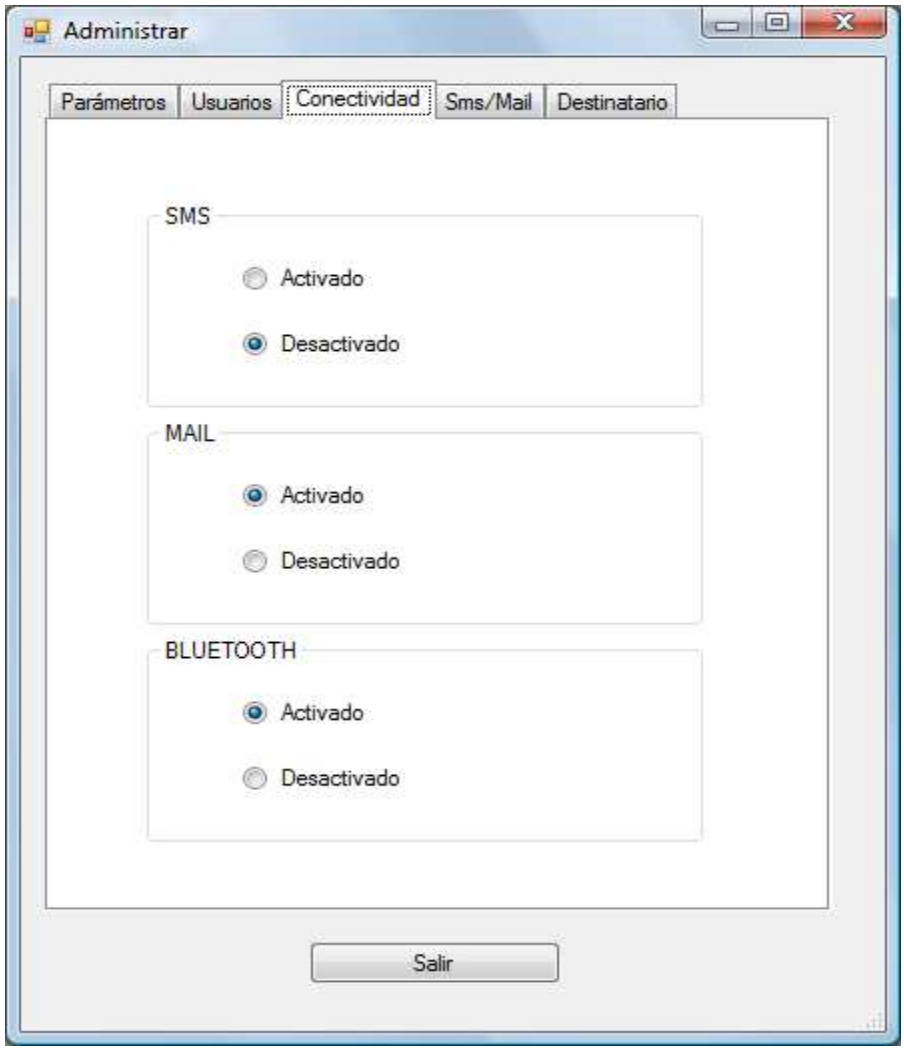


Figura 3-24.- Pestaña Conectividad de la ventana Administración

3.2.2.5 SMS/Mail

A cada mensaje de texto y correo electrónico se le puede añadir texto de encabezamiento antes del valor o estado de la variable en alerta. Así un usuario, con cualquier nivel de privilegio, puede ingresar un encabezado como el que se muestra en la siguiente figura:

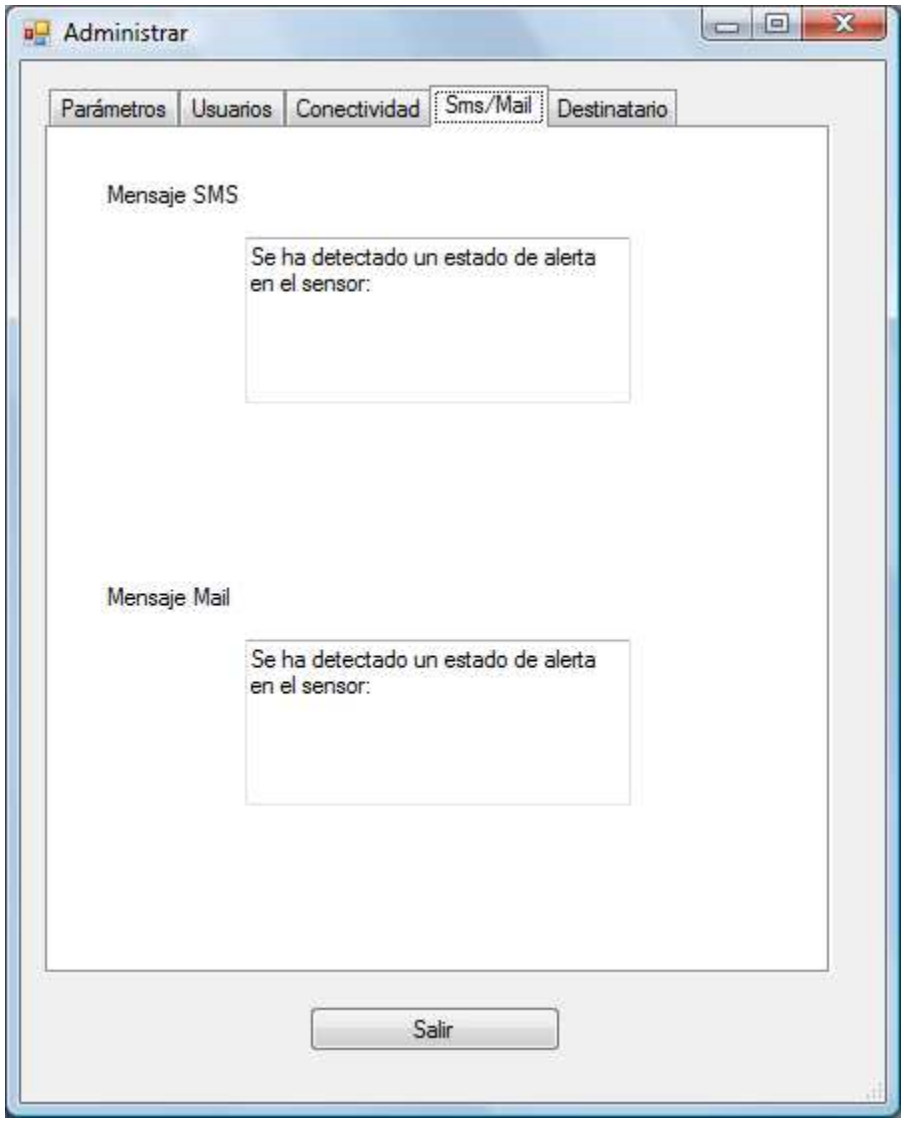


Figura 3-25.- Pestaña SMS/Mail de la ventana Administración

Como medio de ayuda para el usuario se disponen de los siguientes mensajes de guía:

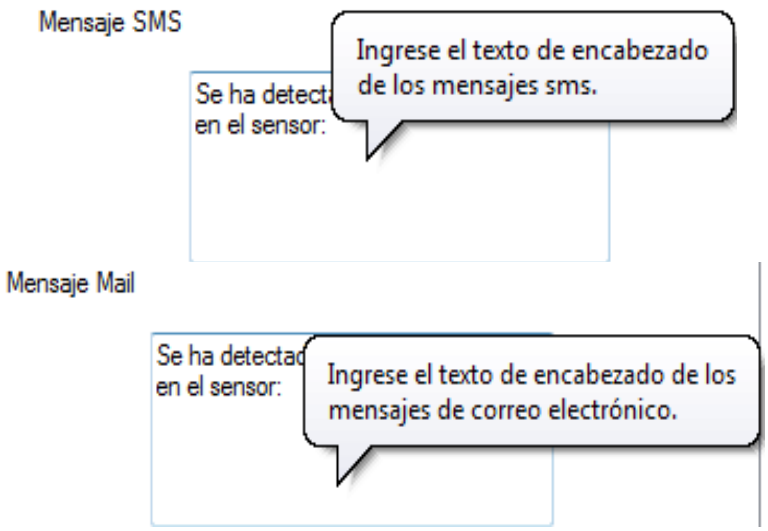


Figura 3-26.- Mensajes de ayuda de la pestaña SMS/Mail

3.2.2.6 Destinatarios

En la pestaña “Destinatario” se puede seleccionar los destinatarios a los cuales se desea enviar mensajes de texto y/o correo electrónico en caso de suscitarse una alerta. También, se puede registrar o eliminar un destinatario mediante los botones “Editar Destinatario SMS” y “Editar Destinatario Mail”.

Sólo un usuario con nivel de privilegio alto puede realizar cambios en la sección de “Envío SMS”. La figura siguiente muestra el diseño de esta pestaña:

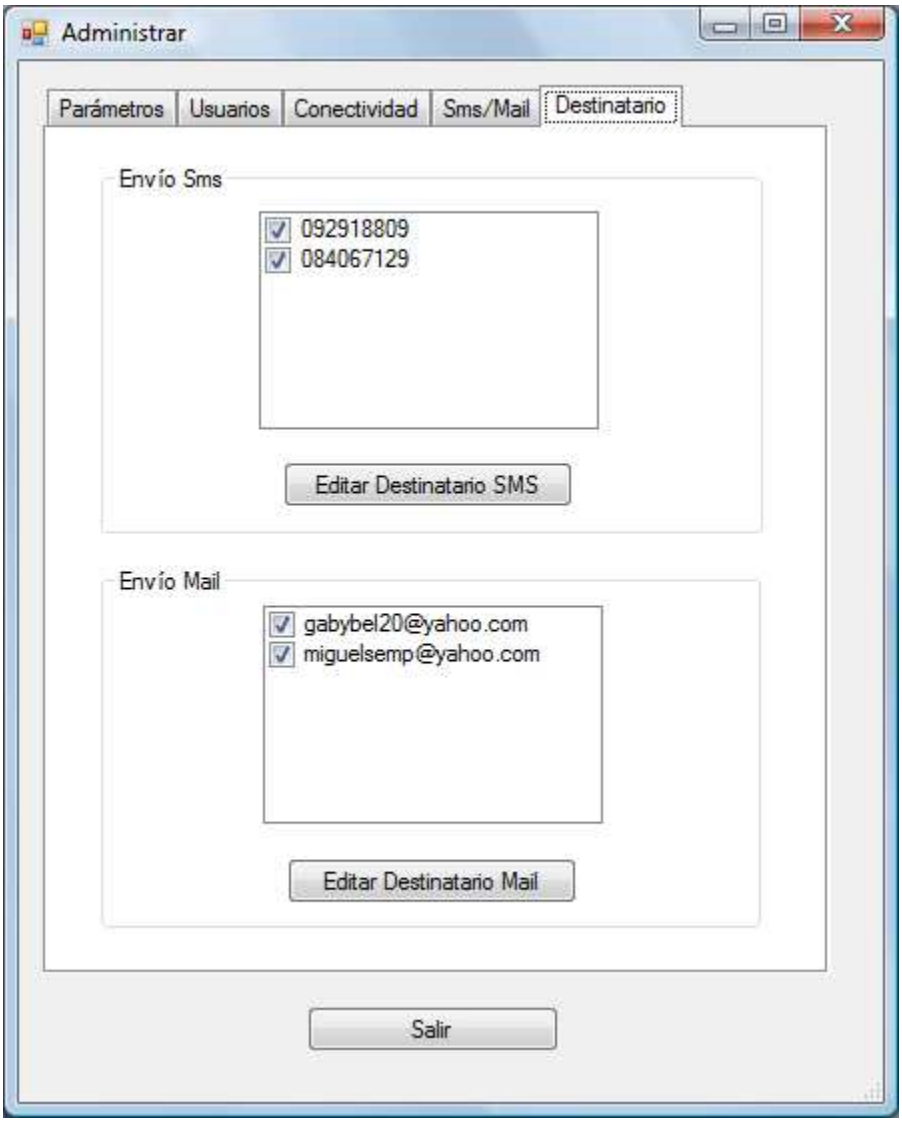


Figura 3-27.- Pestaña Destinatario de la ventana Administración

De igual forma que en las anteriores pestañas, se disponen de mensajes de ayuda para el usuario para que pueda saber con certeza lo que se espera que haga.

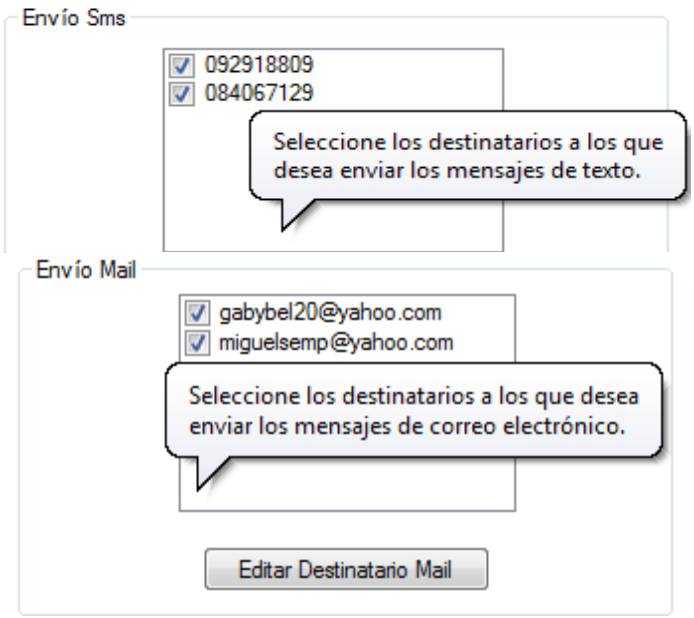


Figura 3-28.- Mensajes de ayuda de la pestaña Destinatario

3.2.2.6.1 Registro de destinatarios para envío de mensajes de texto

Al dar un clic en el botón “Editar Destinatario SMS”, se despliega la siguiente ventana, en la cual se deben llenar los campos nombre y número telefónico, para el registro de un nuevo destinatario para envío de mensajes de texto.

Figura 3-29.- Pestaña Agregar de la ventana Edición Destinatarios SMS

Esta ventana dispone de mensajes de guía para el usuario, que le comunican qué tipo de caracteres está permitido digitar, como los que se muestran en la siguiente figura.

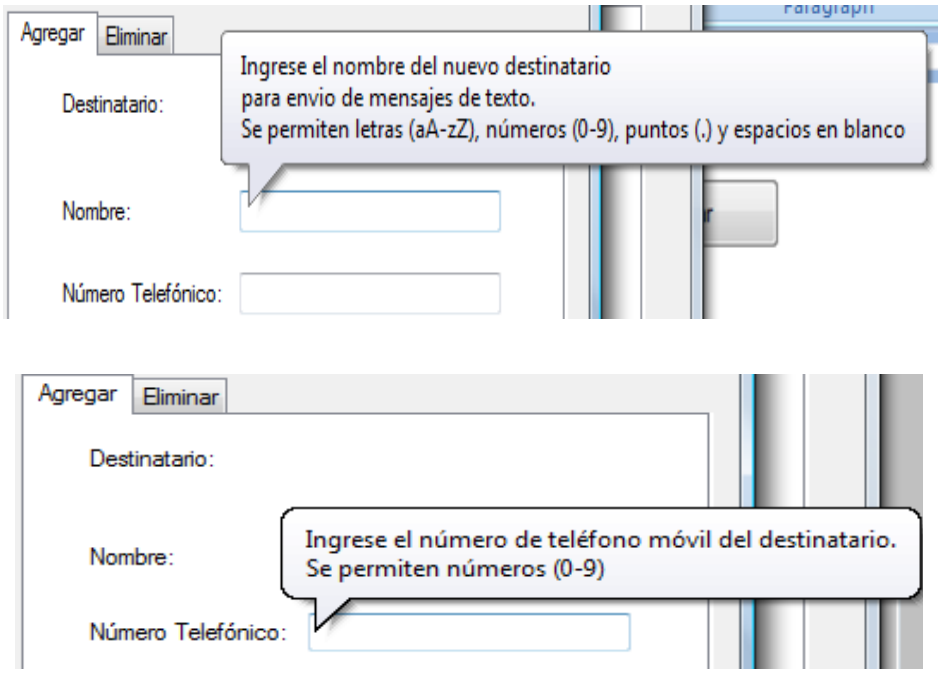


Figura 3-30.- Mensajes de ayuda de la pestaña Agregar de la ventana Editar Destinatario SMS

En el ingreso de nombre del usuario, se permiten letras mayúsculas y minúsculas, números del 0 al 9, puntos y espacios en blanco. En tanto para el número telefónico, como es de esperarse, se permiten únicamente números.

En caso de no haber cumplido con las restricciones mencionadas anteriormente, se muestra un mensaje de alerta al lado derecho de la caja de texto que contiene el error.

Figura 3-31.- Indicadores de datos inválidos de la ventana Editar Destinatario SMS

Si todos los parámetros son ingresados correctamente, la ventana siguiente aparece después de dar clic en el botón “Agregar”

Figura 3-32.- Ventana de ingreso exitoso de un nuevo destinatario SMS

De lo contrario, si uno de los datos fue erróneo o no se completó algún campo, se despliega un mensaje de error.

3.2.2.6.2 Eliminar destinatarios para envío de mensajes de texto

En la pestaña “Eliminar” de la ventana “Editar Destinatario Sms” se puede retirar permanentemente de la base de datos un destinatario para envío de mensajes de texto. Para esto se debe seleccionar su nombre y en caso de que se haya

ingresado más de un número telefónico para dicho usuario, se deberá seleccionar el número que se desea eliminar. En la siguiente figura, se muestra el diseño de esta pestaña:

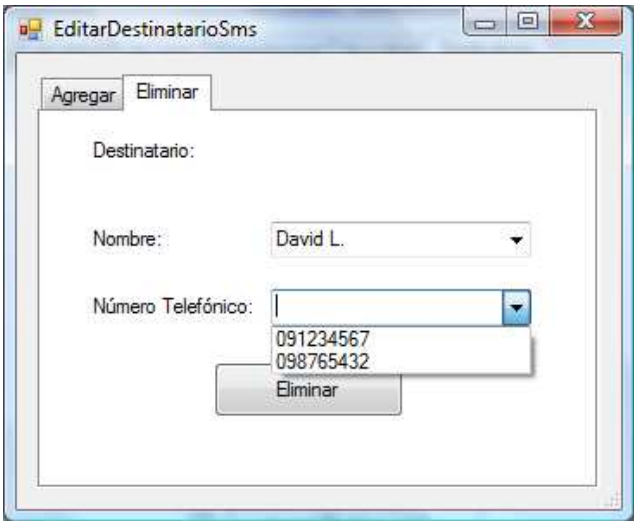


Figura 3-33.- Pestaña Eliminar de la ventana Edición Destinatario SMS

De igual forma que en las ventanas anteriores, se dispone de los siguientes mensajes de ayuda para el usuario:

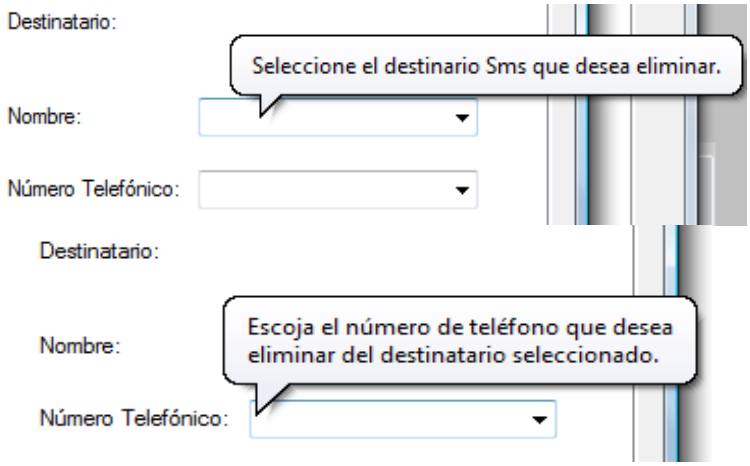


Figura 3-34.- Mensajes de ayuda para eliminar un destinatario SMS

Para confirmar la eliminación del destinatario seleccionado se empleó la siguiente ventana. Si se confirma, se borrará de la base de datos dicho destinatario.

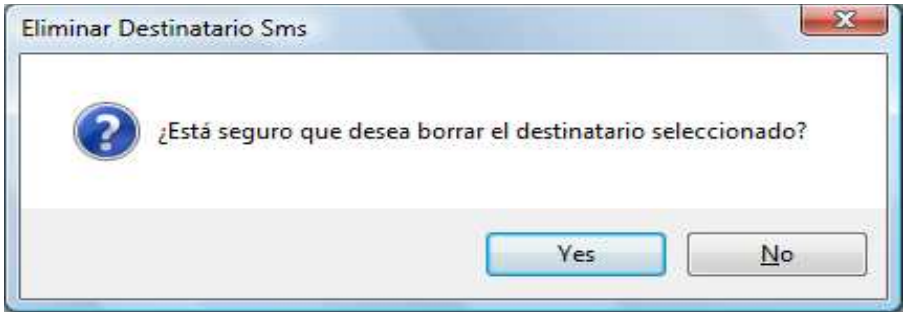


Figura 3-35.- Confirmación de eliminación de un destinatario SMS

3.2.2.6.3 Registro de destinatarios para envío de correo electrónico

Para agregar un nuevo destinatario de correo electrónico se requiere ingresar su nombre y su dirección de correo electrónico. En el caso de este último, únicamente se permiten letras minúsculas, números y puntos, que es lo que los servidores de correo electrónico admiten cuando se crean cuentas de correo.

Figura 3-36.- Pestaña Agregar de la ventana Editar Destinatario Mail

Se emplean los siguientes mensajes de ayuda para comunicar al usuario el tipo de datos que son permitidos en cada campo.

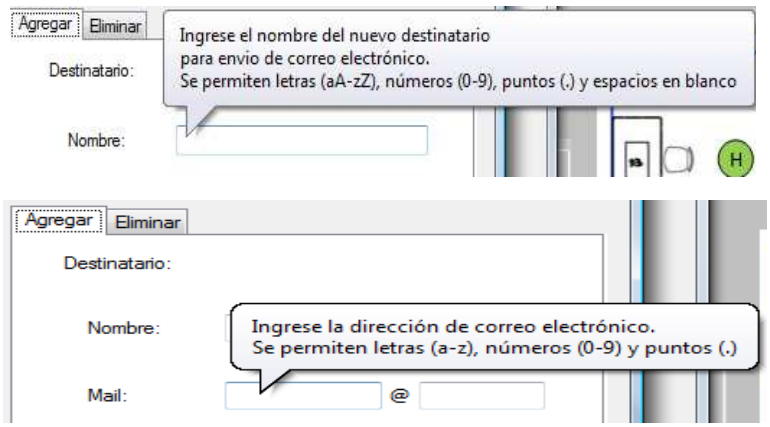


Figura 3-37.- Mensajes de ayuda para el registro de destinatarios de correo electrónico

Si los diferentes campos contienen información inválida, se despliega un símbolo de error y un mensaje que indica el por qué de dicho error, así como se muestra a continuación:

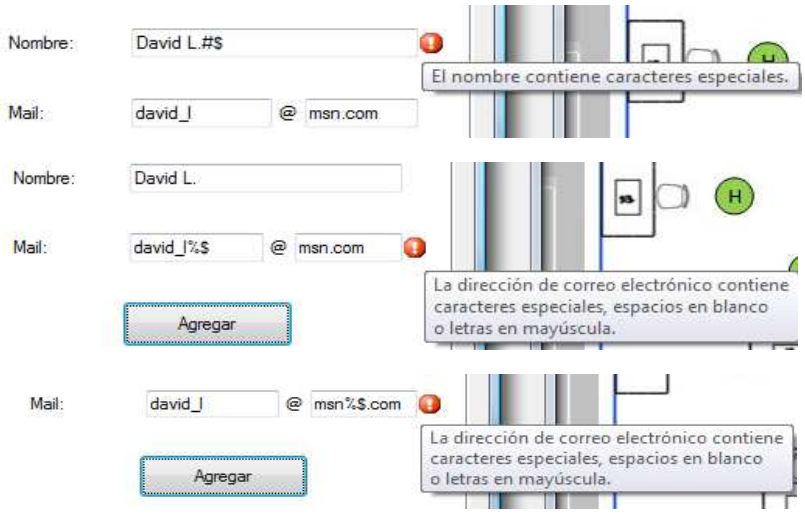


Figura 3-38.- Indicadores de datos inválidos en el registro de destinatarios de correo electrónico

Al dar clic en agregar, si todos los datos ingresados son correctos se mostrará un mensaje de ingreso exitoso del destinario, pero si existe algún error se desplegará un mensaje de alerta.

3.2.2.6.4 Eliminar destinatarios para envío de correo electrónico

De igual forma que en el caso de eliminación de destinatarios SMS, se debe seleccionar el nombre del destinatario a eliminar y su correo electrónico es caso de haber más de uno.

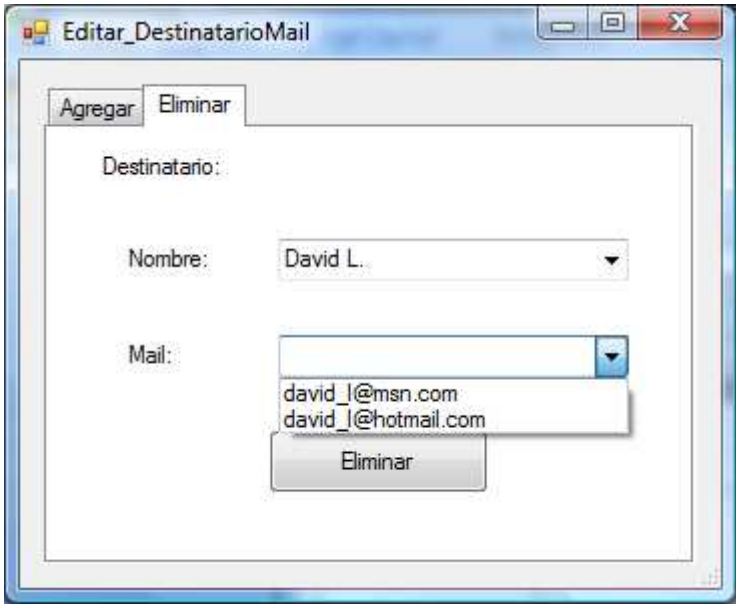


Figura 3-39.- Pestaña Eliminar de la ventana Editar Destinatario Mail

Similarmente, se incorporaron mensajes que sirvan de ayuda para ingresar los datos requeridos.

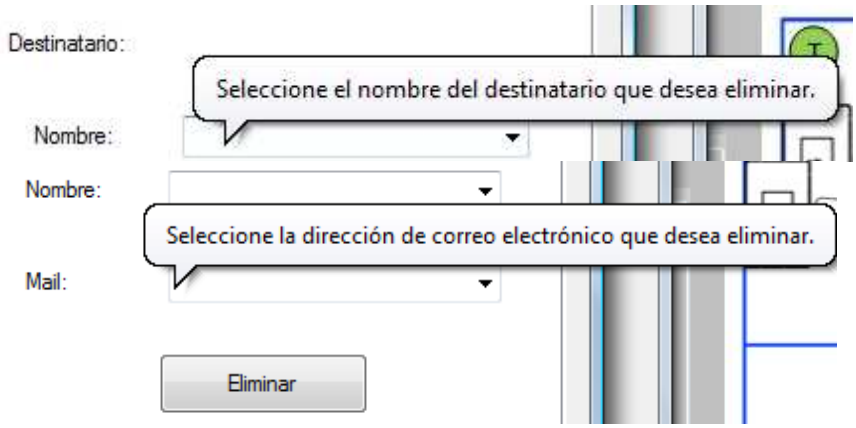


Figura 3-40.- Mensajes de ayuda para eliminar destinatarios de correo electrónico

Al dar clic en el botón “Eliminar”, se abrirá una ventana de confirmación igual que en el caso de eliminación de destinatarios SMS. Sí se ratifica esta decisión, el destinatario será eliminado. Caso contrario, se volverá a la ventana Editar Destinatario Mail sin realizar ninguna acción.

3.2.2.7 Guardar configuraciones

Al dar clic en el botón salir de la ventana de “Administración”, se despliega la ventana “Guardar Cambios” la cual realiza las siguientes acciones:

- Sí: Guarda los cambios realizados en la base de datos.
- No: Descarta los cambios y regresa a la ventana principal de la aplicación.
- Cancelar: Cierra la ventana “Guardar Cambios” y regresa a la ventana “Administración”.

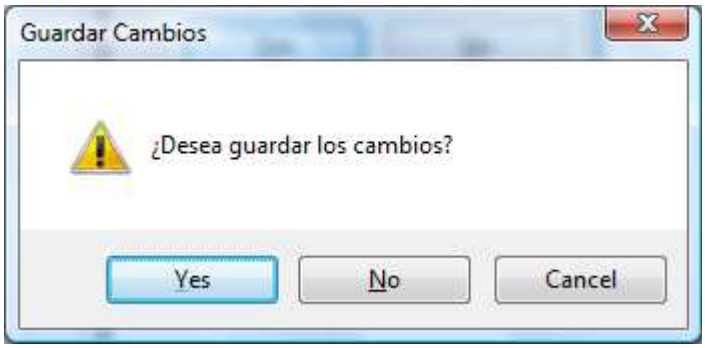


Figura 3-41.- Ventana Guardar Cambios

3.2.3 ALERTAS

Para la señalización de alertas en la ventana principal, se emplean los siguientes íconos:

Señalización de puerta cerrada

Señalización de puerta abierta

Señalización de subcorriente

Señalización de sobrecorriente

Señalización de nivel normal de corriente

Señalización de ausencia de agua

Señalización de presencia de agua

Señalización de ausencia de humo

Señalización de presencia de humo

3.3 DESARROLLO DE LA APLICACIÓN DE LA INTERFAZ MÓVIL

Después de realizar el diseño de la aplicación estacionaria se procedió a diseñar una aplicación a ser instalada en un dispositivo móvil, con el fin de realizar consultas remotas del estado del sistema SECURE SYS y de comandar, también de forma remota, la activación de la cerradura magnética.

Para el diseño de esta aplicación se empleó el lenguaje de programación JAVA en la plataforma Java Micro Edition dedicada al desarrollo de software para dispositivos con un número restringido de recursos, tal como se describe en el Capítulo 1.

3.3.1 DIAGRAMA DE FLUJO DE LA APLICACIÓN DE LA INTERFAZ MÓVIL

Para la creación de la aplicación que fue instalada en el dispositivo móvil (celular) se siguió la lógica expuesta en el diagrama de flujo de la figura siguiente:

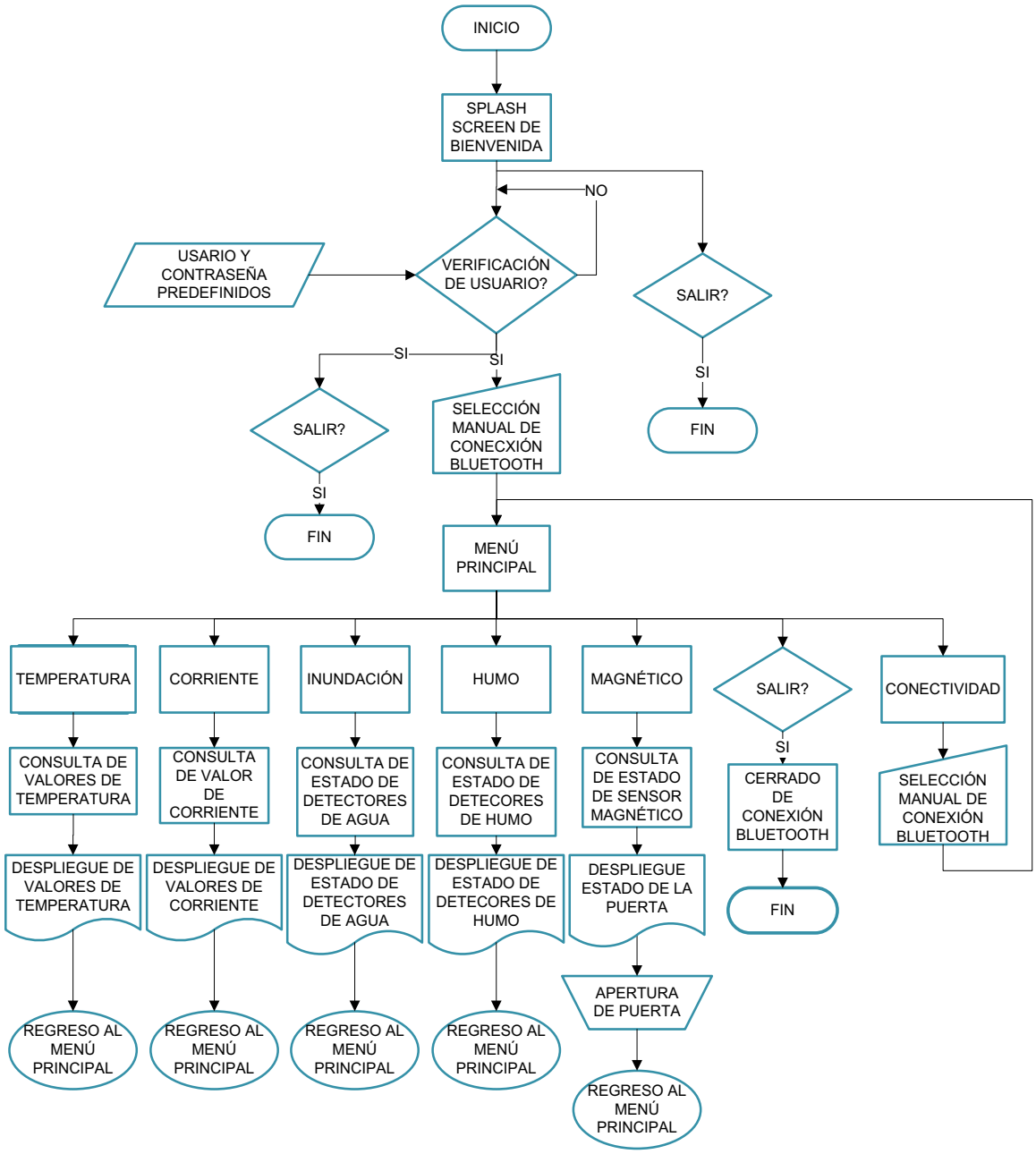


Figura 3-42.- Diagrama de flujo de la aplicación móvil

A continuación se detalla el funcionamiento de la aplicación para la interfaz móvil.

3.3.2 PANTALLA DE BIENVENIDA

Con la finalidad de hacer amigable a la aplicación mientras se realizan los procesos de carga y creación de variables al cargarse la aplicación, se diseñó y programó la siguiente pantalla de inicio.

Figura 3-43.- Pantalla de bienvenida – “Bienvenido”

Para la creación de esta pantalla se empleó el objeto SplashScreen, el cual permite desplegar una imagen por un período de tiempo programable, en este caso de 2 segundos.

3.3.3 SEGURIDAD DE LA APLICACIÓN

Para limitar el acceso a la aplicación, debido a que esta constituye una forma de acceso a las instalaciones de la empresa, se creó una pantalla de identificación

de usuario, la cual requiere el ingreso de un nombre de usuario y contraseña registrados. Esta pantalla fue programada a partir del objeto Form, agregándosele objetos Text Field para el ingreso de nombre de usuario y contraseña, además de un objeto Image para mejorar la apariencia de la pantalla.

Tras ingresar los datos requeridos el usuario debe dar clic o seleccionar el botón “Login”. Después de esta acción la aplicación se encarga de realizar una comparación de los datos ingresados con los se encuentran programados como credenciales de ingreso, y de dar paso a los siguientes procesos de la aplicación en caso de que los datos ingresados sean los correctos; caso contrario se borran los campos de la pantalla de identificación de usuario.

No existe un número determinado de iteraciones que limiten el número de intentos de ingreso de credenciales de identificación.

Para salir de la aplicación en caso de haber ingresado por error o de no recordar las credenciales de identificación se programó un botón de salida “Salir”, el cual inicia la tarea de terminación de todos los procesos y cerrado de la aplicación.

Figura 3-44.- Pantalla de identificación de usuario – “Login”

3.3.4 CONEXIÓN A LA INTERFAZ ESTACIONARIA

Si las credenciales de identificación ingresadas son correctas la aplicación carga la pantalla de conectividad. Esta pantalla tiene la función de permitir al usuario seleccionar la conexión Bluetooth correspondiente a la interfaz estacionaria.

Figura 3-45.- Pantalla “Conectividad”

La búsqueda de dispositivos Bluetooth dentro del rango de alcance de la interfaz móvil empieza automáticamente al cargase esta pantalla. El usuario tiene la opción de detener la búsqueda o salir de la aplicación desde esta pantalla empleando los botones configurados en la misma (Ver Figura 3-46).

Figura 3-46.- Opciones de pantalla “Conectividad”

Además, de no encontrarse el dispositivo deseado, la pantalla cuenta con la opción de actualizar, mediante la cual se comanda realizar nuevamente la búsqueda de los dispositivos que se encuentran al alcance. El botón “Actualizar” es reemplazado por el botón “Parar” durante el proceso de búsqueda, con el cual se puede detener dicho proceso (Ver Figura 3-47).

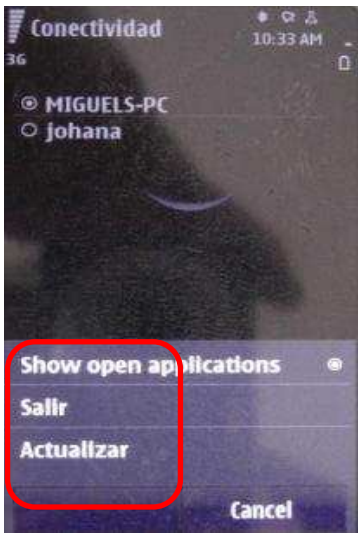


Figura 3-47.- Opciones de pantalla “Conectividad” – Botón Actualizar

Al seleccionar un dispositivo para realizar la conexión, la aplicación guarda el código de identificación único de este dispositivo para direccionar las posteriores peticiones de información.

3.3.5 MENÚ DE SENSORES

Tras haberse realizado la selección del dispositivo Bluetooth que sirve de enlace a la interfaz estacionaria, la aplicación carga la pantalla de menú de sensores. Esta pantalla se programa a partir del objeto Form al cual se le agregó un objeto de la clase ChoiceGroup al cual se le añadieron los elementos “Temperatura”, “Corriente”, “Inundacion”, “Humo” y “Magnetico”, que conforman el menú principal de la aplicación de la interfaz móvil.

Para realizar la consulta de los diferentes nodos del sistema el usuario debe seleccionar uno de los elementos del menú y dar clic en el botón “Aceptar” de esta pantalla.

Figura 3-48.- Menú principal – “Sensores”

Esta pantalla también tiene las opciones “Salir” y “Conectividad” (Ver Figura 3-49). La primera comanda el cerrado de la comunicación Bluetooth y la finalización de la aplicación. La segunda opción, permite al usuario volver a seleccionar el dispositivo Bluetooth al que se realizan las peticiones de información, abriendo nuevamente la pantalla “Conectividad”.

Figura 3-49.- Opciones del menú principal

A continuación se explican las acciones tomadas por la aplicación al seleccionarse cada una de las opciones del menú principal.

3.3.5.1 Opción “Temperatura”

Al ser seleccionada esta opción en el menú principal, se envía el comando de texto UDCST, a la interfaz estacionaria, para pedir los datos de temperatura tanto

del Departamento Técnico como del Taller. Tras esta acción, se abre la recepción de datos por el dispositivo Bluetooth hasta que se recibe la respuesta desde la interfaz estacionaria. Esta respuesta se almacena en un buffer de entrada y se lee entero por entero para obtener los valores de temperatura (Ver Figura 3-50).

Figura 3-50.- Pantalla “Temperatura”

En la pantalla “Temperatura” también se programó un botón “Regresar” para retornar al menú principal.

En caso de que se haya realizado una consulta de información previa y la conexión no se haya cerrado todavía, existirá un error en la comunicación; en este caso se desplegará un mensaje de error en la pantalla, como se muestra en la siguiente figura.

Figura 3-51.- Mensaje de error en la comunicación – Temperatura

Este mensaje de error aparece también en las pantallas “Sensor Corriente”, “Sensores Inundacion”, “Presencia de Humo” y “Puerta” bajo las mismas circunstancias.

3.3.5.2 Opción “Corriente”

Al seleccionarse esta opción la aplicación envía el comando UDCSC como petición del valor de corriente que alimenta a las instalaciones. Tras realizarse la petición, el dispositivo pasa a modo de recepción hasta que obtenga la respuesta de la interfaz estacionaria.

Del mismo modo que en la opción anterior, la respuesta obtenida de la interfaz estacionaria se almacena en un buffer del cual se obtiene el valor de corriente (Ver Figura 3-52).

Figura 3-52.- Pantalla “Sensor Corriente”

En la pantalla “Corriente” fue programado un botón “Regresar” de retorno al menú principal.

3.3.5.3 Opción “Inundación”

Cuando se selecciona esta opción la aplicación envía el comando UDCSI a la interfaz estacionaria. Luego de esta acción, el dispositivo espera la respuesta que, al llegar, es interpretada como los estados de los dos detectores de presencia de agua en sus respectivas áreas de detección (Ver Figura 3-53). De manera similar a las pantallas antes descritas, esta fue programada con un botón de regreso al menú principal.

Figura 3-53.- Pantalla “Sensores Inundacion”

3.3.5.4 Opción “Humo”

La selección de esta opción comanda el envío del código UDCSH, para obtener la información del estado de los sensores de humo en el taller y departamento técnico (Ver Figura 3-54).

Figura 3-54.- Pantalla “Presencia de Humo”

3.3.5.5 Opción “Magnetico”

Al seleccionarse esta opción se envía el código UDCSM del cual se obtiene como respuesta el estado de la puerta de ingreso a las instalaciones como se indica en la figura siguiente.

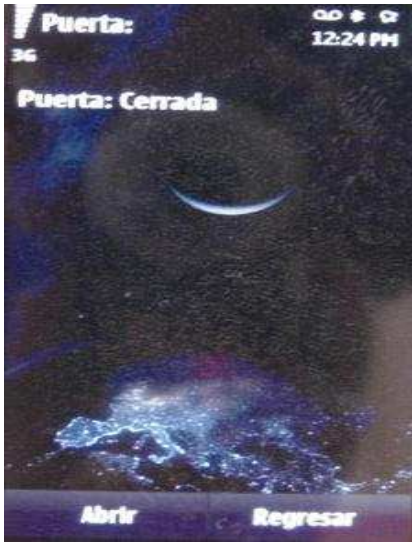


Figura 3-55.- Pantalla “Puerta”

Esta pantalla tiene programado el botón “Abrir”, el cual al ser seleccionado hace que la aplicación envíe el código 1AP0001 que es interpretado por la interfaz estacionaria como la orden de activación de la cerradura eléctrica de la puerta.

3.4 SOFTWARE DE CONTROL DE LOS NODOS DE LA RED RS−485

El código de control para cada microcontrolador de los distintos nodos fue desarrollado mediante el programa Bascom AVR.

En todos los programas para los nodos se emplea el siguiente código para establecer el valor del cristal que se va a emplear y la velocidad de transmisión y recepción de datos por el puerto serial.

```
$crystal = 8000000
$baud = 9600
```

Así se establece un cristal de 8MHz y una velocidad de comunicación de 9600 baudios.

En el caso del nodo maestro se configura también un segundo puerto serial para la comunicación con el computador, utilizando el siguiente comando.

```
$baud1 = 9600
```

Todos los nodos esclavos envían el valor de las variables medidas al nodo maestro empleando el siguiente formato:

Código del sensor + 4 dígitos

Es decir, se envía tramas como: 1SH0001. Donde 1SH es el código del sensor, como se verá más adelante y 0001 son los cuatro dígitos que contienen el valor de la variable medida.

A continuación se explican las rutinas más importantes de cada nodo:

3.4.1 NODO MAESTRO

En este apartado se explica el programa empleado para cumplir con todas las tareas del nodo maestro, que se vieron anteriormente. El código que se expone es el que se utilizó en el microcontrolador Atmega164. También se desarrolló un programa para el microcontrolador Atmega64, el cual puede verse en el Anexo B.

Después de declarar todas las variables a ser utilizadas, se configuró dos timers. Uno de estos (timer1) es utilizado para determinar cuánto tiempo se espera la respuesta de los esclavos, para así continuar con la petición de información. El segundo timer (timer2) se utiliza para determinar el tiempo en el que se va a accionar la cerradura magnética para abrir la puerta. Así se establecieron los siguientes tiempos:

- Tiempo de espera de respuesta de los esclavos = 0.5s
- Tiempo de accionamiento de la cerradura magnética = 0.16s

La rutina que realiza todo el tiempo el microcontrolador del nodo maestro es la siguiente:

```
Do
  Incr Descolgar
  If Permiso = 1 Or Descolgar >= 100000 Then
    Descolgar = 0
    Gosub Cargar
    Start Timer1
    Permiso = 0
  End If
  Gosub Vermh
Loop
```

Se emplea la variable permiso para determinar cuándo se debe hacer la petición de información a los esclavos. Así, cuando permiso es 1 quiere decir que ya se ha esperado el tiempo anteriormente mencionado y es momento de pedir información al siguiente esclavo. La subrutina “Cargar”, como se verá a continuación, es en la que se realiza el envío, por la red RS-485, de los códigos acordados para pedir información de las distintas variables medidas por los nodos esclavos. También, se testean los pines correspondientes a los nodos de presencia de humo (I1 e I2) y al sensor magnético (M1) para determinar si hay presencia de agua o si la puerta ha sido abierta. Es última tarea se realiza en la subrutina “Vermh”.

Subrutina “Cargar”:

```
Incr Auxi
Select Case Auxi
Case 1 : Frame1 = "1ST"
Case 2 : Frame1 = "2ST"
Case 3 : Frame1 = "3ST"
Case 4 : Frame1 = "4ST"
Case 5 : Frame1 = "1SC"
Case 6 : Frame1 = "1SU"
Case 7 : Frame1 = "2SU"
Case 8 : Frame1 = "1SI"
Case 9 : Frame1 = "2SI"
        Auxi = 0
Case Else : Auxi = 0
End Select
Print Frame1
```

La variable Frame1 es aquella que se carga con la codificación que se utiliza para pedir información de las variables medidas a los nodos esclavos.

- 1ST: Petición de temperatura al sensor T1 (Dpto. Técnico)
- 2ST: Petición de temperatura al sensor T2 (Dpto. Técnico)
- 3ST: Petición de temperatura al sensor T3 (Taller)
- 4ST: Petición de temperatura al sensor T4 (Taller)
- 1SC: Petición del valor medido de corriente
- 1SU: Petición del valor de humedad en el ambiente del Dpto. Técnico
- 2SU: Petición del valor de humedad en el ambiente del Taller
- 1SI: Petición del estado de presencia de agua en el Dpto. Técnico
- 2SI: Petición del estado de presencia de agua en el Taller

La instrucción Select case se usa para que en cada iteración se envíe cada uno de dichos códigos, después de haber esperado el tiempo antes mencionado.

Subrutina “Vermh”:

```
If M1 = 1 Then
Magnetico = 1
Elseif M1 = 0 Then
Magnetico = 0
End If
```

Con este código se determina si la puerta se ha abierto o no, testeando el pin M1. La variable “Magnetico” se utiliza para guardar la información e irla modificando para enviar datos actualizados cada vez que el computador lo pida. Si Magnetico =1 quiere decir que la puerta de las instalaciones está abierta.

```
If H1 = 0 Then
Incr Cuentahumo
Cuentaapagado = 0
Alarmhumo = 1
Elseif H1 = 1 Then
Incr Cuentaapagado
   If Cuentahumo >= 100000 Or Cuentaapagado >= 25000 Then
    Alarmhumo = 0
    Cuentahumo = 0
    Cuentaapagado = 0
    End If
End If
```

Estas líneas de código también están dentro de la subrutina “Vermh”. Se emplean sentencias similares para el sensor de humo (H2). Debido a que los detectores de humo suelen tener un bip de testeo que resulta en un pico de voltaje a su salida, que puede difundir una falsa alerta, una vez que se reciba información de que hay presencia de humo se realizan testeos consecutivos del pin de humo correspondiente para validar la señal de alerta y si esta información se mantiene entonces se hace la variable Alarmhumo=1, lo cual indica presencia de humo.

Para reportar el estado de los detectores de humo y del sensor magnético, el nodo maestro utiliza también los siguientes códigos:

- 1SH: Petición del estado del detector de humo en el Dpto. Técnico
- 2SH: Petición del estado del detector de humo en el Taller
- 1SM: Petición del estado del sensor magnético

En el programa también se configuran dos interrupciones de recepción de datos. Una de estas se emplea para recibir información del computador y la segunda interrupción para recibir información de la red RS-485 en la que se encuentran los nodos esclavos.

El computador utiliza la misma codificación antes descrita, para pedir información al nodo maestro acerca de los datos obtenidos de los nodos esclavos, de los detectores de humo y del sensor magnético. En tanto, los nodos esclavos envían tramas como la siguiente: 1ST1695, la cual quiere decir que el sensor de temperatura T1 ha medido un valor de temperatura de 16.95°C.

Cuando uno de los nodos esclavos (temperatura, corriente o presencia de agua) no responde hasta que se termina el tiempo de espera, el string que guarda la información de dicho nodo se establece como ERR2. Esto se interpreta como que el nodo esclavo respectivo no se encuentra en la red. Además, debido a que los sensores de temperatura implementados tienen un código único, se puede determinar si uno de estos se desconecta de su nodo esclavo, por tanto, cuando esto suceda dicho nodo envía el código del sensor más ERR1, esta opción se tiene únicamente para estos sensores. Así, si se envía 1STERR1, quiere decir que el sensor de temperatura T1 ha sido desconectado del nodo esclavo. Mientras que si se envía 1STERR2, significa que todo el nodo esclavo de temperatura 1 se ha desconectado de la red.

3.4.2 NODO DE PRESENCIA DE AGUA

Este nodo, como se explicó en el anterior capítulo, fue diseñado para cumplir con dos tareas. Una de ellas es detectar la presencia de agua en lugares específicos y la segunda es medir la humedad en el ambiente. Debido a que la tarea de medición de humedad fue integrada a este nodo para una futura expansión del sistema SECURE SYS y está fuera del alcance del presente proyecto, se omitirá la explicación del código utilizado para la medición de humedad mediante el sensor HS1101, pero se lo puede encontrar en el Anexo B.

A continuación se muestra la rutina que ejecuta el microcontrolador del nodo de presencia de agua del departamento técnico. La rutina es la misma para el caso del nodo de presencia de agua del taller, en la cual se emplea la codificación 2SU y 2SI.

```
Do
If Indicador = 1 Then
   Gosub Convertir
   Dato = "1SU" + Humedad
   Call Enviar
   Envia = 0
   Indicador = 0
End If

If Indicador2 = 1 Then
  If I1 = 0 Then
  Dato = "1SI0001"
  Call Enviar
  Envia = 0
  Inundacion = 1
  Elseif I1 = 1 Then
  Dato = "1SI0000"
  Call Enviar
```

```
  Envia = 0
  Inundacion = 0
  End If
  Indicador2 = 0
End If
Recibe = 0
Loop
End
```

— En estas líneas de programación se llama a la subrutina “Convertir”, en la cual se transforma a porcentaje de humedad relativa la salida del sensor de humedad, la cual es una señal de pulsos, cuya frecuencia representa el valor de humedad. A continuación se llama a la subrutina “Enviar” en la cual se habilita la transmisión de datos para posteriormente enviar el dato de humedad, con la codificación requerida, al nodo maestro.

— Este código de programación se emplea para leer el dato de presencia de agua y enviarlo al nodo maestro empleando la subrutina “Enviar”.

La variable “Indicador” se establece en 1 cuando el nodo maestro pide información de humedad. Mientras que, la variable “Indicador2” se hace 1 cuando este pide información de presencia de agua. Así se enviará únicamente el dato solicitado.

El programa contiene también una subrutina de recepción de datos, la cual se ejecuta cada vez que llega un dato al puerto serial del microcontrolador. Es en esta subrutina en donde se ponen en 1 las variables “Indicador” e “Indicador2” de acuerdo a la información que el nodo maestro haya solicitado.

3.4.3 NODO CORRIENTE

En el programa para el nodo de corriente se configura un conversor análogo digital para adquirir la información que envía el sensor de corriente, descrito en el anterior capítulo.

La rutina que ejecuta constantemente el microcontrolador es la siguiente:

```
Do
  Poder = 1
  Call Muestreo
  Envia = 0
  Recibe = 0
  Poder = 0
Loop
End
```

Los pines del microcontrolador nombrados como “Poder”, “Envia” y “Recibe” se emplean para señalización:

- Poder: Señaliza que el microcontrolador está ejecutando el programa, lo cual significa que no se ha colgado.
- Envia: Señaliza cada vez que se envía un dato. Si Envia = 1, quiere decir que en ese momento va a enviarse un dato.
- Recibe: Señaliza la recepción de datos. De igual forma, si Recibe=1, significa que en ese momento se recibió un dato por el puerto serial del microcontrolador.

La subrutina “Muestreo”, que se muestra a continuación, es aquella en la que se adquiere el valor de corriente que llega al conversor análogo digital y lo va agregando a la variable “Suma” hasta que se hayan tomado 61 muestras, después de lo cual se llama a la subrutina “Rmscalc”. En esta última subrutina se hace el promedio de todos los datos adquiridos y al resultado le da formato para

que esté listo para enviarse cuando el nodo maestro lo solicite. Este formato es el que se describió al inicio del numeral 3.4.

```
Sub Muestreo
 V = Getadc(1)
 Incr Aux
Suma = Suma + V

 If Aux > 60 Then
Call Rmscalc
 Aux = 0
 End If
End Sub
```

De igual forma que en los anteriores nodos, se tiene una subrutina que se ejecuta cada vez que se recibe un dato. Una sección de dicha subrutina se muestra a continuación:

```
For Num = 1 To Lon
      Trama = Mid(buf , Num , 3)
      Trama = Left(trama , 3)

      If Trama = "1SC" Then
      Dato = "1SC" + Current
      Call Enviar
      Num = Lon
      End If
Next
```

Por medio de la estructura FOR se va comprobando si dentro de la trama recibida se encuentra el código del sensor de corriente (1SC), ya que puede darse el caso en el que lleguen datos no deseados en conjunto con el código. Si se detecta

dicho código, se llama a la subrutina “Enviar” para responder al nodo maestro con el valor de corriente.

3.4.4 NODO TEMPERATURA

En el programa del nodo de temperatura se configuró el PINB1 del pórtico B como entrada de los sensores 1wire. Por este pin se realiza la búsqueda de los sensores y la adquisición de los valores de temperatura.

Debido a la forma de manejo de este tipo de sensores, descrito en el capitulo anterior, el programa realiza la siguiente rutina al iniciar:

```
Dsid1(1) = 1wsearchfirst(pinb , 1)

For Aux = 1 To 10
Dsid2(1) = 1wsearchnext(pinb , 1)

  For Aux2 = 1 To 8
  If Dsid2(aux2) <> Dsid1(aux2) Then
  Aux2 = 8
  Ind = 1
  End If
  Next

If Ind = 1 Then
  Aux = 10
  Ind = 0
Else
  Dsid2(1) = 0
End If
Next Aux
```

Las instrucciones 1wsearchfirst y 1wsearchnext se emplean para adquirir el código único de los dos sensores conectados en paralelo. Al no tener la seguridad de haber adquirido dos códigos diferentes y no un solo código dos veces se hace una comparación de las variables Dsid1 y Dsid2, que son arreglos de 8 bytes. La comparación se realiza byte a byte para determinar si los códigos son distintos. La búsqueda de un segundo código único se realiza por diez ocasiones; si no se encuentra un código de identificación diferente al encontrado primero se asigna el valor de cero al arreglo Dsid2.

Tras realizar la adquisición de los códigos de identificación, el microcontrolador realiza la siguiente rutina para cumplir con la función de enviar los valores de temperatura al nodo maestro:

```
Do

Call Temperatura

If Indicador1 = 1 Then
Longitud = Len(ctg1)
   If Longitud = 4 Then
   Dato = "1ST" + Ctg1
   End If
Call Enviar
Indicador1 = 0
End If

If Indicador2 = 1 Then
Longitud = Len(ctg2)
   If Longitud = 4 Then
   Dato = "2ST" + Ctg2
   End If
Call Enviar
Indicador2 = 0
```

```
End If
Recibe = 0
Loop
End
```

Los pines nombrados como “Recibe” y “Envia” se emplean para señalización del correcto funcionamiento del nodo, de la siguiente manera:

- Recibe: señaliza que se ha recibido un dato. Para esto se pone el estado del pin en 1 lógico.
- Envia: señaliza la acción de enviar un dato al nodo maestro. El estado de 1 lógico en este pin indica que se ha realizado un envío.

La subrutina “Temperatura” se encarga de la adquisición de los valores de temperatura. Sus líneas de código más relevantes se muestran a continuación:

```
1wverify Dsid1(1)

If Err = 1 Then
   Sensor1 = 0
   Incr Uno
   If Uno >= 3 Then
   Start Timer2
   Ctg1 = "ERR1"
   Uno = 4
   End If

Elseif Err = 0 Then
   Stop Timer2
    Timer2 = 0
    Sensor1 = 1
    Uno = 0
    1wwrite &HBE
```

```
  Sc(1) = 1wread(9)
  If Sc(9) = Crc8(sc(1) , 8) Then
    Dg = Decigrades(sc(9))
    Ctg1 = Str(dg)
    Ctg1 = Format(ctg1 , "0000")

  End If
End If
```

La instrucción 1wverify Dsid1(1) verifica que el sensor con el código Dsid1 se encuentre todavía conectado al microcontrolador y devuelve la variable Err con estado de 1 si el sensor no se encuentra conectado. De ser este el caso se carga el código ERR1 al valor del dato de temperatura del sensor. Caso contrario se devuelve el valor de Err como cero y se continúa con la adquisición del valor de temperatura. Tras recibir los datos empleando los comandos descritos en el capítulo 2 se realiza la lectura del valor de temperatura del arreglo de bytes recibido mediante la función “Decigrades” que se muestra a continuación:

```
Function Decigrades(byval Sc(9) As Byte)
  Decigrades = Makeint(sc(1) , Sc(2))
  Decigrades = Decigrades * 10
  Decigrades = Decigrades / 2
End Function
```

Al darse el caso de que el sensor con el código único guardado en Dsid1 se haya desconectado, se produce el desborde del Timer2 cuya rutina de interrupción se encarga de buscar nuevamente los sensores conectados al microcontrolador. Luego se continúa con la tarea de recepción y envío de datos.

CAPÍTULO 4
PRUEBAS Y RESULTADOS

4.1 INTRODUCCION

Después de culminar la programación de la red de nodos, la aplicación para la interfaz estacionaria y la aplicación para la interfaz móvil, se procedió a realizar pruebas con del sistema SECURE SYS con el fin de comprobar su correcto funcionamiento.

4.2 PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA SECURE SYS.

Todas las pruebas de funcionamiento del sistema se realizaron dentro de las instalaciones de la empresa UPDATECOM Cia. Ltda, bajo las siguientes condiciones físicas generales:

- Instalaciones de una sola planta.
- Paredes gruesas y de ladrillo.
- Interferencia electromagnética proveniente de los dispositivos electrónicos presentes en las instalaciones.

4.2.1 PRUEBA DE ACTIVACION DE ALERTAS DE TEMPERATURA.

Para realizar la prueba de activación de las alertas de temperatura se realizaron pruebas en las dos diferentes áreas de interés.

- Taller:

Para esta prueba se puso como límite de activación de la alerta de temperatura el valor de 23°C en la pestaña “Parametros” dentro de la ventana “Administrar” de la aplicación para la interfaz estacionaria. También se habilitó el envío de mensajes SMS y de correo electrónico en la ventana “Conectividad”.

A continuación se procedió a calentar uno de los sensores de temperatura ubicados en el área del Taller, empleando para esto un cautín caliente.

Como resultado la activación de la alarma de temperatura sucedió en 24.75°C, como se puede observar en la ventana de monitoreo de sensores en la figura siguiente.

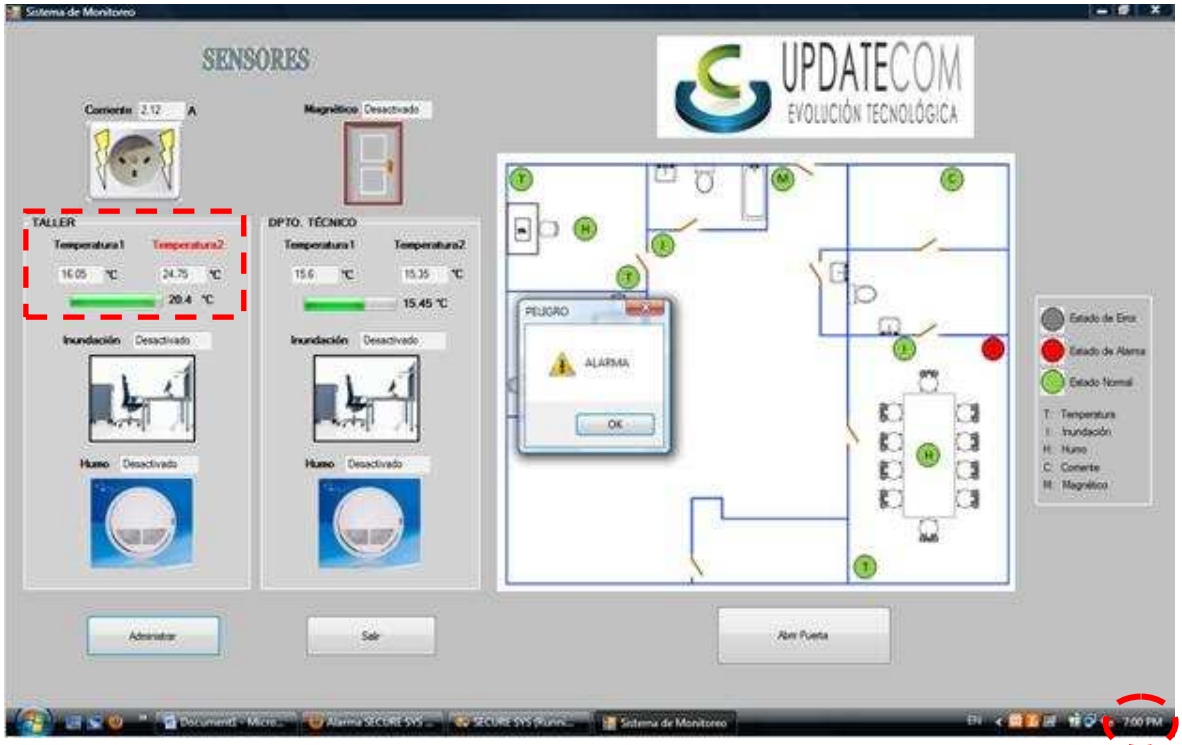


Figura 4-1.- Alerta de temperatura en el Taller

Tras darse el estado de alerta, el sistema de monitoreo envió un mensaje SMS y un mensaje de correo electrónico con el valor de temperatura que provocó el estado de alerta, como se observa en las figuras siguientes.

Figura 4-2.- Mensaje SMS de alerta de temperatura en el Taller

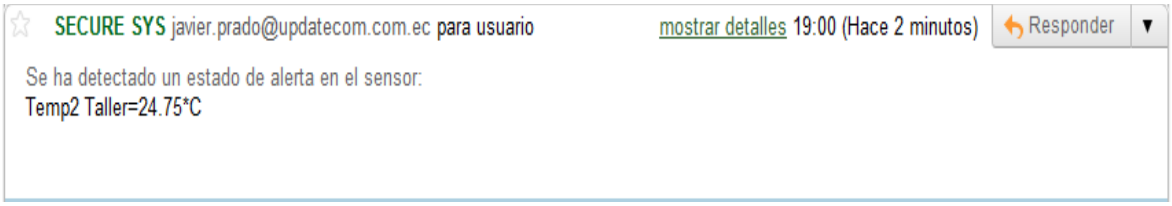


Figura 4-3.- Correo electrónico de alerta de temperatura en el Taller

Como se puede observar en las imágenes la hora de recepción de los mensajes de texto y correo electrónico (7:00 pm) es la misma que la hora en que se detectó el estado de alerta (7:00pm).

- Departamento Técnico

El valor límite de alerta de temperatura se configuró en 22°C y se habilitaron las opciones de envío de alertas, igual que en la prueba anterior.

Como resultado, el estado de alerta se detectó en 28.05°C, debido a que la temperatura varió rápidamente por el uso de un cautín, como se ve a continuación.

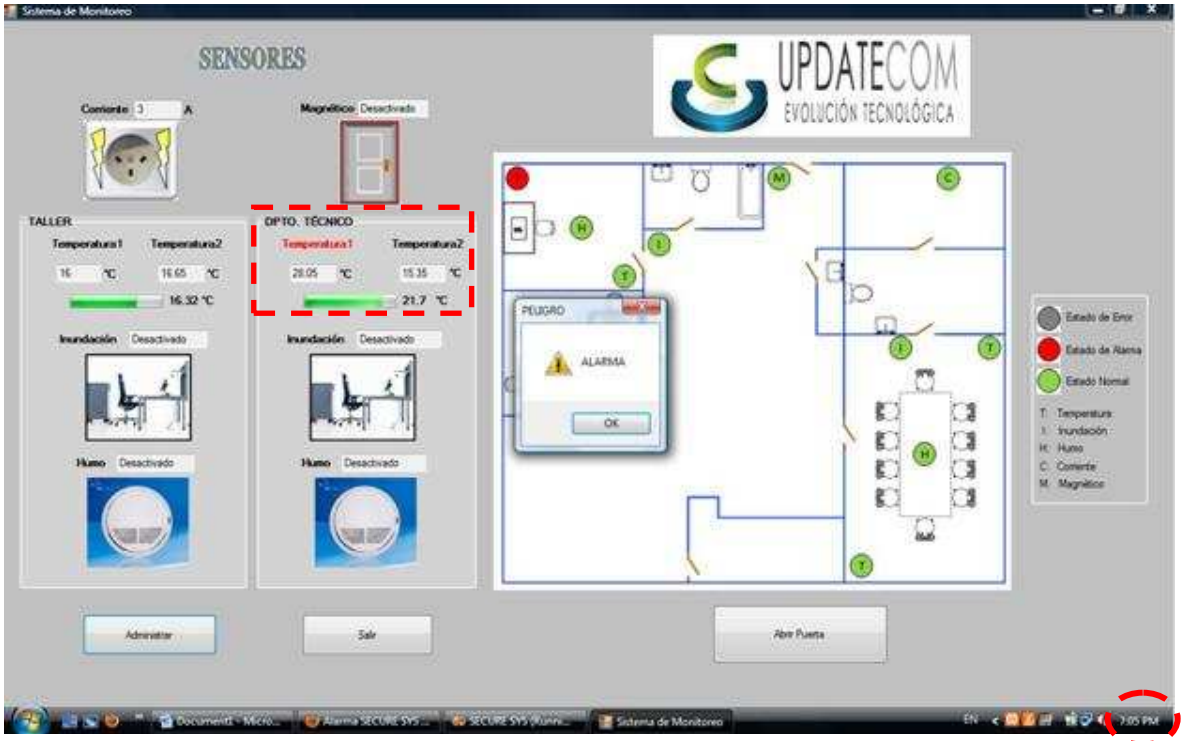


Figura 4-4.- Alerta de temperatura en el Dpto.Técnico

Los mensajes SMS y de correo electrónico que indicaron el estado de alerta junto con el valor de temperatura que lo provocó se muestran en las figuras siguientes. Como se observa la alarma fue detectada a las 7:05pm y los mensajes de texto y correo electrónico se recivieron a la misma hora.

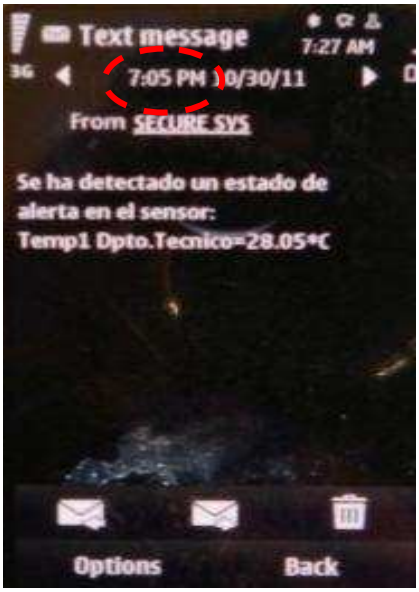


Figura 4-5.- Mensaje SMS de alerta de temperatura en el Dpto. Técnico

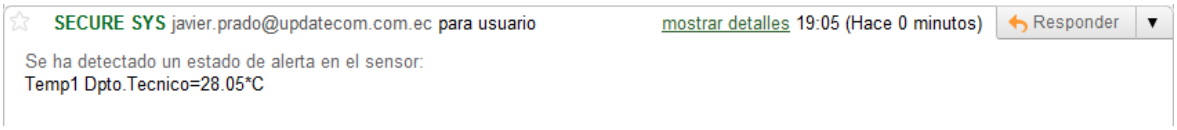


Figura 4-6.- Correo electrónico de alerta de temperatura en el Dpto. Técnico

4.2.2 PRUEBA DE ACTIVACION DE ALERTAS DE PRESENCIA DE AGUA.

Las pruebas de activación de los detectores de presencia de agua se realizaron en las dos áreas de interés.

- Taller:

La activación de la alerta de detección de agua se provocó vertiendo una pequeña cantidad de agua sobre uno de los pares de electrodos del circuito de detección de presencia de agua. Como resultado el sistema SECURE SYS realizó el envío de alertas por medio de mensajes SMS y correo electrónico, además de la alerta

visual en la pantalla de la interfaz estacionaria como se puede observar en las figuras siguientes.

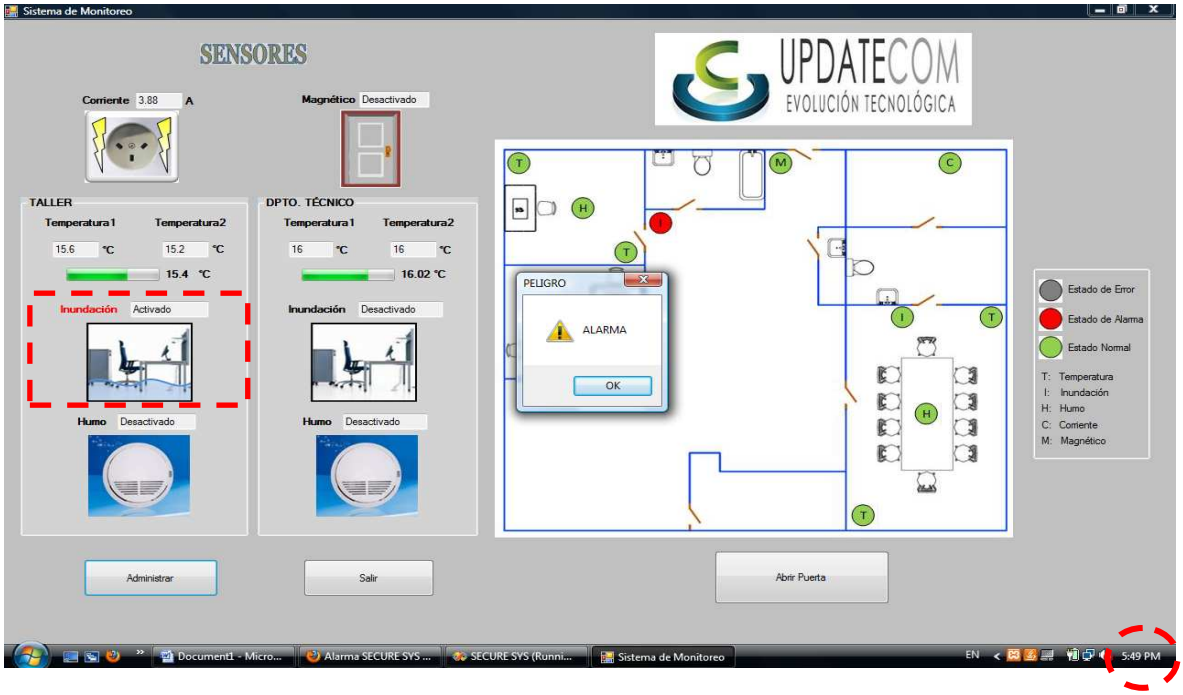


Figura 4-7.- Alerta de detección de agua en el Taller

Figura 4-8.- Mensaje SMS de alerta de detección de agua en el taller

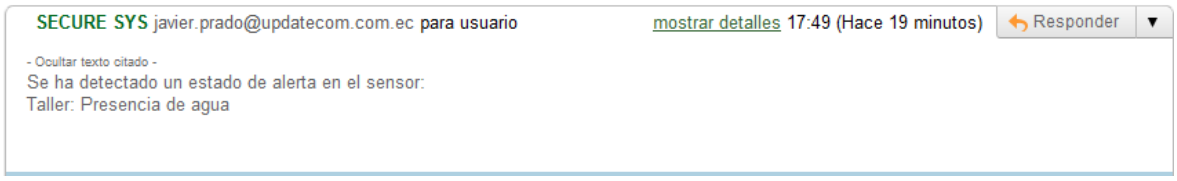


Figura 4-10.- Correo electrónico de alerta de detección de agua en el Taller

La hora de detección de la alerta se llevó a cabo a las 5:49pm, al igual que la recepción del mensaje de correo electrónico y SMS.

- Departamento Técnico

Para la prueba de activación del detector de agua en el Dpto. Técnico se tomaron medidas similares a las tomadas en la prueba anterior y se obtuvieron resultados similares como se observa en las figuras siguientes.

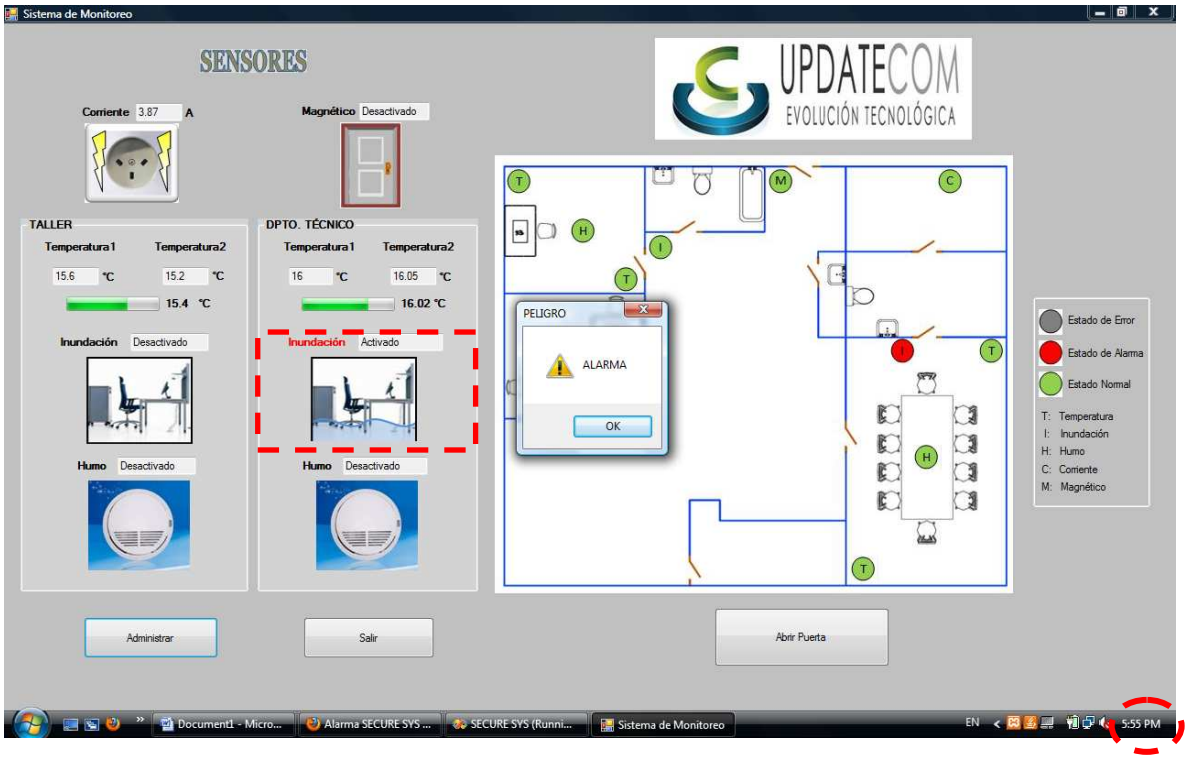


Figura 4-11.- Alerta de detección de agua en el Dpto. Técnico

Figura 4-12.- Mensaje SMS de alerta de detección de agua en el Dpto. Técnico

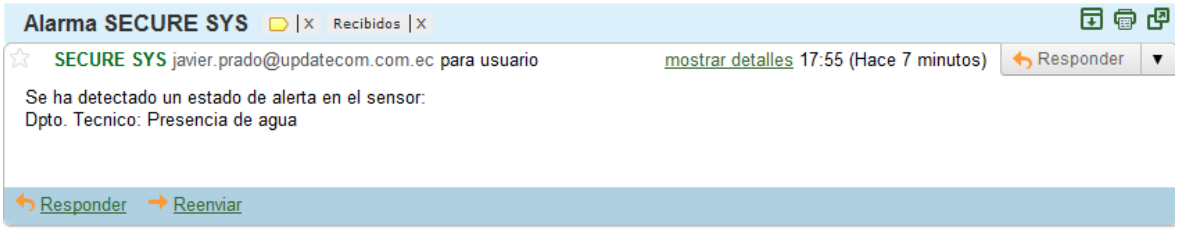


Figura 4-13.- Correo electrónico de alerta de detección de agua en el Dpto. Técnico

Como se observa el estado de alerta de agua fue detectada a la 5:55pm, misma hora en al que fueron recibidas los mensajes de alerta.

4.2.3 PRUEBA DE ACTIVACION DE ALERTAS DE PRESENCIA DE HUMO.

De la misma manera que en los casos anteriores estas pruebas se realizaron en las dos áreas de interés.

Para ambas pruebas se colocó un papel en llamas a una altura 72 cm del piso (sobre las mesas de trabajo), esto es, a una distancia vertical de 1.95m de los detectores de humo. Obteniéndose los resultados expuestos a continuación.

- Taller:

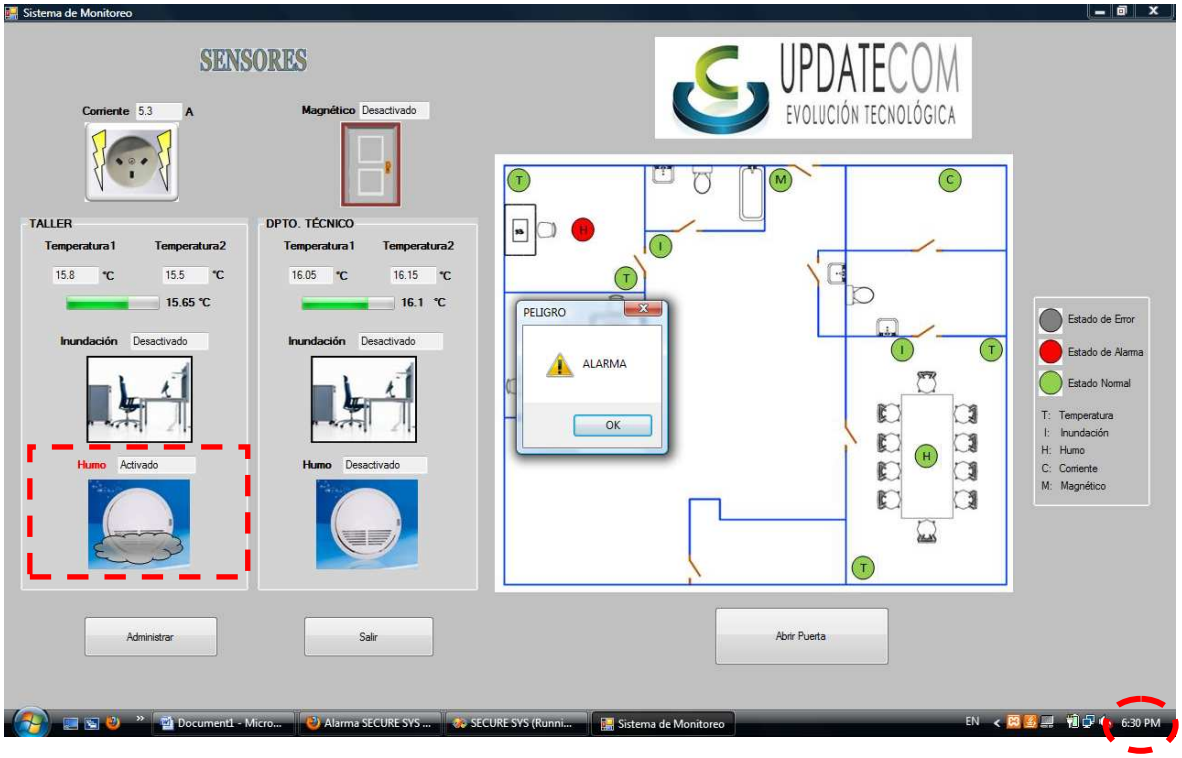


Figura 4-14.- Alerta de detección de humo en el taller

Figura 4-15.- Mensaje SMS de alerta de detección de humo Taller

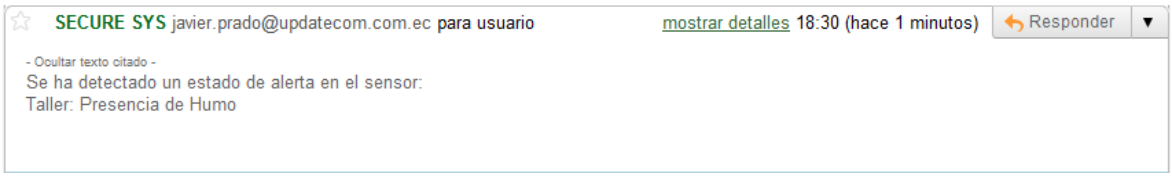


Figura 4-16.- Correo electrónico de alerta de detección de humo taller

De acuerdo a las figuras anteriores la situación de alerta se detectó a las 6:30pm y a la misma hora llegaron a su destino los mensajes de texto y de correo electrónico.

- Departamento Técnico

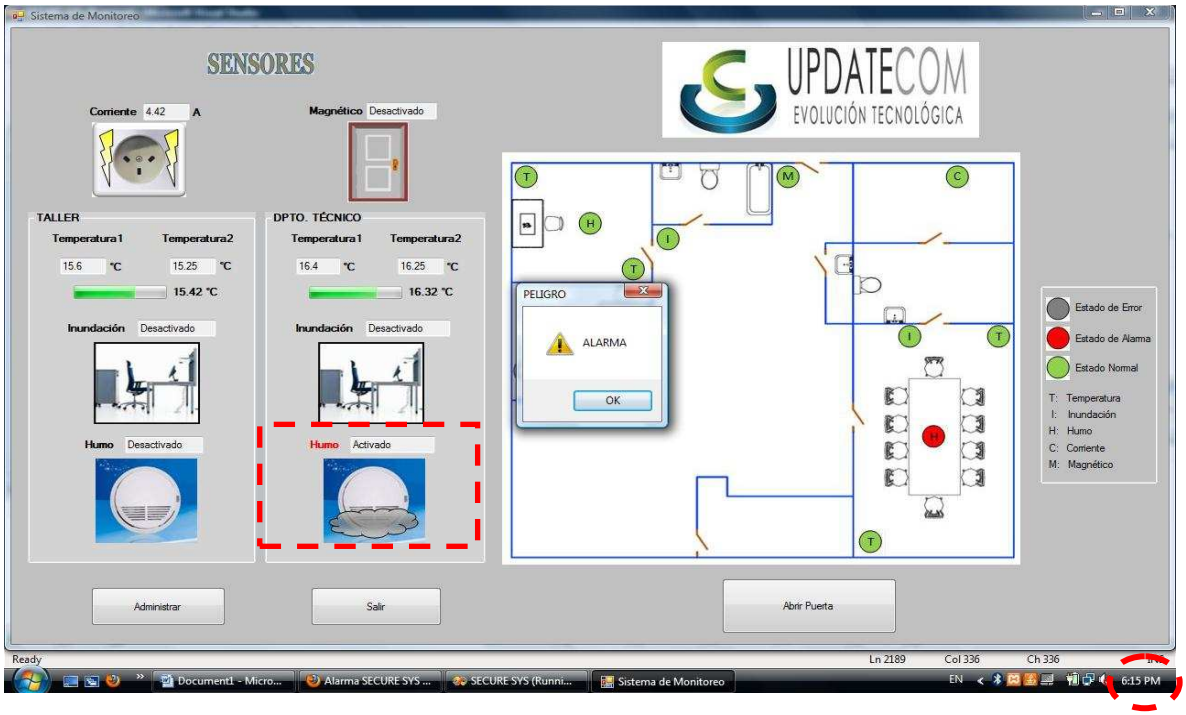


Figura 4-17.- Alerta de detección de humo Dpto. Técnico

Figura 4-18.- Mensaje SMS de alerta de detección de humo Dpto. Técnico

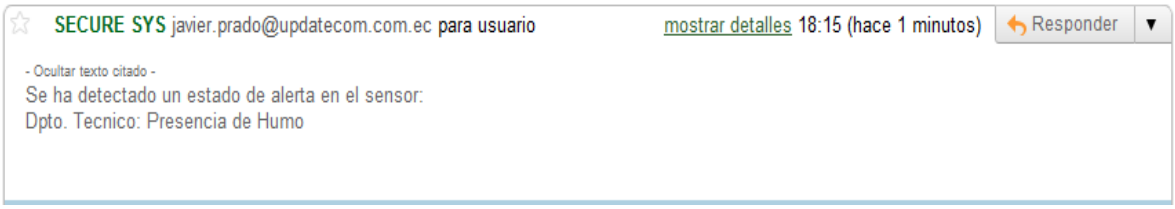


Figura 4-19.- Correo electrónico de alerta de detección de humo Dpto. Técnico

Como se observa, la detección de la alerta de humo y la recpción de los mensajes de alerta se llevaron a cabo a las 6:15pm.

4.2.4 PRUEBA DE ACTIVACION DE ALERTAS DE CORRIENTE.

Para las pruebas de detección de sobrecorriente y subcorriente se pusieron los límites como se indican a continuación.

Límite superior: 5A
Límite inferior: 1A

Para provocar el incremento y decremento de corriente consumida por las instalaciones se encendieron y apagaron diferentes cargas eléctricas dentro de las instalaciones.

Como resultado se obtuvo que la activación de la alerta de sobrecorriente se dio al valor de 5.14A, mientras que la alerta de subcorriente se dio en el valor de 0.9A como se muestra en las figuras a continuación.

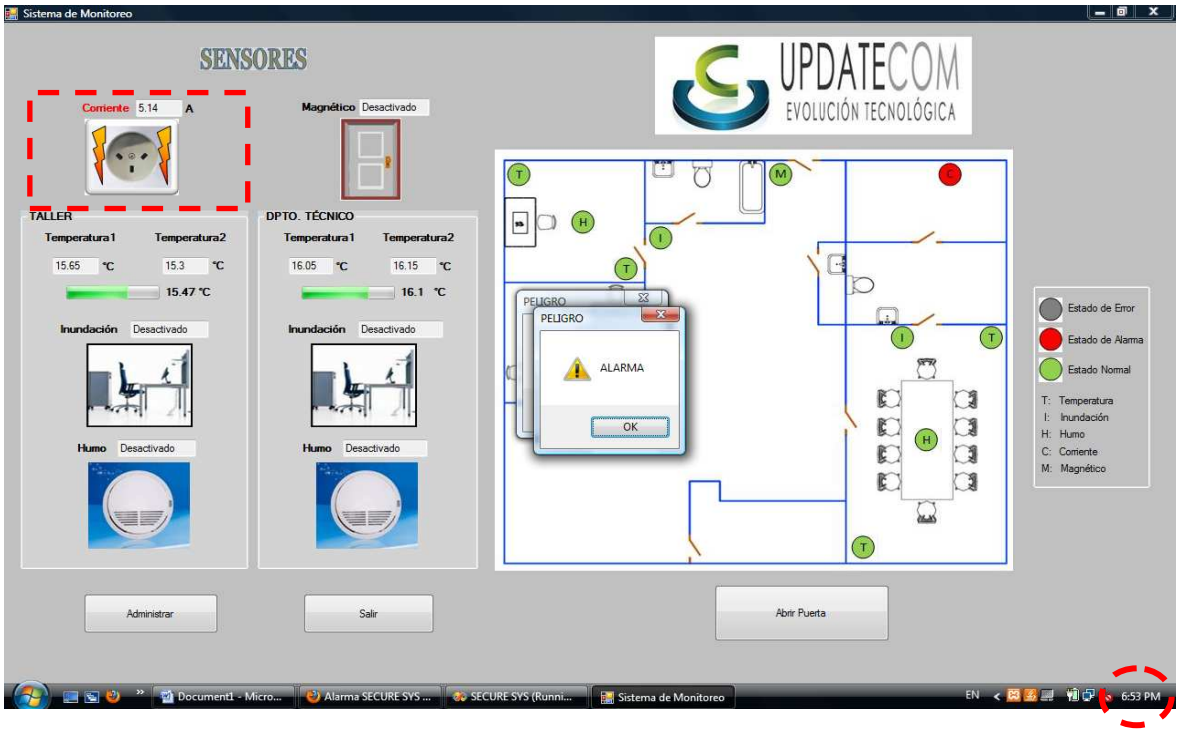


Figura 4-20.- Alerta de sobrecorriente

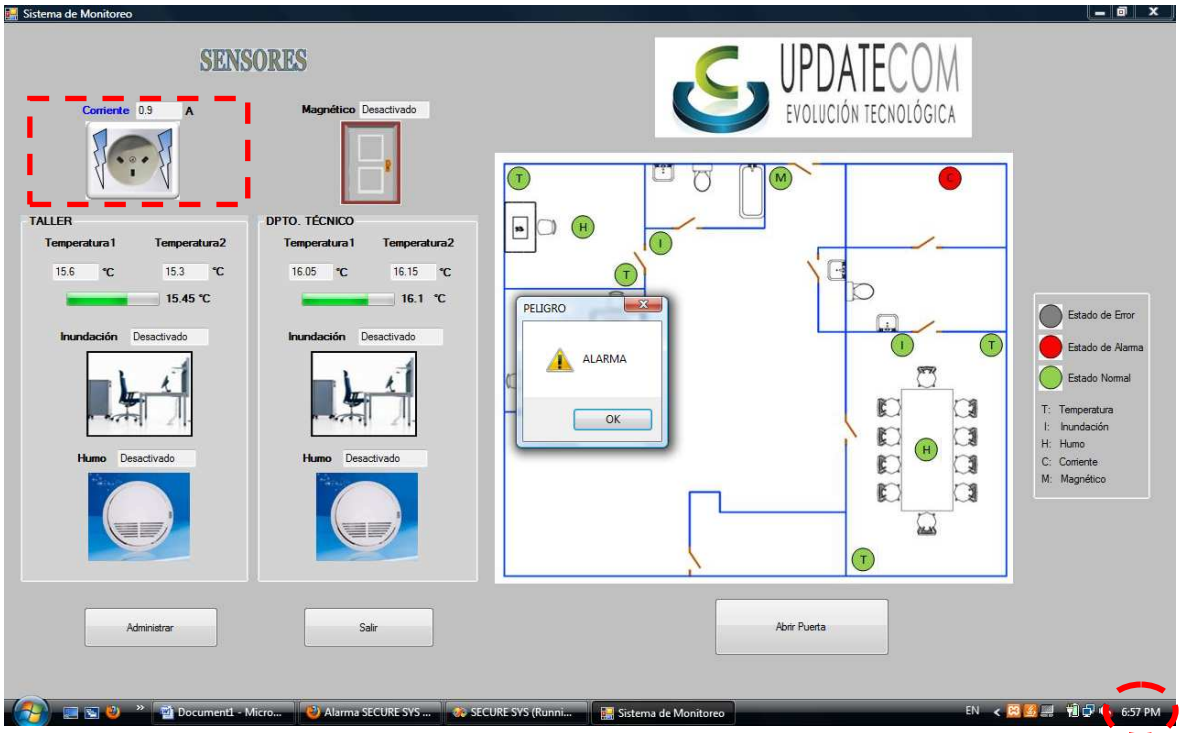


Figura 4-21.- Alerta de subcorriente.

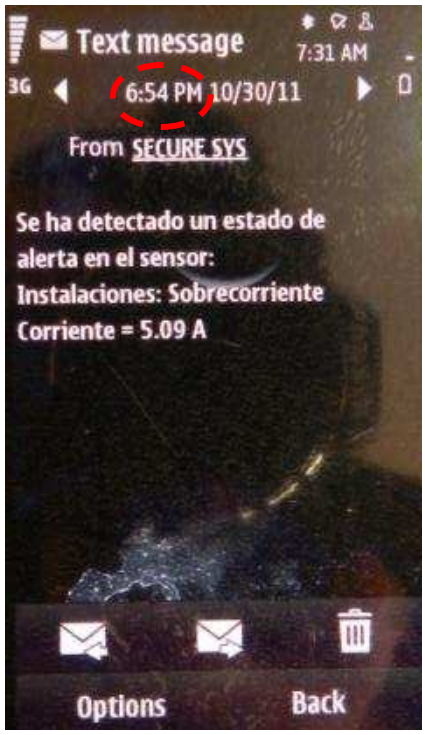


Figura 4-22.- Mensaje SMS de alerta de sobrecorriente

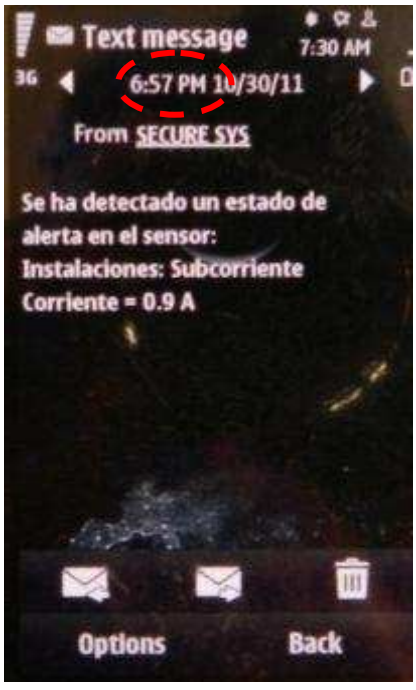


Figura 4-23.- Mensaje SMS de alerta de subcorriente

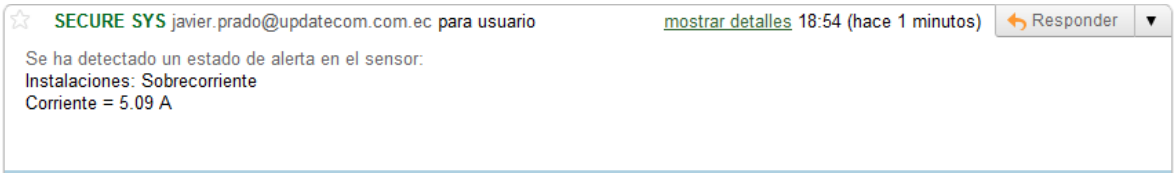


Figura 4-24.- Correo electrónico de alerta de sobrecorriente

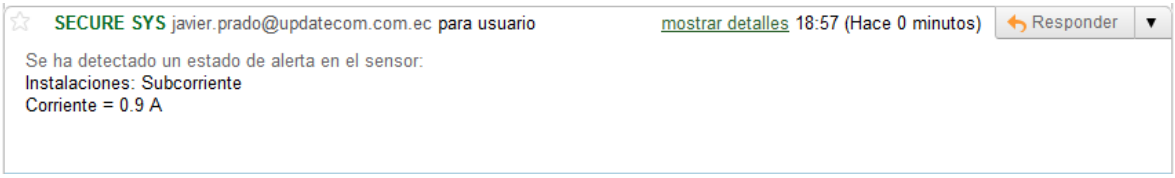


Figura 4-25.- Correo electrónico de alerta de subcorriente

4.2.5 PRUEBA DE ACTIVACION DE ALERTA DE APERTURA DE PUERTA.

Como se explicó anteriormente para que se realice el envío de alertas de apertura de puerta se requiere que se habilite este envío en la pestaña “Parametros” de la ventana “Administrar”. Después de ser habilitada esta opción, la aplicación espera un lapso de 2 minutos, tiempo en el cual las personas dentro de la empresa pueden salir sin ocasionar ninguna alarma. Al término de este período de tiempo el sistema envía mensajes de alerta cada vez que se detecta la apertura de la puerta, como se muestra en los resultados de la prueba realizada, en las figuras siguientes.

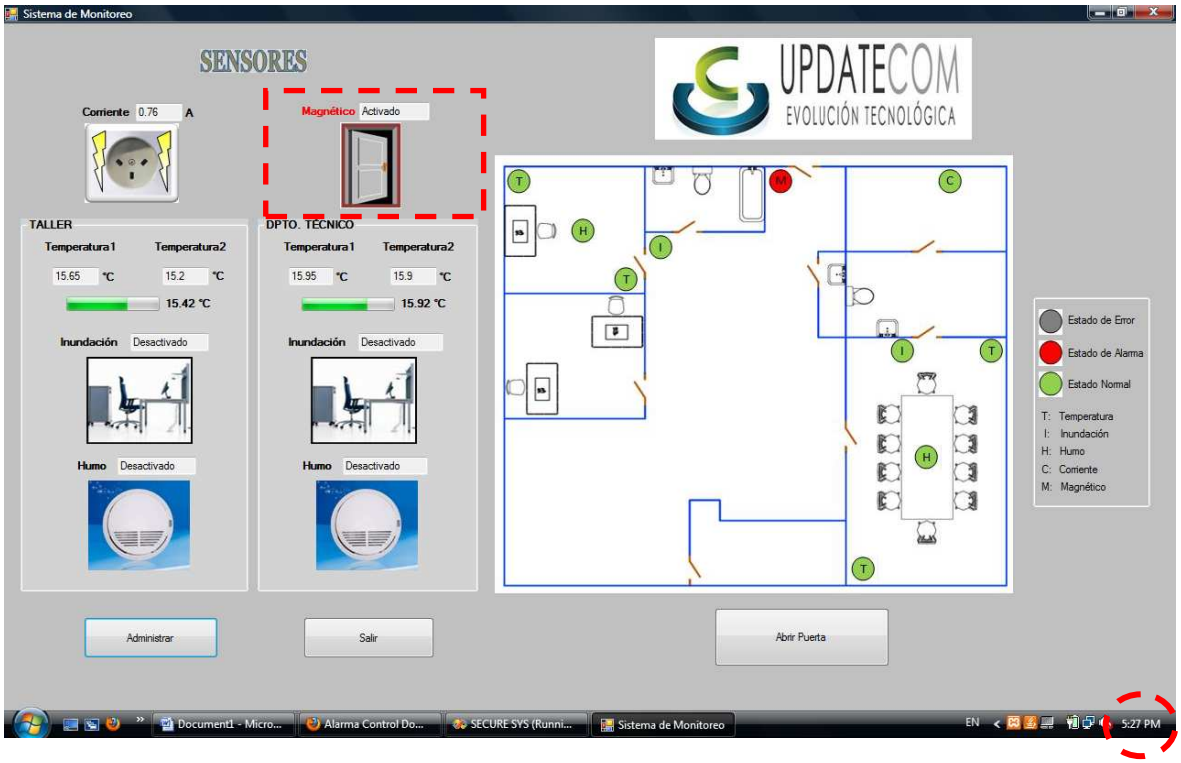


Figura 4-29.- Alerta de apertura de puerta

Figura 4-30.- Mensaje SMS de alerta de apertura de puerta

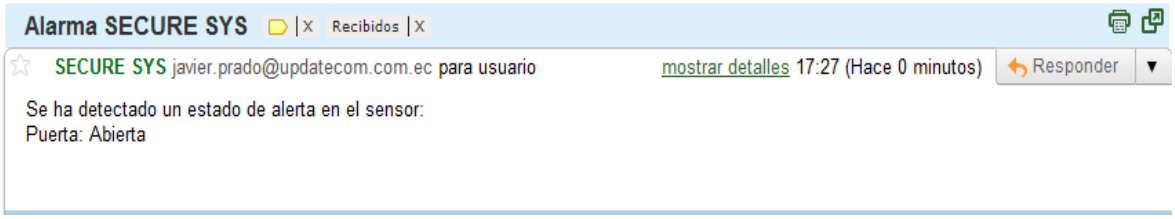


Figura 4-31.- Correo electrónico de alerta de apertura de puerta

4.2.6 PRUEBA DE ALCANCE DE INTERFAZ MÓVIL.

Ya que la finalidad de la interfaz móvil es proveer de información del sistema SECURE SYS al usuario dentro de las instalaciones sin necesidad de estar junto a la interfaz estacionaria se realizaron pruebas de conectividad en varios puntos de las instalaciones para determinar el alcance del dispositivo bluetooth empleado en el computador. Así, se tuvo como resultado que la antena del dispositivo Bluetooth, que tiene como radio de alcance 10m con línea de vista, es capaz de enviar datos a la interfaz móvil en cualquier punto dentro de las instalaciones. Además, la conectividad entre el dispositivo móvil y el computador se mantiene hasta llegar a 1.5m fuera de las instalaciones, con la puerta cerrada.

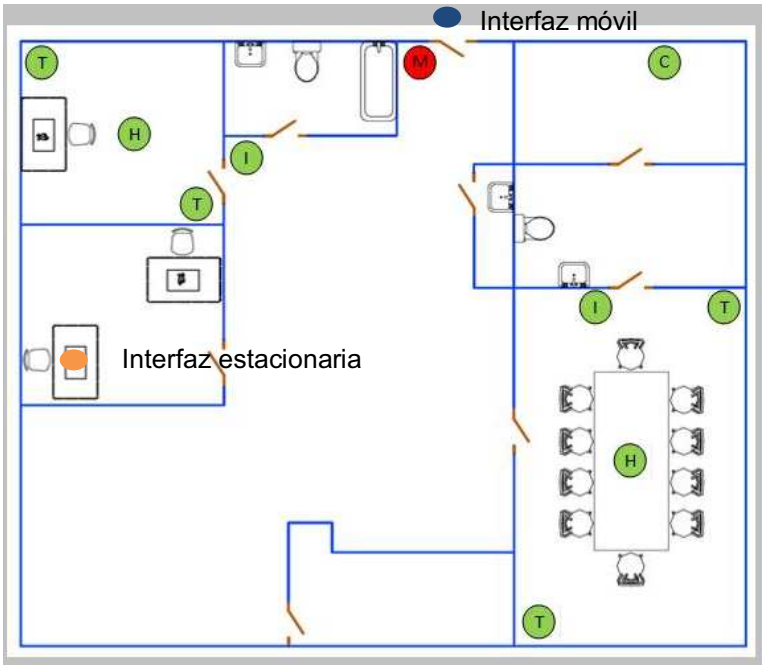


Figura 4-32.- Alcance de conexión de la interfaz movil

CAPÍTULO 5
CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Los sensores de temperatura de circuito integrado facilitan de gran manera la obtención de información, ya que requieren de poco o ningún tipo de acondicionamiento, a diferencia de sensores de temperatura como termopares o termistores que requieren de circuitos adicionales que deben ser calibrados para obtener datos fiables.

- Los electrodos de los detectores de presencia de agua que emplean la diferencia de potencial entre ellos para indicar presencia o ausencia de agua requieren estar muy cerca unos de otros. En tanto, los electrodos de detectores que se basan en la circulación de corriente para indicar presencia o ausencia de agua pueden ser colocados a una distancia mayor.

- Cuando se desea medir el consumo de corriente de cargas que causan distorsión en la forma de onda de corriente, se debe obtener el valor True-RMS de la misma, ya sea mediante un algoritmo de integración de la señal muestreada o a través de un elemento sensor que realice esta tarea y entregue una señal proporcional al valor True-RMS de la corriente.

- La señal de activación obtenida a la salida de los detectores de humo empleados en el sistema SECURE SYS debe ser acondicionada antes de poder ser ingresada al nodo maestro, ya que esta tiene como función original la activación de otros detectores de humo los cuales no requieren de una señal sostenida, ni de voltaje constante.

- Al utilizar un lazo de redundancia para asegurar la comunicación de datos, se debe comprobar que los niveles de voltaje en los puntos que se va a unir sean iguales, para así asegurar un correcto funcionamiento de los elementos de comunicación y evitar que estos se quemen.

- La creación de procesos paralelos en lenguaje C# demanda la distribución de las diferentes rutinas del programa tomando en cuenta cuáles de estas pueden ser susceptibles de ser interrumpidas y cuáles no.

- El manejo de dispositivos Bluetooth internos de celulares requiere el conocimiento de los procesos internos que realiza este para obtener información necesaria para crear un enlace con otros dispositivos Bluetooth.

- Sistemas de monitoreo de sensores inmóticos, como SECURE SYS, constituyen una herramienta útil para garantizar la integridad física no solo de inmuebles destinados a actividades laborales sino también de inmuebles destinados a la vivienda, además constituye un medio de seguridad para las personas que se encuentran dentro de estos.

- Gracias a que el sistema SECURE SYS cuenta con imágenes demostrativas del funcionamiento del sistema y mensajes de ayuda, crea un ambiente de manejo amigable, permitiendo que un nuevo usuario se adapte fácilmente a la utilización del mismo.

- La utilización de lenguajes de programación abiertos como Java y C# hace que la implementación del sistema SECURE SYS se realice sin complicaciones de compatibilidad en cualquier sistema operativo y distintos dispositivos móviles.

- La implementación de un sistema de monitoreo distribuido, en lugar de uno centralizado, facilita el diseño del software de control, la instalación, mantenimiento y ampliación del mismo, mas requiere el empleo de un

medio de comunicación que reduzca la pérdida y distorsión de los datos. Por tanto, de acuerdo a los resultados obtenidos en el presente proyecto, el estándar de comunicación RS-485 resultó ser una opción que satisface las necesidades de comunicación dentro distancias considerables ya que no se tuvo corrupción de datos.

- La implementación de una red de comunicación en configuración de anillo facilita la instalación de un sistema ya que la tarea de cableado no resulta tan extensa como sucedería en caso de implementar una configuración en estrella.

- El disponer de cableado de redundancia brinda al sistema SECURE SYS un grado de robustez ya que si un nodo esclavo es retirado de la red o uno de los cables presenta daños, el funcionamiento del resto del sistema no se verá interrumpido.

5.2 RECOMENDACIONES

- Para implementar una red RS-485 full dúplex se debe lograr una buena coordinación en la transmisión de datos de cada uno de los nodos. Caso contrario se puede tener pérdida de datos e incluso la avería de los circuitos de comunicación.

- El usuario encargado de determinar lo limites de alerta para las diferentes variables debe tener un conocimiento previo de los equipos que están funcionando dentro de las instalaciones y de las condiciones en que estos trabajan optimamente, así como de los requerimientos de energía de los mismos.

- Es recomendable esperar un lapso de 5 segundos entre consultas de información mediante la interfaz móvil, para que la transmisión sea exitosa.

- Para ampliar la capacidad de conectividad entre la interfaz móvil y la interfaz estacionaria se recomienda usar un dispositivo Bluetooth con mayor cobertura.

- Se debe mantener una cantidad razonable de saldo disponible en el chip del modem GSM, ya que de no ser así no se podrá realizar el envío de alertas por medio de mensajes de texto.

REFERENCIAS BIBLIOGRÁFICAS:

[1] MAYNÉ GRAU, Jordi. "Sistemas de Comunicaciones Industriales". Disponible en http://tec.upc.es/ie/practi/Sistemas.pdf

[2] Grupo de Investigación en Señales, Telemática y Comunicaciones, Universidad de Granada. "Capa física – Interfaz RS-232". Disponible en http://ceres.ugr.es/~alumnos/redrs232/fisica.htm

[3] FORTUNE CITY. "TUTORIAL DEL RS-232 C". Disponible en http://members.fortunecity.es/davidweb2/esquemas/rs-232.htm

[4] GONZÁLEZ, Agustín. "Comunicación Asíncrona Local (RS-232)", pg 38. Disponible en http://profesores.elo.utfsm.cl/~agv/elo309/lectures/p2p.pdf

[5] WIKIPEDIA. "RS-485". Disponible en http://es.wikipedia.org/wiki/RS-485

[6] ARIAN, CONTROL & INSTRUMENTATION. "Comunicaciones Seriales Vía RS-485". Disponible en http://www.arian.cl/downloads/cat-cl-ape-s.pdf

[7] LAMMER BIES. "RS485 serial information". Disponible en http://www.lammertbies.nl/comm/info/RS-485.html

[8] SINEMED SYSTEMS. "¿Qué es MySQL?". Disponible en http://www.sinemed.com/recursos/docs/MySQL.pdf

[9] ENRÍQUEZ TOLEDO, Alma; MALDONADO AYALA, Jesús; NAKAMURA ORTEGA, Yunko; NOGUERON TOLEDO, Goretty. "MY SQL". Disponible en http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf

[10] RECIO, Francisco; PROVENCIO, David. DESARROLLOWEB. ".NET Framework". Disponible en http://www.desarrolloweb.com/articulos/1328.php

[11] MSDN. "Visual C#". Disponible en http://msdn.microsoft.com/en-us/library/kx37x362.aspx

[12] URRIELLUNET. "¿Por qué C#?". Disponible en http://urriellu.net/es/articles-software/csharp-advantages.html

[13] WIKIPEDIA. "C Sharp". Disponible en http://es.wikipedia.org/wiki/C_Sharp

[14] ANONIMO. "COMANDOS AT". Disponible en http://www.google.com/url?sa=t&source=web&cd=1&sqi=2&ved=0CBwQFjAA&url=http%3A%2F%2Falarmagsm.googlecode.com%2Ffiles%2FCOMANDOS%2520AT.doc&rct=j&q=comandos%20at&ei=UeqQTtmKFcnt0gGQ_rQu&usg=AFQjCNFwrx6U6kpSOGftmpB0GxeVt_m4yA&cad=rja

[15] NETBEANS. “Bienvenido a NetBeans y www. netbeans.org”. Disponible en http://netbeans.org/index_es.html

[16] CIBERAULA. “Introducción a J2ME”. Disponible en http://java.ciberaula.com/articulo/introduccion_j2me

[17] WEBTELLER. Qué es Java”. Disponible en http://www.webtaller.com /construccion/lenguajes/java/lecciones/que-es-java.php

[18] MEZA RAMÍREZ, Julio César. “Programación para dispositivos móviles en J2ME, Conceptos Básicos II”.
Disponible en http://ingjmeza.110mb.com/j2me2.pdf

[19] SLIDE5HARE. “J2M3”, pg 3.
Disponible en http://www.slideshare.net/mileti/j2me-3971635

[20] DINASTIA SOFT. “BASCOM AVR”.
Disponible en http://www.dinastiasoft.com.ar/bascomavr.htm

[21] CEDOM. “Qué es domótica”.
Disponible en: http://www.cedom.es/que-es-domotica.php

[22] SEVILLA MARTÍN, Juan Pablo; GARCÍA SÁNCHEZ Pablo. “Ulfsark Comunicación Entre Dispositivos Bluetooth”. Universidad de Granada. 2006
Disponible en: http://ulfsark.sourceforge.net/ulfsarkdocweb.pdf

[23] ALMEIDA ORDOÑEZ, José Andrés ;BARRERA QUEZADA, Wilson Alejandro. “Características de servicio de mensajería corta (SMS) y obtención de datos del servicio SMS en la ciudad de Cuenca”.Universidad Politécnica Salesiana. 2006.
Disponible en: http://dspace.ups.edu.ec/bitstream/123456789/200/3/ Capitulo%201.pdf.

[24] GONZALES GOMEZ, Juan.UAM.2002 “El Servicio SMS: un enfoque práctico.” Disponible en: http://www.iearobotics.com/personal/juan/doctorado/ sms/sms.pdf.

[25] IGLUPPIWEB. “Apuntes de Correo Electrónico.” Disponible en: http://tutoriales.igluppiweb.com.ar/ecorreo.pdf

[26] WIKIPEDIA. “Asymmetric Digital Subscriber Line”. Disponible en: http://es.wikipedia.org/wiki/Asymmetric_Digital_Subscriber_Line

[27] ECAMEC TECNOLOGÍA. “Sensores de Corriente, comparativa de tecnologías”. 2009.

Disponible en: http://www.ecamec.com/newsletter/bajarnotab0509.pdf

[28] ANONIMO. “Sensores y Actuadores Domóticos”. Disponible en: www.dtf.fi.upm.es/~noedelta/DEI/DEI-Tema6-0910.ppt

[29] PLASNAVI. “Detectores”. Disponible en: http://www.supercontrols.com.ar/productos/detector_humo.html

[30] CLICK7DEF BLOG. “¿Qué es XAMPP y para qué sirve?”. Disponible en http://www.click7def.com/que-es-xampp-y-para-que-sirve

[31] TARINGA. ““¿Qué es XAMPP?”. Disponible en http://www.taringa.net/posts/info/7522503/_Que-Es-Xampp_.html

[32] PHP. “What is PHP”. Disponible en http://www.php.net/manual/en/intro-whatis.php

[33] ALVAREZ, Miguel Angel. DESARROLLOWEB. “phpMyAdmin”. Disponible en http://www.desarrolloweb.com/articulos/844.php

[34] PERUCREATIVO. “Que es FTP y como utilizarlo”. Disponible en http://www.perucreativo.com/articulo-8-Consejos-para-tener-un-sitio-web-exitoso.html

[35] NAVAS, Eduardo. ”Tecnología CDMA”. Escuela Técnica Industrial “Joaquín Avellán”.2003. Disponible en: http://www.monografias.com/trabajos13/modu/modu.shtml

[36] MATEO,Ramón. “Transductores de temperatura.” Disponible en: http://www.monografias.com/trabajos3/transductores/transductores.shtml

[37] WESLWY, Anthony.” Temperature Logger for 10" Newtonian telescope”. 2005. Disponible en: http://acquerra.com.au/astro/equipment/temp-logger

[38] WIKIPEDIA.”AVR”. Disponible en: http://es.wikipedia.org/wiki/AVR

[39] ATMEL. “ATmega164P Datasheet”.
Disponoble en: moon-20.googlecode.com/files/ATmega164p_guide.pdf

[40] ATMEL. “ATmega48 Datasheet”. Disponoble en:
http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf

[41] WIKIPEDIA. “Complementary metal oxide semiconductor”. Disponible en: http://es.wikipedia.org/wiki/Complementary_metal_oxide_semiconductor

[42] WIKIPEDIA.”RISC”. Disponible en: http://es.wikipedia.org/wiki/RISC

[43] WIKIPEDIA. “MIPS”. Disponible en: http://es.wikipedia.org/wiki/MIPS

[44] MAXIM. “MAX1487 - MAX491”. Disponible en: http://www.datasheetcatalog.org/datasheet/maxim/MAX1487-MAX491.pdf

[45] MAXIM. “SN75176A DIFFERENTIAL BUS TRANSCEIVER”. Disponible en http://www.datasheetcatalog.org/datasheets/208/492412_DS.pdf

[46] I-MICRO. “Ingeniería en Microcontroladores, Protocolo RS-485”. Disponible en http://www.i-micro.com/pdf/articulos/rs-485.pdf

[47] FAIRCHILD SEMICONDUCTOR. “KA78XX/KA7800A”. Disponible en http://www.datasheetcatalog.org/datasheets/228/390068_DS.pdf

[48] Moreno Velasco Ignacio. “Aislamiento”, Universidad de Burgos. Disponible en http://www2.ubu.es/ingelec/tecelec/inaki/Instelec/docsinstrum/Apuntes/5-AISLAMIENTO_v6-0PW.pdf

[49] PHILIPS. “4-to-16 line decoder/demultiplexer 74HC/HCT154”. Disponible en http://www.datasheetcatalog.org/datasheet/philips/74HC_HCT154_CNV_2.pdf

[50] WIKIPEDIA. “Universal asynchronous receiver/transmitter”. Disponible en http://en.m.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter

[51] LEM. “AKR Split core Current tranducer, 50A”. Diponible en: http://www.lem.com/hq/en/component/option,com_catalog/task,displaymodel/id,F2.16.25.000.0/

[52] LEM. “AK & AKR Installation Guide.” Disponible en: http://www.lem.com/docs/manuals/AK%26AKR%20series.pdf

[53] MAXIM. “DS18B20 Programmable Resolution 1-Wire Digital Thermometer”. Disponible en: http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf

[54] ÖGREN, Joakin. “Basic circuit blocks”. Disponible en: http://ciecfie.epn.edu.ec/hwb/ci_ActiveFilterButterworth2L.html

[55] HONORABLE CAMARA DE DIPUTADOS DE SANTA CRUZ. “El amplificador operacional: Fundamentos y aplicaciones prácticas.” Disponible en: http://www.hcdsc.gov.ar/biblioteca/ises/tecnologia/informatica/amplificador%20operacional.pdf

[56] HUMIREL. “Sensor de humedad relativa”. Disponible en: http://www.todopic.com.ar/utiles/hs1100es.pdf

[57] WIKIPEDIA. “Comprobación de redundancia cíclica.” Disponible en: http://es.wikipedia.org/wiki/Comprobaci%C3%B3n_de_redundancia_c%C3%ADclica

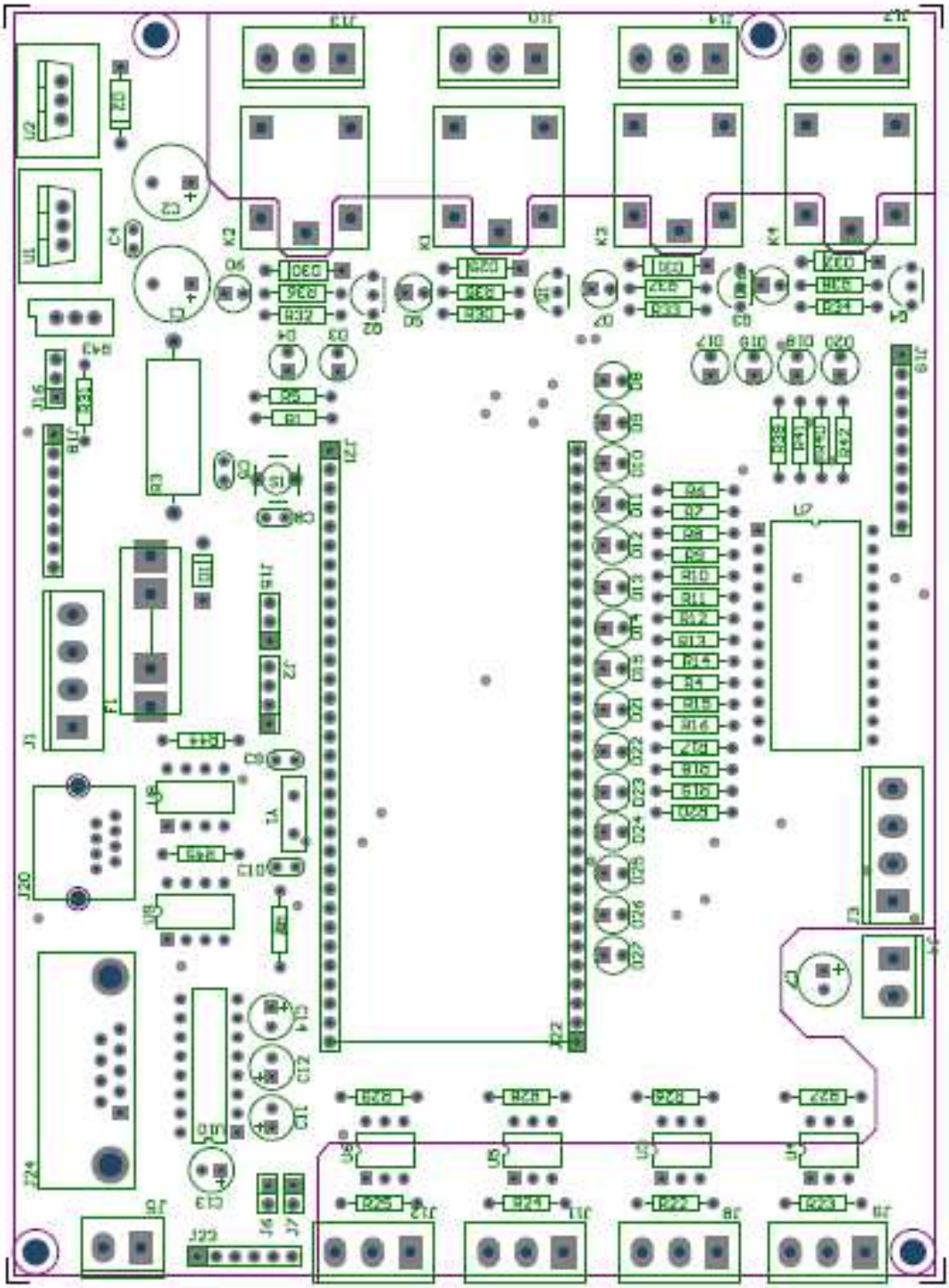
[58] PCPAUDIO. “Ruidos en Amplificadores”. Disponible en: http://www.pcpaudio.com/pcpfiles/doc_amplificadores/Bajoruido/Bajoruido.html

[59] RUMEU, Fragoso; LÓPEZ, Tomás. “RUIDO”. Disponible en: http://www.google.com.ec/url?sa=t&rct=j&q=ruido%20que%20afecta%20a%20circuitos%20electricos&source=web&cd=10&sqi=2&ved=0CGkQFjAJ&url=http%3A%2F%2Fwww.isaatc.ull.es%2Fasignaturas%2FIE1%2Fpresentacion%2520alumnos%2FRUIDO%2520electronico%2520Tomas%2520.ppt&ei=uQ3GTqexMcnZgQeA18VL&usg=AFQjCNF744i8JMtZoXCrqw_dTaD43lFK8A&sig2=JT0vm4itlvh_rzQigZni3w&cad=rja

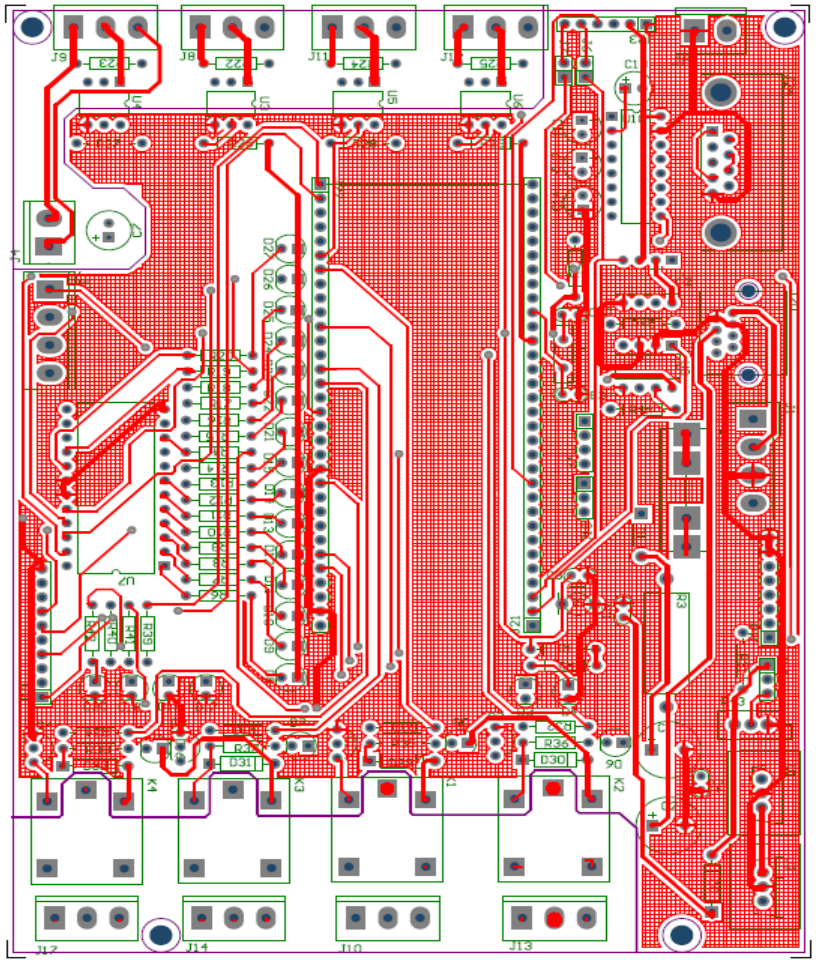
[60] HOSPITAL DE TALCA. “Definiciones”. Disponible en: http://www.hospitaldetalca.cl/documentos/definiciones.pdf

ANEXO A

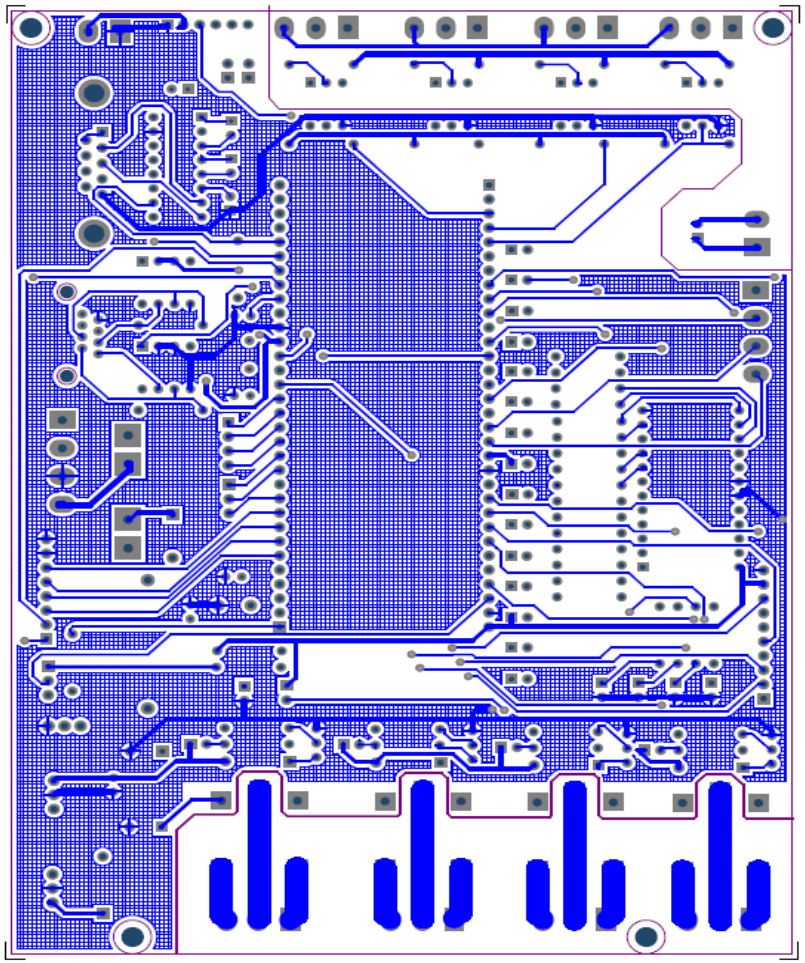
TARJETA DEL NODO MAESTRO:



RUTEADO DE LA TARJETA DEL NODO MAESTRO:



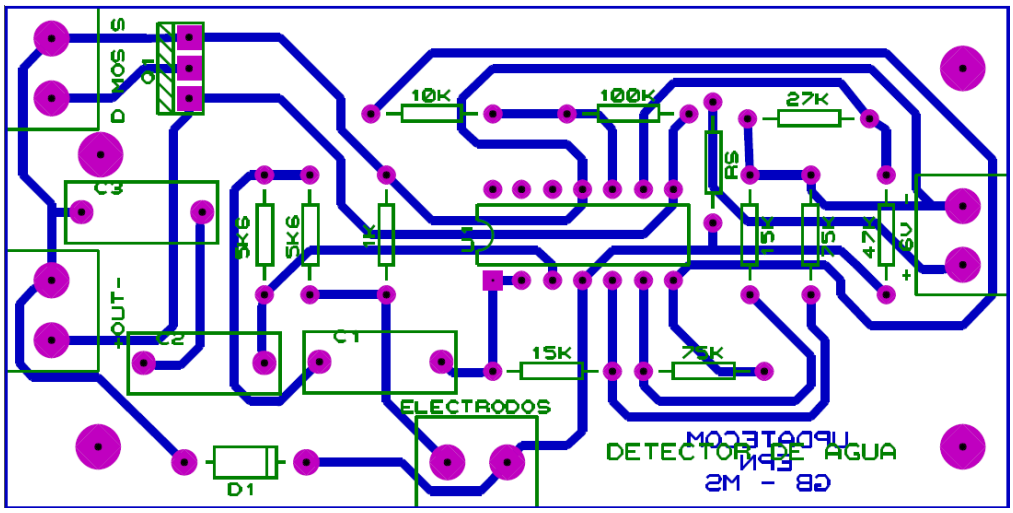
TOP SCREEN



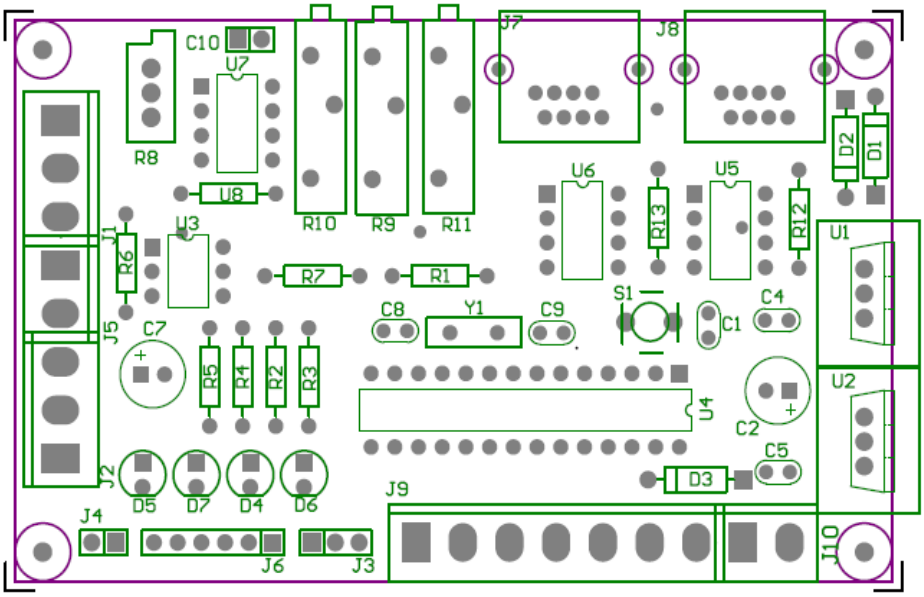
BOTTOM SCREEN

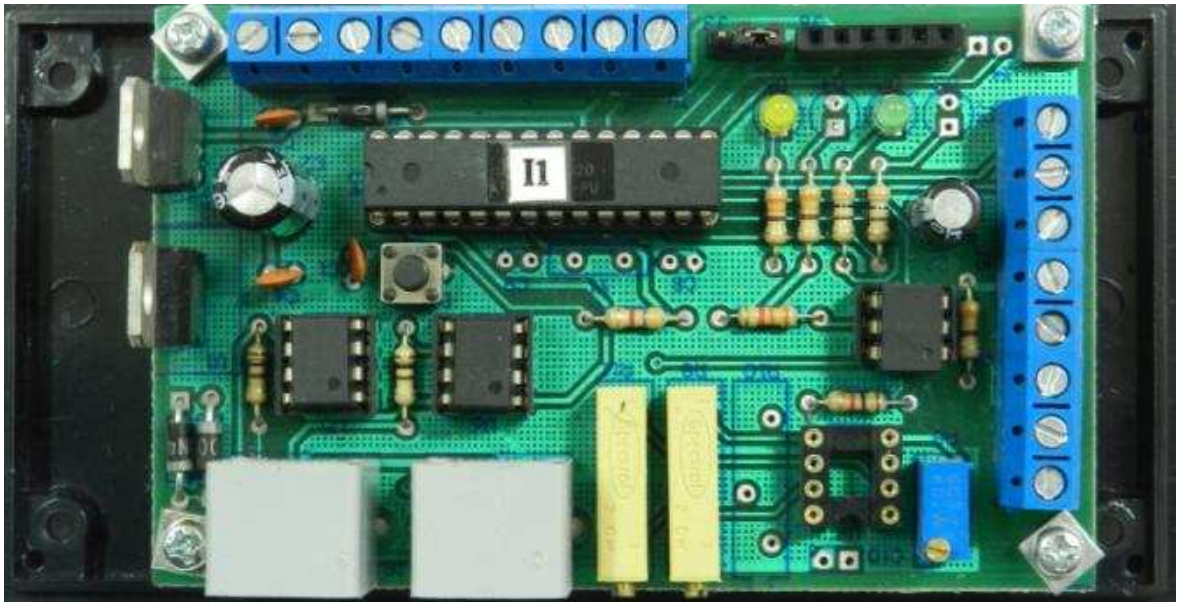
TARJETA DEL DETECTOR DE AGUA:

RUTEADO DE LA TARJETA DEL DETECTOR DE AGUA:

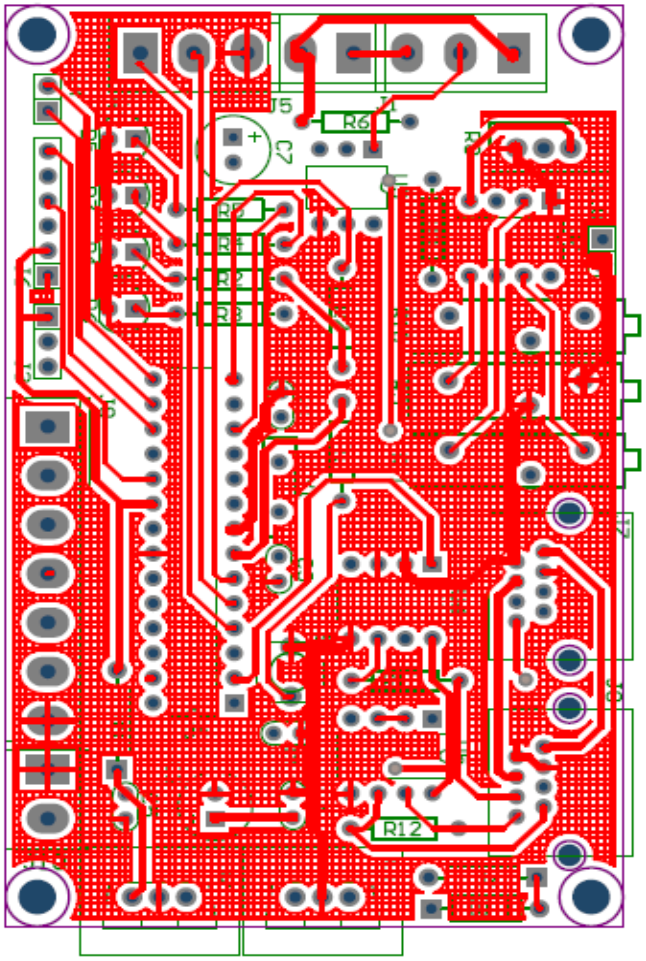


TARJETA DE ADQUISICIÓN DE DATOS DEL NODO DE PRESENCIA DE AGUA:

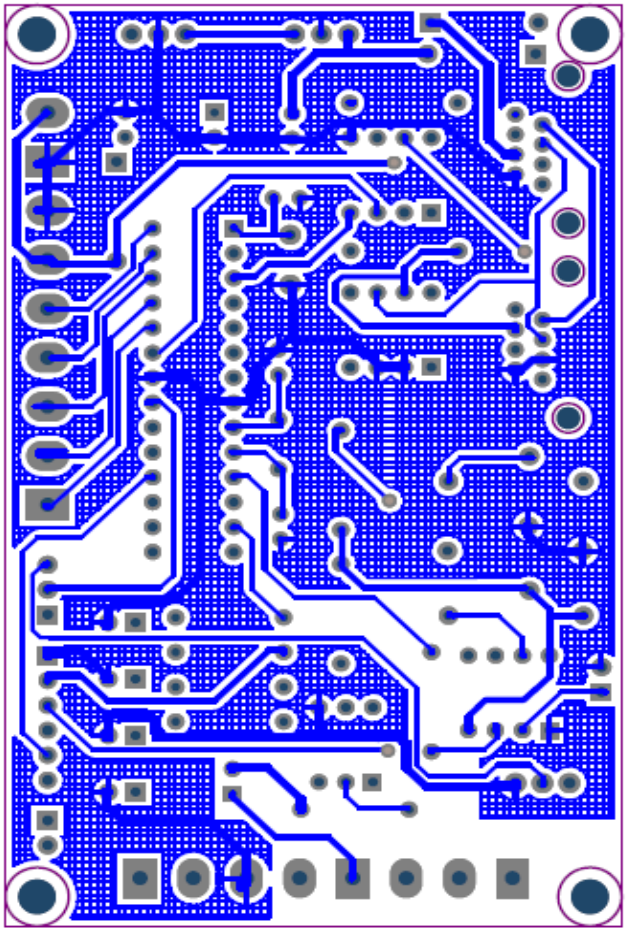




RUTEADO DE LA TARJETA DE ADQUISICIÓN DE DATOS DEL NODO DE PRESENCIA DE AGUA:

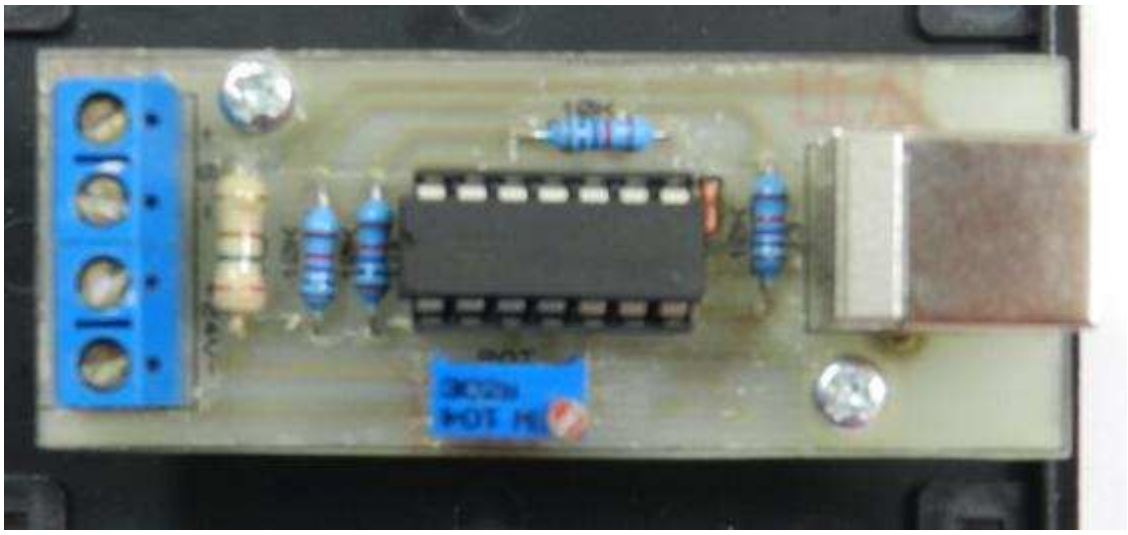


TOP SCREEN

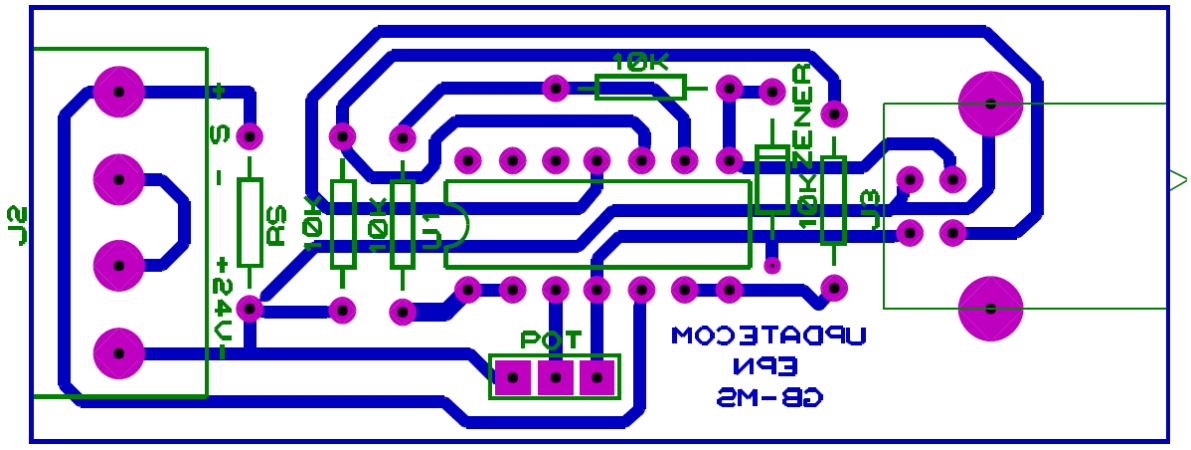


BOTTOM SCREEM

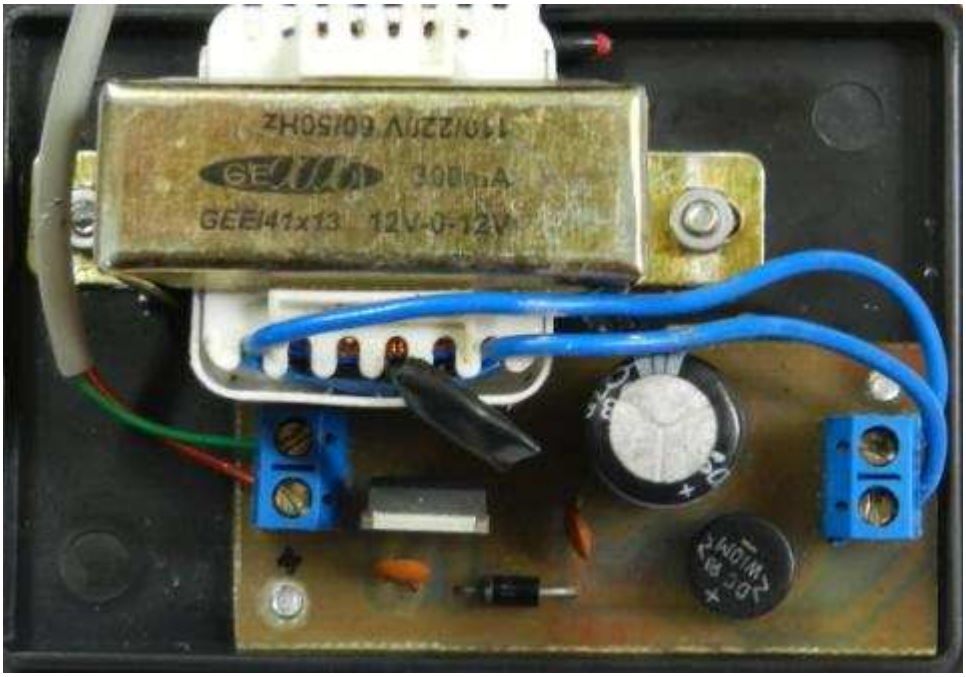
TARJETA DE ACONDICIONAMIENTO DE SEÑAL PARA EL NODO DE CORRIENTE:



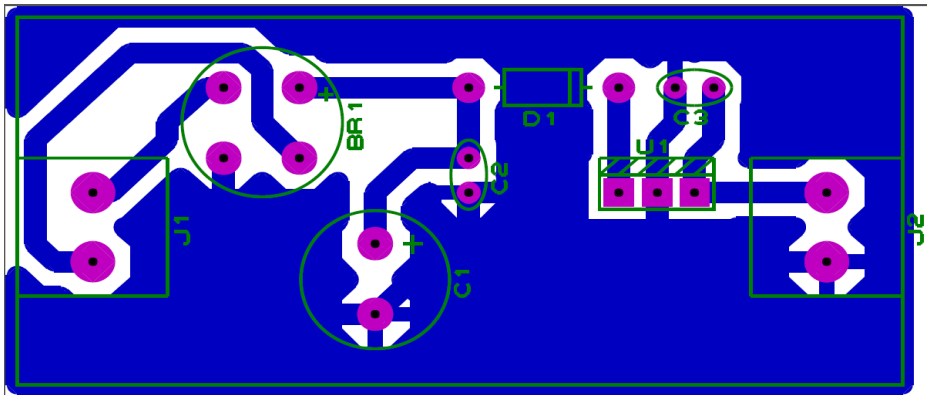
RUTEADO DE LA TARJETA DE ACONDICIONAMIENTO DE SEÑAL PARA EL NODO DE CORRIENTE:



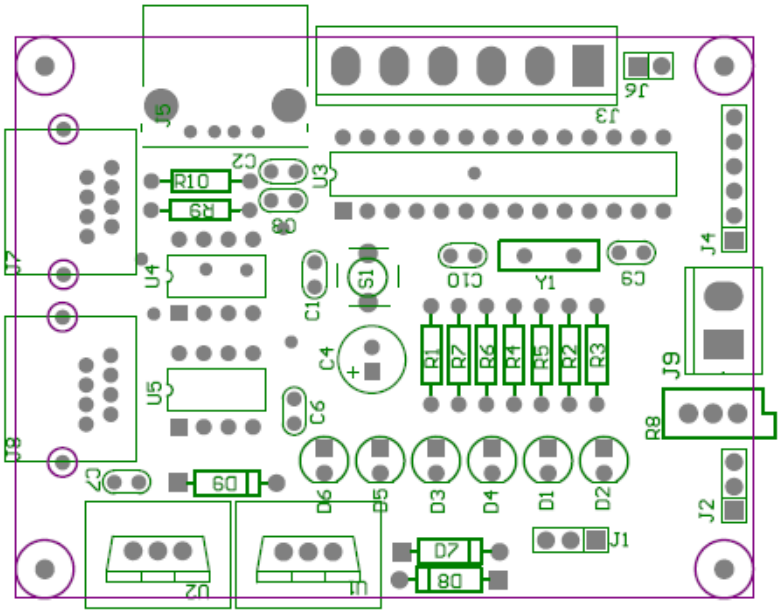
FUENTE DE ALIMENTACIÓN DE 24VDC PARA EL SENSOR DE CORRIENTE:

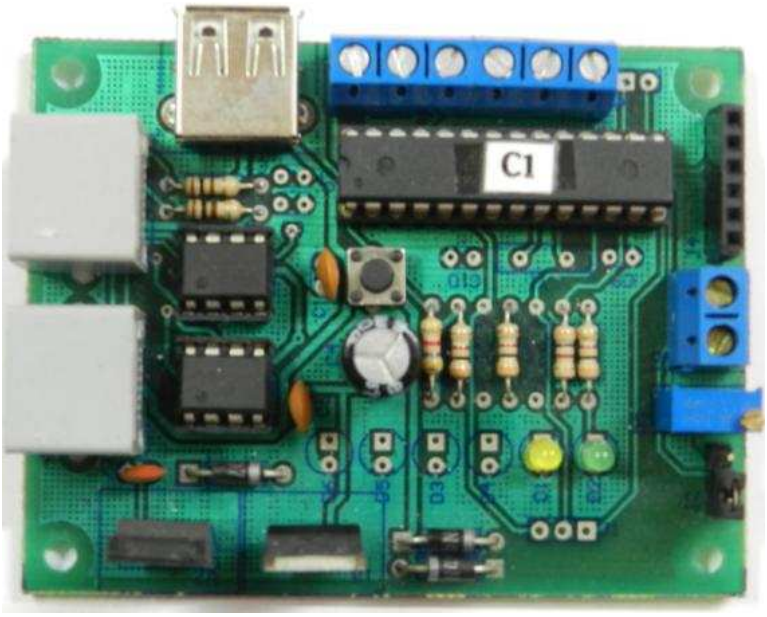


RUTEADO DE LA FUENTE DE ALIMENTACIÓN DE 24VDC PARA EL SENSOR DE CORRIENTE:

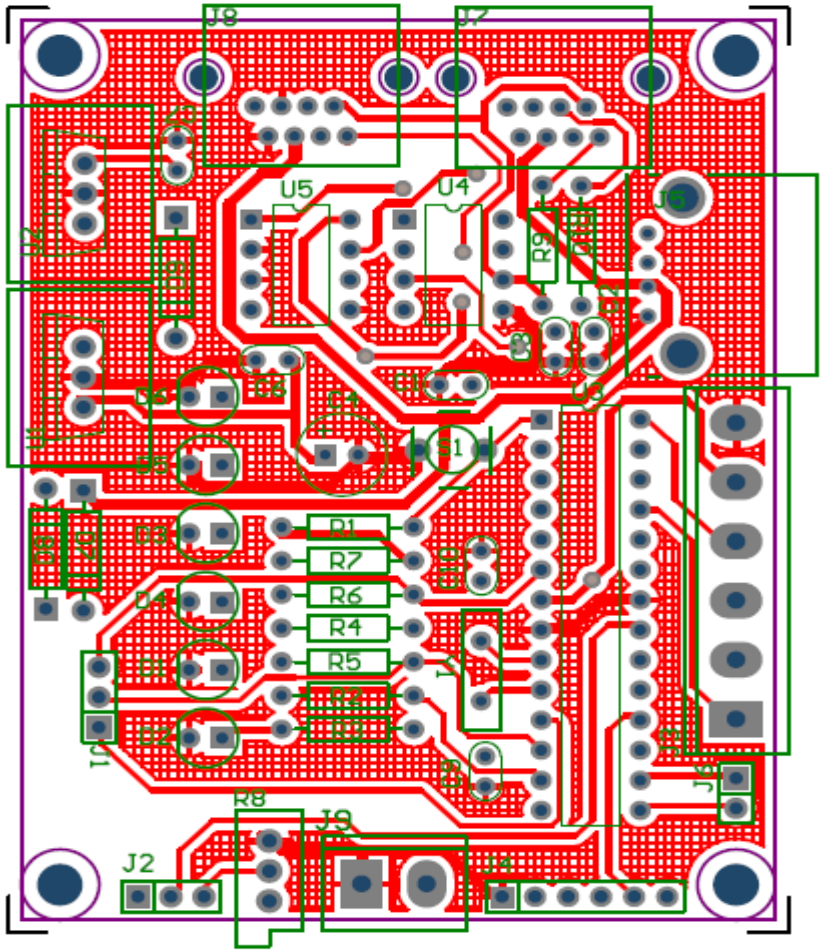


TARJETA DE ADQUISICIÓN DE DATOS DEL NODO DE CORRIENTE:

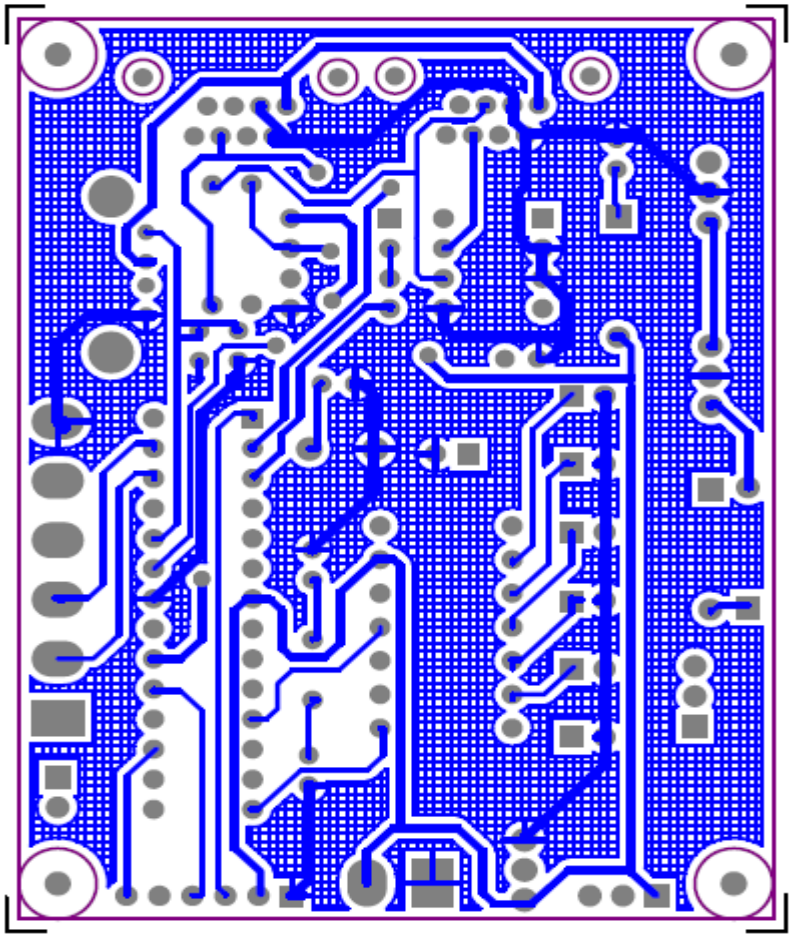




RUTEADO DE LA TARJETA DE ADQUISICIÓN DE DATOS DEL NODO DE CORRIENTE:

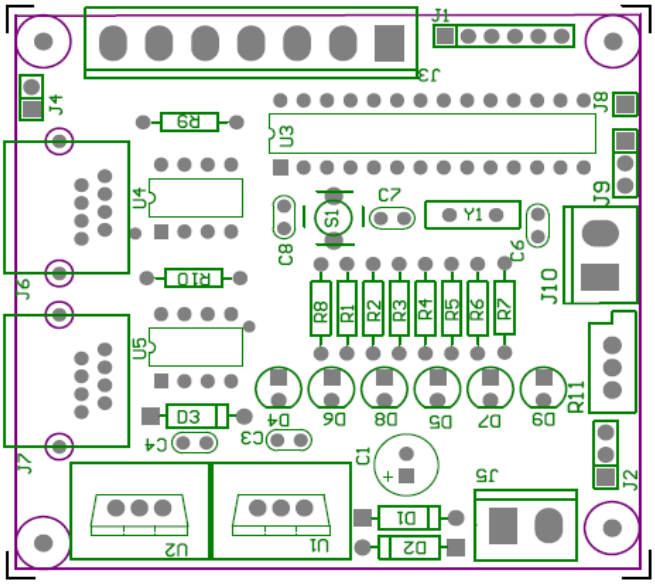


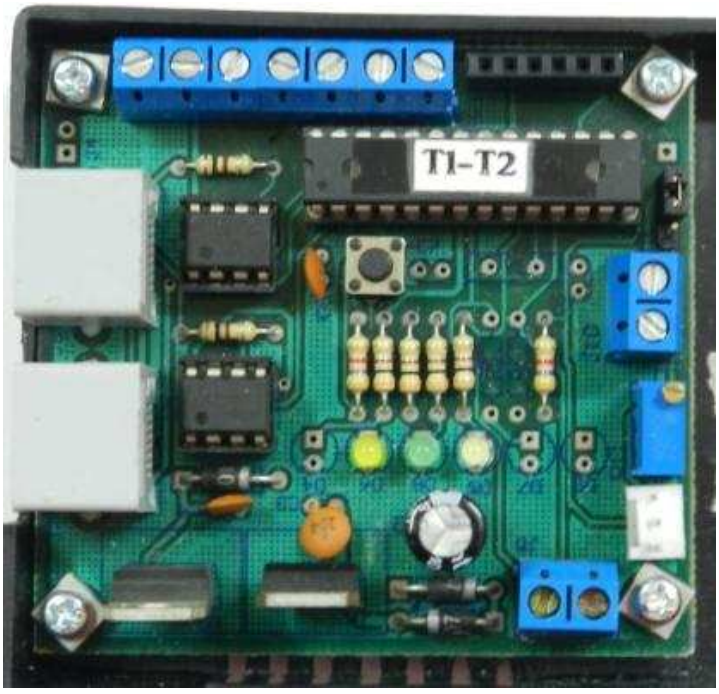
TOP SCREEN



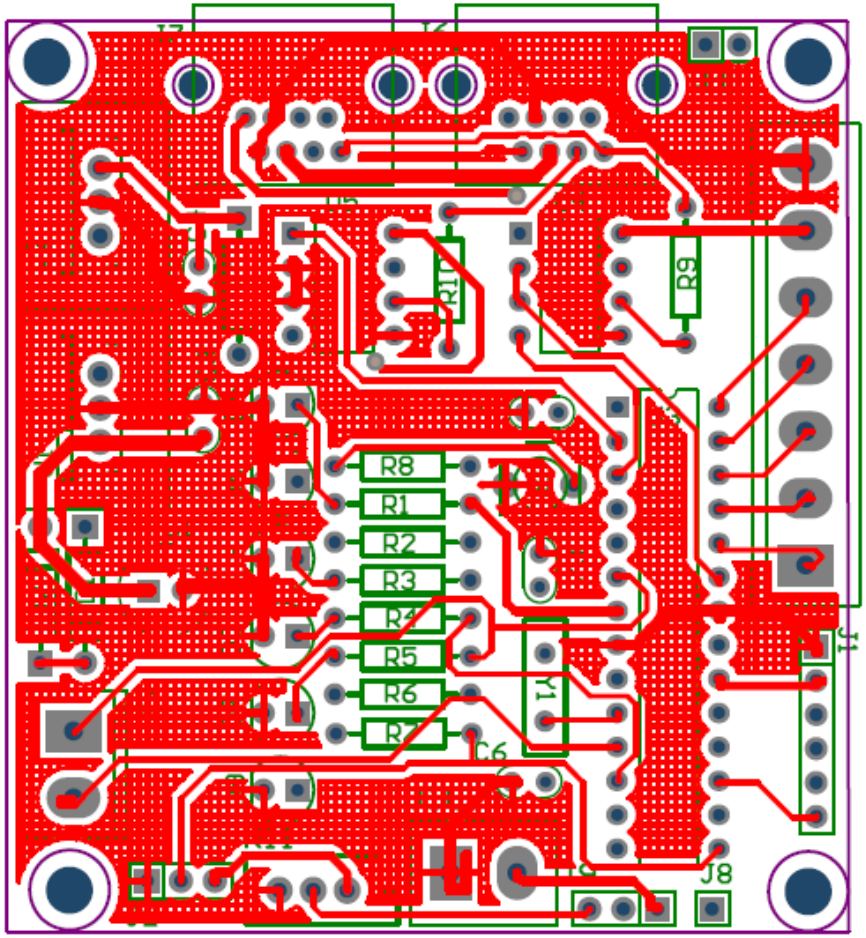
BOTTOM SCREEN

TARJETA DE ADQUISICIÓN DE DATOS DEL NODO DE TEMPERATURA:

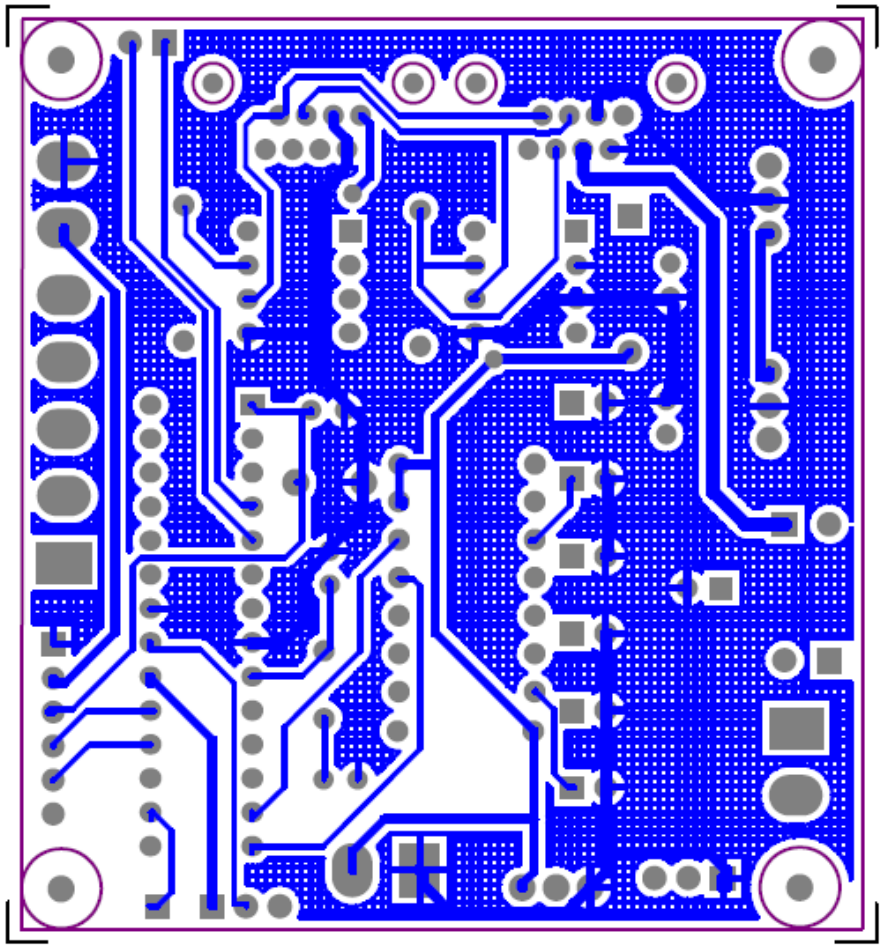




RUTEADO DE LA TARJETA DE ADQUISICIÓN DE DATOS DEL NODO DE TEMPERATURA:



TOP SCREEN



BOTTOM SCREEN

ANEXO B

CÓDIGO FUENTE DE LOS NODOS:

NODO MAESTRO:

```
$regfile = "m164pdef.dat"
$crystal = 8000000
$baud = 9600
$baud1 = 9600
$hwstack = 32
$swstack = 10
$framesize = 40

'Dimensionamiento de variables:
Dim Sensorm As Bit
Dim Sensorh As Bit
Dim T1 As String * 10
Dim T2 As String * 10
Dim T3 As String * 10
Dim T4 As String * 10
Dim C1 As String * 10
Dim U1 As String * 10
Dim U2 As String * 10
Dim I1 As String * 10
Dim I2 As String * 10
Dim Aux As String * 10

Dim Trama As String * 10
Dim Trama2 As String * 10
Dim Tramap As String * 10
Dim Frame1 As String * 10

Const Cmaxchar = 20

Dim Buf As String * Cmaxchar
Dim Buf1 As String * Cmaxchar
Dim D As Byte
Dim D1 As Byte
Dim Pausa As Byte
Dim Cuenta As Byte
Dim Permiso As Bit
Dim Auxi As Byte
Dim Tiemp As Word
Dim Tiemptemp As Word
Dim Cuentahumo As Long
Dim Cuentahumo2 As Long
Dim Cuentaapagado As Word
Dim Cuentaapagado2 As Word
Dim Alarmhumo As Bit
Dim Alarmhumo2 As Bit

Dim Lon As Word
Dim Num As Word
Dim Num2 As Word
Dim Descolgar As Long


'Configuración timer:
'Timer1:
Tiemp = 3000
Tiemptemp = 3000

Config Timer1 = Timer , Prescale
= 64
Stop Timer1
Timer1 = Tiemptemp

'Timer2:
Config Timer2 = Timer , Prescale
= 1024
Stop Timer2
Timer2 = 0

'Configuración interrupciones:
On Ovf1 Inttiempo
On Ovf2 Intpuerta
On Urxc Rec_isr
On Urxc1 Rec_isr1
Enable Ovf1
Enable Ovf2
Enable Urxc
Enable Urxc1

Enable Interrupts

Waitms 10


'Configuración puerto com:

Open "com2:" For Binary As #1
Open "com1:" For Binary As #0

'Puertos para sensores: magnético
, humo.
Config Portd.7 = Input
Config Portc.1 = Input
Config Portc.2 = Input

M1 Alias Pind.7
H1 Alias Pinc.1
H2 Alias Pinc.2


'Pines para señalización:
Config Portc.4 = Output
Config Portc.5 = Output
Config Portc.6 = Output
Config Portc.7 = Output
Config Porta.7 = Output
Config Portc.3 = Output

Puerta Alias Portc.4
```

```
Pin1 Alias Portc.5
Pin2 Alias Portc.6
Pin3 Alias Portc.7
Pin4 Alias Porta.7
Magnetico Alias Portc.3

'Inicialización variables:
Auxi = 0
Num = 0
Permiso = 1
Puerta = 0
Pin1 = 0
Pin2 = 0
Pin3 = 0
Pin4 = 0
Cuentahumo = 0
Cuentahumo2 = 0
Alarmhumo = 0
Alarmhumo2 = 0
Descolgar = 0


'PROGRAMA:
Print "1ST"
Wait 1


Do
   Incr Descolgar
   If Permiso = 1 Or Descolgar >=
 100000 Then
      Descolgar = 0
      Gosub Cargar
      Start Timer1
      Permiso = 0

   End If

   Gosub Vermh
Loop

End


Vermh:

If M1 = 1 Then
Magnetico = 1
Elseif M1 = 0 Then
Magnetico = 0
End If

If H1 = 0 Then

Incr Cuentahumo
Cuentaapagado = 0
Alarmhumo = 1
Elseif H1 = 1 Then

Incr Cuentaapagado
   If Cuentahumo >= 100000 Or Cue
ntaapagado >= 25000 Then

    Alarmhumo = 0
    Cuentahumo = 0
    Cuentaapagado = 0
    End If
End If

If H2 = 0 Then

Incr Cuentahumo2
Cuentaapagado2 = 0
Alarmhumo2 = 1
Elseif H2 = 1 Then

Incr Cuentaapagado2
    If Cuentahumo2 >= 100000 Or C
uentaapagado2 >= 25000 Then

    Alarmhumo2 = 0
    Cuentahumo2 = 0
    Cuentaapagado2 = 0
    End If
End If

Return

Cargar:
Incr Auxi
Select Case Auxi
Case 1 : Frame1 = "1ST"
Case 2 : Frame1 = "2ST"
Case 3 : Frame1 = "3ST"
Case 4 : Frame1 = "4ST"
Case 5 : Frame1 = "1SC"
Case 6 : Frame1 = "1SU"
Case 7 : Frame1 = "2SU"
Case 8 : Frame1 = "1SI"
Case 9 : Frame1 = "2SI"
          Auxi = 0
Case Else : Auxi = 0
End Select
Print Frame1

'Codificación señalización (Maste
r pide a esclavos):
         Pin1 = 1
         Pin2 = 0
         Pin3 = 0
         Pin4 = 1
Return

Error2:
Select Case Frame1
Case "1ST":
   T1 = "1STERR2"
Case "2ST":
   T2 = "2STERR2"
Case "3ST":
```

```
   T3 = "3STERR2"
Case "4ST":
   T4 = "4STERR2"
Case "1SC":
   C1 = "1SCERR2"
Case "1SU":
   U1 = "1SUERR2"
Case "2SU":
   U2 = "2SUERR2"
Case "1SI":
   I1 = "1SIERR2"
Case "2SI":
   I2 = "2SIERR2"
Case Else
End Select
Return


Inttiempo:

Stop Timer1
If Auxi < 5 Then
   Timer1 = Tiemptemp
Else
   Timer1 = Tiemp
End If
   Gosub Error2
   Permiso = 1
Return

Intpuerta:
Incr Cuenta
   If Cuenta = 5 Then
      Stop Timer2
      Puerta = 0
      Cuenta = 0
   End If
Timer2 = 0
Return

'Recepción esclavos:
Rec_isr:
  D = Udr

  If D = 13 Then

      Stop Timer1

      If Auxi < 5 Then
         Timer1 = Tiemptemp
      Else
         Timer1 = Tiemp
      End If

      Permiso = 1

      Lon = Len(buf)


     For Num = 1 To Lon

Trama = Mid(buf , Num , 3)
Trama = Left(trama , 3)


 If Trama = "1ST" Then
   Aux = Mid(buf , Num , 7)

   If Aux = "1ST" Then
   T1 = "1STERR1"

   Elseif Aux = "1STERR1"
   Then
   T1 = "1STERR1"

   Else
   T1 = Aux

'Codificación señalización:
   Pin1 = 1
   Pin2 = 0
   Pin3 = 0
   Pin4 = 0

   End If

 End If

 If Trama = "2ST" Then
   Aux = Mid(buf , Num , 7)

   If Aux = "2ST" Then
   T2 = "2STERR1"

   Elseif Aux = "2STERR1"
   Then
   T2 = "2STERR1"

   Else
   T2 = Aux

'Codificación señalización:
   Pin1 = 0
   Pin2 = 1
   Pin3 = 0
   Pin4 = 0

   End If
 End If

 If Trama = "3ST" Then
   Aux = Mid(buf , Num , 7)

   If Aux = "3ST" Then
   T3 = "3STERR1"

   Elseif Aux = "3STERR1"
   Then
   T3 = "3STERR1"
```

```
   Else
   T3 = Aux

'Codificación señalización:
   Pin1 = 1
   Pin2 = 1
   Pin3 = 0
   Pin4 = 0

   End If

 End If

 If Trama = "4ST" Then
   Aux = Mid(buf , Num , 7)

   If Aux = "4ST" Then
   T4 = "4STERR1"

   Elseif Aux = "4STERR1"
   Then
   T4 = "4STERR1"

   Else
   T4 = Aux

'Codificación señalización:
   Pin1 = 0
   Pin2 = 0
   Pin3 = 1
   Pin4 = 0
   End If

 End If

 If Trama = "1SC" Then
   C1 = Mid(buf , Num , 7)

'Codificación señalización:
   Pin1 = 1
   Pin2 = 0
   Pin3 = 1
   Pin4 = 0

 End If


 If Trama = "1SU" Then
   U1 = Mid(buf , Num , 7)

'Codificación señalización:
   Pin1 = 0
   Pin2 = 1
   Pin3 = 1
   Pin4 = 0

 End If

 If Trama = "2SU" Then
   U2 = Mid(buf , Num , 7)

'Codificación señalización:
   Pin1 = 1
   Pin2 = 1
   Pin3 = 1
   Pin4 = 0

 End If


 If Trama = "1SI" Then
   I1 = Mid(buf , Num , 7)

'Codificación señalización:
   Pin1 = 0
   Pin2 = 0
   Pin3 = 0
   Pin4 = 1

 End If


 If Trama = "2SI" Then
   I2 = Mid(buf , Num , 7)

'Codificación señalización:
   Pin1 = 0
   Pin2 = 1
   Pin3 = 0
   Pin4 = 1

   End If

 Next Num

     Buf = Buf + Chr(0)
     Buf = ""


 Else
    Buf = Buf + Chr(d)
 End If
Salir:
Return

'Recepción Computador
Rec_isr1:
  D1 = Udr1

'Codificación señalización(Master
 recibe del computador):
        Pin1 = 0
        Pin2 = 1
        Pin3 = 1
        Pin4 = 1

  If D1 = 13 Then
      Lon = Len(buf1)
```

```
   Num2 = 1
 For Num2 = 1 To Lon

Trama2 = Mid(buf1 , Num2 , 3)
Tramap = Mid(buf1 , Num2 , 7)
   Trama2 = Left(trama2 , 3)
   Tramap = Left(tramap , 7)

   If Tramap = "1SV0001" Then
   Print #1 , "OK"
   Goto Regreso
   End If

   If Trama2 = "1ST" Then
   Gosub Signal
   Print #1 , T1
   Goto Regreso
   End If

   If Trama2 = "2ST" Then
   Gosub Signal
   Print #1 , T2
   Goto Regreso
   End If

   If Trama2 = "3ST" Then
   Gosub Signal
   Print #1 , T3
   Goto Regreso
   End If

   If Trama2 = "4ST" Then
   Gosub Signal
   Print #1 , T4
   Goto Regreso
   End If

   If Trama2 = "1SC" Then
   Gosub Signal
   Print #1 , C1
   Goto Regreso
   End If

   If Trama2 = "1SU" Then
   Gosub Signal
   Print #1 , U1
   Goto Regreso
   End If

   If Trama2 = "2SU" Then
   Gosub Signal
   Print #1 , U2
   Goto Regreso
   End If

   If Trama2 = "1SI" Then
   Gosub Signal
   Print #1 , I1
   Goto Regreso
   End If

   If Trama2 = "2SI" Then
   Gosub Signal
   Print #1 , I2
   Goto Regreso
   End If

   If Trama2 = "1SM" Then
   Gosub Signal
     If M1 = 0 Then
     Sensorm = 0
     Elseif M1 = 1 Then
     Sensorm = 1
     End If
   Print #1 , "1SM000" + Str(
sensorm)
   Goto Regreso
   End If

   If Trama2 = "1SH" Then
   Gosub Signal
   Print #1 , "1SH000" + Str(
alarmhumo)
   Goto Regreso
   End If

   If Trama2 = "2SH" Then
   Gosub Signal
   Print #1 , "2SH000" + Str(
alarmhumo2)
   Goto Regreso
   End If

   If Tramap = "1AP0001" Then
   Gosub Signal
   Puerta = 1
   Cuenta = 0
   Start Timer2
   Goto Regreso
   End If

 Next Num2
Regreso:

     Buf1 = Buf1 + Chr(0)
     Buf1 = ""


 Else
    Buf1 = Buf1 + Chr(d1)
 End If
Salida:
Return


Signal:
'Codificación señalización(Master
 envia al computador):
     Pin1 = 1
     Pin2 = 0
```

```
        Pin3 = 1
        Pin4 = 1
```

```
Return
```

NODO TEMPERATURA DEPARTAMENTO TÉCNICO:

```
$regfile = "m48def.dat"
$crystal = 8000000
$baud = 9600

'Declaración funciones:
Declare Sub Temperatura
Declare Sub Enviar
Declare Function Decigrades(byval
 Sc(9) As Byte) As Integer

'Declaración variables
temperatura:
Dim Uno As Byte
Dim Dos As Byte
Dim Ctg1 As String * 5
Dim Ctg2 As String * 5
Dim B As Byte
Dim W As Word
Dim Dg As Integer
Dim Dsid1(8) As Byte
Dim Dsid2(8) As Byte
Dim Ident(8) As Byte
'Dim DsidN(8) As Byte
Dim Sc(9) As Byte
Dim Aux As Byte
Dim Aux2 As Byte
Dim Ind As Bit
Dim Cuenta As Byte
Dim Cuenta2 As Byte
Dim Busqueda As Byte
Dim Busqueda2 As Byte


'Variables para la recepción y
envio de datos

Const Cmaxchar = 20

Dim Longitud As Word
Dim Lon As Word
Dim Num As Word

Dim Buf As String * Cmaxchar
Dim Trama As String * 3
Dim D As Byte

Dim Indicador1 As Bit
Dim Indicador2 As Bit

'Dim IndicadorN As Bit

Dim Dato As String * 15

'Configuración puertos:
Config Portc.0 = Output
Control Alias Portc.0

'Configuración Timer para buscar
sensor:
Config Timer2 = Timer , Prescale
= 1024
Stop Timer2
Timer2 = 0

'Configuracion senalizacion:
Config Portd.4 = Output
Config Portd.6 = Output
Config Portd.7 = Output
Config Portb.0 = Output

Sensor1 Alias Portd.4
Sensor2 Alias Portd.6
Recibe Alias Portd.7
Envia Alias Portb.0

'Configuración interrupciones:
On Ovf2 Buscarsensor
On Urxc Rec_isr
Enable Ovf2
Enable Urxc

Enable Interrupts

'Configuración sensor de temperat
ura:
Config 1wire = Portb.1

'Inicializaciones:
Control = 0
Uno = 0
Dos = 0
Ind = 0
Sensor1 = 0
Sensor2 = 0
Recibe = 0
Envia = 0
Cuenta = 0
Cuenta2 = 0

'PROGRAMA:

Dsid1(1) = 1wsearchfirst(pinb,1)
```

```
For Aux = 1 To 10
Dsid2(1) = 1wsearchnext(pinb , 1)

   For Aux2 = 1 To 8
   If Dsid2(aux2) <> Dsid1(aux2)
Then
   Aux2 = 8
   Ind = 1
   End If
   Next

If Ind = 1 Then
  Aux = 10
  Ind = 0
Else
  Dsid2(1) = 0
End If
Next Aux


Do
Call Temperatura

If Indicador1 = 1 Then
Longitud = Len(ctg1)
   If Longitud = 4 Then
   Dato = "1ST" + Ctg1
   End If
Call Enviar
Indicador1 = 0
End If
If Indicador2 = 1 Then
Longitud = Len(ctg2)
   If Longitud = 4 Then
   Dato = "2ST" + Ctg2
   End If
Call Enviar
Indicador2 = 0
End If

'Para agregar un nuevo sensor de
temperatura, emplear el siguiente
 código
'cambiando 'N' por el número de s
ensor a agregar

'If IndicadorN = 1 Then
'Dato = "NST" + CtgN
'Call Enviar
'IndicadorN = 0
'End If
Recibe = 0

Loop
End

Buscarsensor:
Incr Busqueda
   If Busqueda = 61 Then
      Stop Timer2
      Busqueda = 0
Dsid1(1) = 1wsearchfirst(pinb,1)
         For Aux2 = 1 To 8
            If Dsid1(aux2) <> Dsi
d2(aux2) Then
            Aux2 = 8
            Ind = 1
            End If
         Next
      If Ind = 1 Then
        Ind = 0
      Else
        Dsid1(1) = 0
      End If

Dsid2(1) = 1wsearchnext(pinb , 1)

         For Aux2 = 1 To 8
           If Dsid2(aux2) <> Dsid
1(aux2) Then
           Aux2 = 8
           Ind = 1
           End If
         Next

      If Ind = 1 Then
        Ind = 0
      Else
        Dsid2(1) = 0
      End If

   End If
Timer2 = 0
Return

Sub Temperatura

   1wreset
   1wwrite &HCC
   1wwrite &H44
   Waitms 2

  Waitus 200

1wverify Dsid1(1)

If Err = 1 Then
   Sensor1 = 0
   Incr Uno
   If Uno >= 3 Then
   Start Timer2
   Ctg1 = "ERR1"
   Uno = 4
   End If
Elseif Err = 0 Then
    Stop Timer2
    Timer2 = 0
    Sensor1 = 1
    Uno = 0
    1wwrite &HBE
```

```
    Sc(1) = 1wread(9)

    If Sc(9) = Crc8(sc(1) , 8) Th
en
       Dg = Decigrades(sc(9))
       Ctg1 = Str(dg)
       Ctg1 = Format(ctg1,"0000")

    End If
End If


1wverify Dsid2(1)

If Err = 1 Then
   Sensor2 = 0
   Incr Dos
   If Dos >= 3 Then
   Start Timer2
   Ctg2 = "ERR1"
   Dos = 4
   End If
Elseif Err = 0 Then
    Stop Timer2
    Timer2 = 0
    Sensor2 = 1
    Dos = 0
    1wwrite &HBE
    Sc(1) = 1wread(9)

    If Sc(9) = Crc8(sc(1) , 8) Th
en
       Dg = Decigrades(sc(9))
       Ctg2 = Str(dg)
       Ctg2 = Format(ctg2,"0000")
    End If
Else

End If

End Sub

Sub Enviar
   Envia = 1
   Control = 1
   Waitms 5
   Print Dato
   Waitms 5
   Control = 0
   Envia = 0
End Sub

Rec_isr:
  D = Udr
  Recibe = 1
If D = 13 Then
    Lon = Len(buf)

     For Num = 1 To Lon

        Trama = Mid(buf ,Num, 3)
        Trama = Left(trama , 3)

       If Trama = "1ST" Then
           Incr Cuenta
           If Cuenta > 3 Then
           Indicador1 = 1
           Cuenta = 4
           End If
       Num = Lon
       End If

       If Trama = "2ST" Then
           Incr Cuenta2
           If Cuenta2 > 3 Then
           Indicador2 = 1
           Cuenta2 = 4
           End If
       Num = Lon
       End If

'Para agregar un nuevo sensor de
temperatura después de hacer lo a
nteriormente
'indicado utilizar este código, c
ambiando la N por el número de se
nsor

        'If trama= "NST" Then
        'IndicadorN = 1
        'Num = Lon
        'End If
     Next

        Buf = Buf + Chr(0)
        Buf = ""

        Num = 0

        Goto Salir
     Else
       Buf = Buf + Chr(d)
     End If
Salir:
Return

Function Decigrades(byval Sc(9) A
s Byte)
Decigrades = Makeint(sc(1),Sc(2))
   Decigrades = Decigrades * 10
   Decigrades = Decigrades / 2
End Function
```

NODO TEMPERATURA TALLER:

```
$regfile = "m48def.dat"
$crystal = 8000000
$baud = 9600

'Declaración funciones:
Declare Sub Temperatura
Declare Sub Enviar
Declare Function Decigrades(byval
 Sc(9) As Byte) As Integer

'Declaración variables
temperatura:
Dim Uno As Byte
Dim Dos As Byte
Dim Ctg1 As String * 5
Dim Ctg2 As String * 5
Dim B As Byte
Dim W As Word
Dim Dg As Integer
Dim Dsid1(8) As Byte
Dim Dsid2(8) As Byte
Dim Ident(8) As Byte
'Dim DsidN(8) As Byte
Dim Sc(9) As Byte
Dim Aux As Byte
Dim Aux2 As Byte
Dim Ind As Bit
Dim Cuenta As Byte
Dim Cuenta2 As Byte
Dim Busqueda As Byte
Dim Busqueda2 As Byte


'Variables para la recepción y
envio de datos

Const Cmaxchar = 20

Dim Longitud As Word
Dim Lon As Word
Dim Num As Word

Dim Buf As String * Cmaxchar
Dim Trama As String * 3
Dim D As Byte

Dim Indicador1 As Bit
Dim Indicador2 As Bit

'Dim IndicadorN As Bit

Dim Dato As String * 15

'Configuración puertos:
Config Portc.0 = Output
Control Alias Portc.0

'Configuración Timer para buscar
sensor:
Config Timer2 = Timer , Prescale
= 1024
Stop Timer2
Timer2 = 0

'Configuracion senalizacion:
Config Portd.4 = Output
Config Portd.6 = Output
Config Portd.7 = Output
Config Portb.0 = Output

Sensor1 Alias Portd.4
Sensor2 Alias Portd.6
Recibe Alias Portd.7
Envia Alias Portb.0

'Configuración interrupciones:
On Ovf2 Buscarsensor
On Urxc Rec_isr
Enable Ovf2
Enable Urxc

Enable Interrupts
'Configuración sensor de
temperatura:
Config 1wire = Portb.1

'Inicializaciones:
Control = 0
Uno = 0
Dos = 0
Ind = 0
Sensor1 = 0
Sensor2 = 0
Recibe = 0
Envia = 0
Cuenta = 0
Cuenta2 = 0

'PROGRAMA:

Dsid1(1) = 1wsearchfirst(pinb,1)

For Aux = 1 To 10
Dsid2(1) = 1wsearchnext(pinb , 1)

   For Aux2 = 1 To 8
   If Dsid2(aux2) <> Dsid1(aux2)
Then
   Aux2 = 8
   Ind = 1
   End If
   Next

If Ind = 1 Then
```

```
  Aux = 10
  Ind = 0
Else
  Dsid2(1) = 0
End If
Next Aux


Do
Call Temperatura

If Indicador1 = 1 Then
Longitud = Len(ctg1)
   If Longitud = 4 Then
   Dato = "3ST" + Ctg1
   End If
Call Enviar
Indicador1 = 0
End If

If Indicador2 = 1 Then
Longitud = Len(ctg2)
   If Longitud = 4 Then
   Dato = "4ST" + Ctg2
   End If
Call Enviar
Indicador2 = 0
End If

'//Para agregar un nuevo sensor d
e temperatura, emplear el siguien
te código
'//cambiando 'N' por el número de
 sensor a agregar

'If IndicadorN = 1 Then
'Dato = "NST" + CtgN
'Call Enviar
'IndicadorN = 0
'End If
Recibe = 0

Loop
End

Buscarsensor:
Incr Busqueda
   If Busqueda = 61 Then
      Stop Timer2
      Busqueda = 0

Dsid1(1) = 1wsearchfirst(pinb,1)
         For Aux2 = 1 To 8
            If Dsid1(aux2) <> Dsi
d2(aux2) Then
            Aux2 = 8
            Ind = 1
            End If
         Next
      If Ind = 1 Then
         Ind = 0
      Else
         Dsid1(1) = 0
      End If

Dsid2(1) = 1wsearchnext(pinb , 1)

         For Aux2 = 1 To 8
           If Dsid2(aux2) <> Dsid
1(aux2) Then
            Aux2 = 8
            Ind = 1
            End If
         Next

      If Ind = 1 Then
         Ind = 0
      Else
         Dsid2(1) = 0
      End If

   End If
Timer2 = 0
Return
Sub Temperatura

   1wreset
   1wwrite &HCC
   1wwrite &H44
   Waitms 2

  Waitus 200

1wverify Dsid1(1)

If Err = 1 Then
   Sensor1 = 0
   Incr Uno
   If Uno >= 3 Then
   Start Timer2
   Ctg1 = "ERR1"
   Uno = 4
   End If
Elseif Err = 0 Then
    Stop Timer2
    Timer2 = 0
    Sensor1 = 1
    Uno = 0
    1wwrite &HBE
    Sc(1) = 1wread(9)


    If Sc(9) = Crc8(sc(1) , 8)
    Then
       Dg = Decigrades(sc(9))
       Ctg1 = Str(dg)
       Ctg1 = Format(ctg1,"0000")

    End If
End If
```

```
1wverify Dsid2(1)

If Err = 1 Then
   Sensor2 = 0
   Incr Dos
   If Dos >= 3 Then
   Start Timer2
   Ctg2 = "ERR1"
   Dos = 4
   End If
Elseif Err = 0 Then
    Stop Timer2
    Timer2 = 0
    Sensor2 = 1
    Dos = 0
    1wwrite &HBE
    Sc(1) = 1wread(9)

    If Sc(9) = Crc8(sc(1) , 8) Th
en
       Dg = Decigrades(sc(9))
       Ctg2 = Str(dg)
       Ctg2 = Format(ctg2,"0000")
    End If
Else

End If

End Sub

Sub Enviar
   Envia = 1
   Control = 1
   Waitms 5
   Print Dato
   Waitms 5
   Control = 0
   Envia = 0
End Sub

Rec_isr:
  D = Udr
  Recibe = 1
If D = 13 Then

     Lon = Len(buf)

      For Num = 1 To Lon

         Trama = Mid(buf ,Num, 3)
         Trama = Left(trama , 3)
         If Trama = "3ST" Then
             Incr Cuenta
             If Cuenta > 3 Then
             Indicador1 = 1
             Cuenta = 4
             End If
         Num = Lon
         End If

         If Trama = "4ST" Then
             Incr Cuenta2
             If Cuenta2 > 3 Then
             Indicador2 = 1
             Cuenta2 = 4
             End If
         Num = Lon
         End If


'Para agregar un nuevo sensor de
temperatura después de hacer lo a
nteriormente
'indicado utilizar este código, c
ambiando la N por el número de se
nsor
         'If trama= "NST" Then
         'IndicadorN = 1
         'Num = Lon
         'End If
      Next

         Buf = Buf + Chr(0)
         Buf = ""

         Num = 0

         Goto Salir
      Else
        Buf = Buf + Chr(d)
      End If
Salir:
Return

Function Decigrades(byval Sc(9) A
s Byte)
Decigrades = Makeint(sc(1),Sc(2))
   Decigrades = Decigrades * 10
   Decigrades = Decigrades / 2
End Function
```

NODO DE PRESENCIA DE AGUA DEPARTAMENTO TÉCNICO:

```
$regfile = "m48def.dat"
$crystal = 8000000
$baud = 9600
```

```
'Dimensionamiento variables:
Const Cmaxchar = 20
Const Porcientohum = 100

Dim Lon As Word
Dim Num As Word

Dim Buf As String * Cmaxchar
Dim D As Byte
Dim S As String * 10
Dim S1 As String * 10
Dim Indicador As Bit
Dim Indicador2 As Bit
Dim Dato As String * 15
Dim Humedad As String * 15
Dim Trama As String * 10

Dim Base As Byte
Dim Frecuencia As Word
Dim W As Word

'Configuración puertos:
Config Portc.0 = Output
Control Alias Portc.0

'Configuración señalización:
Config Portd.6 = Output
Config Portd.7 = Output
Config Portb.0 = Output

Inundacion Alias Portd.6
Recibe Alias Portd.7
Envia Alias Portb.0

'Declaración subrutinas:
Declare Sub Enviar

'Configuración timers:
Config Timer2 = Timer , Prescale = 1024
Stop Timer2
Timer2 = 0

Config Timer1 = Counter , Edge = Falling , Capture Edge = Falling , Presc
ale = 1
Stop Counter1
Counter1 = 0

'Puertos para sensores: magnético, humo e inundación.
Config Portd.4 = Input

I1 Alias Pind.4


'Inicialización variables:
Control = 0
Inundacion = 0
Recibe = 0
Envia = 0
Base = 0
Humedad = "INIC"
```

```
'Configuración interrupciones:

On Ovf2 Basetiempo
On Urxc Rec_isr
Enable Urxc
Enable Ovf2
Enable Interrupts

Start Timer2
Start Counter1

Do

   If Indicador = 1 Then
       Gosub Convertir
       Dato = "1SU" + Humedad
       Call Enviar
       Envia = 0
       Indicador = 0

   End If

   If Indicador2 = 1 Then

      If I1 = 0 Then
      Dato = "1SI0001"
      Call Enviar
      Envia = 0
      Inundacion = 1
      Elseif I1 = 1 Then
      Dato = "1SI0000"
      Call Enviar
      Envia = 0
      Inundacion = 0
      End If
      Indicador2 = 0
   End If

Recibe = 0
Loop
End

Sub Enviar
   Envia = 1
   Control = 1
   Waitms 2
   Print Dato
   Waitms 2
   Control = 0
End Sub


Basetiempo:
Timer2 = 0
Incr Base
If Base = 31 Then
Frecuencia = Counter1
Base = 0
Counter1 = 0
```

```
End If
Return


Rec_isr:
  D = Udr
   Recibe = 1
If D = 13 Then
     Lon = Len(buf)

      For Num = 1 To Lon

        Trama = Mid(buf , Num , 3)
        Trama = Left(trama , 3)

        If Trama = "1SI" Then
        Indicador2 = 1
        Num = Lon
        End If

        If Trama = "1SU" Then
        Indicador = 1
        Num = Lon
        End If


      Next
        Buf = Buf + Chr(0)
        Buf = ""

        Num = 0

        Goto Salir
     Else
       Buf = Buf + Chr(d)
     End If
Salir:
Return

Convertir:
If Frecuencia > 7351 Then
   Humedad = "0000"
Elseif Frecuencia < 7251 And Frecuencia > 7224 Then
   Humedad = "0005"
Elseif Frecuencia = 7224 Then
   Humedad = "0010"

Elseif Frecuencia < 7224 And Frecuencia > 7100 Then
   Humedad = "0015"
Elseif Frecuencia = 7100 Then
   Humedad = "0020"

   Elseif Frecuencia < 7100 And Frecuencia > 6976 Then
   Humedad = "0025"
Elseif Frecuencia = 6976 Then
   Humedad = "0030"

   Elseif Frecuencia < 6976 And Frecuencia > 6853 Then
   Humedad = "0035"
Elseif Frecuencia = 6853 Then
```

```
   Humedad = "0040"

   Elseif Frecuencia < 6853 And Frecuencia > 6728 Then
   Humedad = "0045"
Elseif Frecuencia = 6728 Then
   Humedad = "0050"

   Elseif Frecuencia < 6728 And Frecuencia > 6600 Then
   Humedad = "0055"
Elseif Frecuencia = 6600 Then
   Humedad = "0060"

   Elseif Frecuencia < 6600 And Frecuencia > 6468 Then
   Humedad = "0065"
Elseif Frecuencia = 6468 Then
   Humedad = "0070"

   Elseif Frecuencia < 6468 And Frecuencia > 6330 Then
   Humedad = "0075"
Elseif Frecuencia = 6330 Then
   Humedad = "0080"

   Elseif Frecuencia < 6330 And Frecuencia > 6186 Then
   Humedad = "0085"
Elseif Frecuencia = 6186 Then
   Humedad = "0090"

   Elseif Frecuencia < 6186 And Frecuencia > 6033 Then
   Humedad = "0095"
Elseif Frecuencia = 6033 Then
   Humedad = "0100"
Else
   Humedad = "fuera"
End If
Return
```

NODO DE PRESENCIA DE AGUA TALLER:

```
$regfile = "m48def.dat"
$crystal = 8000000
$baud = 9600

'Dimensionamiento variables:
Const Cmaxchar = 20
Const Porcientohum = 100

Dim Lon As Word
Dim Num As Word

Dim Buf As String * Cmaxchar
Dim D As Byte
Dim S As String * 10
Dim S1 As String * 10
Dim Indicador As Bit
Dim Indicador2 As Bit
Dim Dato As String * 15
Dim Humedad As String * 15
Dim Trama As String * 10
```

```
Dim Base As Byte
Dim Frecuencia As Word
Dim W As Word

'Configuración puertos:
Config Portc.0 = Output
Control Alias Portc.0

'Configuración señalización:
Config Portd.6 = Output
Config Portd.7 = Output
Config Portb.0 = Output

Inundacion Alias Portd.6
Recibe Alias Portd.7
Envia Alias Portb.0

'Declaración subrutinas:
Declare Sub Enviar

'Configuración timers:
Config Timer2 = Timer , Prescale = 1024
Stop Timer2
Timer2 = 0

Config Timer1 = Counter , Edge = Falling , Capture Edge = Falling , Presc
ale = 1
Stop Counter1
Counter1 = 0

'Puertos para sensores: magnético, humo e inundación.
Config Portd.4 = Input

I1 Alias Pind.4


'Inicialización variables:
Control = 0
Inundacion = 0
Recibe = 0
Envia = 0
Base = 0
Humedad = "INIC"

'Configuración interrupciones:

On Ovf2 Basetiempo
On Urxc Rec_isr
Enable Urxc
Enable Ovf2
Enable Interrupts

Start Timer2
Start Counter1

Do

If Indicador = 1 Then
Gosub Convertir
```

```
Dato = "2SU" + Humedad
Call Enviar
Envia = 0
Indicador = 0

End If

If Indicador2 = 1 Then

   If I1 = 0 Then
   Dato = "2SI0001"
   Call Enviar
   Envia = 0
   Inundacion = 1
   Elseif I1 = 1 Then
   Dato = "2SI0000"
   Call Enviar
   Envia = 0
   Inundacion = 0
   End If
   Indicador2 = 0
End If

Recibe = 0
Loop
End

Sub Enviar
   Envia = 1
   Control = 1
   Waitms 2
   Print Dato
   Waitms 2
   Control = 0
End Sub


Basetiempo:
Timer2 = 0
Incr Base
If Base = 31 Then
Frecuencia = Counter1
Base = 0
Counter1 = 0
End If
Return


Rec_isr:
  D = Udr
   Recibe = 1
If D = 13 Then
     Lon = Len(buf)
      For Num = 1 To Lon

        Trama = Mid(buf , Num , 3)
        Trama = Left(trama , 3)

        If Trama = "2SI" Then
        Indicador2 = 1
```

```
        Num = Lon
        End If

        If Trama = "2SU" Then
        Indicador = 1
        Num = Lon
        End If


      Next
        Buf = Buf + Chr(0)
        Buf = ""

        Num = 0

        Goto Salir
     Else
       Buf = Buf + Chr(d)
     End If
Salir:
Return

Convertir:
If Frecuencia > 7351 Then
   Humedad = "0000"
Elseif Frecuencia < 7251 And Frecuencia > 7224 Then
   Humedad = "0005"
Elseif Frecuencia = 7224 Then
   Humedad = "0010"

Elseif Frecuencia < 7224 And Frecuencia > 7100 Then
   Humedad = "0015"
Elseif Frecuencia = 7100 Then
   Humedad = "0020"

   Elseif Frecuencia < 7100 And Frecuencia > 6976 Then
   Humedad = "0025"
Elseif Frecuencia = 6976 Then
   Humedad = "0030"

   Elseif Frecuencia < 6976 And Frecuencia > 6853 Then
   Humedad = "0035"
Elseif Frecuencia = 6853 Then
   Humedad = "0040"

   Elseif Frecuencia < 6853 And Frecuencia > 6728 Then
   Humedad = "0045"
Elseif Frecuencia = 6728 Then
   Humedad = "0050"

   Elseif Frecuencia < 6728 And Frecuencia > 6600 Then
   Humedad = "0055"
Elseif Frecuencia = 6600 Then
   Humedad = "0060"

   Elseif Frecuencia < 6600 And Frecuencia > 6468 Then
   Humedad = "0065"
Elseif Frecuencia = 6468 Then
   Humedad = "0070"
```

```
   Elseif Frecuencia < 6468 And Frecuencia > 6330 Then
   Humedad = "0075"
Elseif Frecuencia = 6330 Then
   Humedad = "0080"

   Elseif Frecuencia < 6330 And Frecuencia > 6186 Then
   Humedad = "0085"
Elseif Frecuencia = 6186 Then
   Humedad = "0090"

   Elseif Frecuencia < 6186 And Frecuencia > 6033 Then
   Humedad = "0095"
Elseif Frecuencia = 6033 Then
   Humedad = "0100"
Else
   Humedad = "fuera"
End If
Return
```

NODO DE CORRIENTE:

```
$regfile = "m48def.dat"
$crystal = 8000000
$baud = 9600

'Dimensionamiento variables:
Const Cmaxchar = 20
Const Corrmedida = 20
Const Maxcurrent = 20

Dim Lon As Word
Dim Num As Word
Dim Longitud As Byte

Dim Buf As String * Cmaxchar
Dim D As Byte
Dim S As String * 10
Dim S1 As String * 10
Dim Indicador As Bit
Dim Dato As String * 15
Dim Corriente As String * 15
Dim Trama As String * 7
Dim Decimal As Single

Dim Aux As Byte
Dim V As Long
Dim Suma As Long
Dim Rms As Long
Dim Current As String * 4
Dim Promedio As Long
Dim Cont As Byte


'Configuración puertos:
Config Portc.0 = Output
Control Alias Portc.0
```

```
'Configuración señalización:
Config Portd.4 = Output
Config Portd.7 = Output
Config Portb.0 = Output

Poder Alias Portd.4
Recibe Alias Portd.7
Envia Alias Portb.0

'Declaración de subrutinas

Declare Sub Enviar
Declare Sub Muestreo
Declare Sub Rmscalc

'Habilitación de interrupciones:
On Urxc Rec_isr
Enable Urxc
Enable Interrupts

'Configuración conversor análogo digital:
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc

'Inicialización variables:

Control = 0
Recibe = 0
Envia = 0
Suma = 0
V = 0
Aux = 0
Cont = 0
Rms = 0
Current = "0000"
Poder = 0

'PROGRAMA:
Do
   Poder = 1
   Call Muestreo
   Envia = 0
   Recibe = 0
   Poder = 0
Loop
End

'Subrutinas
Sub Enviar
   Envia = 1
   Control = 1
   Waitms 5
   Print Dato
   Waitms 5
   Control = 0

End Sub

Rec_isr:
  D = Udr
```

```
  Recibe = 1
If D = 13 Then
     Lon = Len(buf)


      For Num = 1 To Lon

        Trama = Mid(buf , Num , 3)
        Trama = Left(trama , 3)

        If Trama = "1SC" Then
        Dato = "1SC" + Current
        Call Enviar
        Num = Lon
        End If

      Next
        Buf = Buf + Chr(0)
        Buf = ""

        Num = 0

        Goto Salir
     Else
       Buf = Buf + Chr(d)
     End If
Salir:
Return


Sub Muestreo
 V = Getadc(1)
 Incr Aux
 Suma = Suma + V

 If Aux > 60 Then
 Call Rmscalc
 Aux = 0
 End If

End Sub

Sub Rmscalc
Incr Cont
'cálculo rms
Decimal = Suma / Aux
Decimal = Decimal * 100
Rms = Decimal / 1023
Rms = Rms * Maxcurrent
Promedio = Promedio + Rms

If Cont > 50 Then

Promedio = Promedio / Cont

Corriente = Str(promedio)

Longitud = Len(corriente)

If Longitud <= 4 Then
```

```
Current = Left(current , 4)
Current = Format(corriente , "0000")

End If

Promedio = 0
Cont = 0

End If

Aux = 0
Suma = 0

End Sub
```

CÓDIGO FUENTE DE LA APLICACIÓN DE LA INTERFAZ ESTACIONARIA:

FORM SENSORES:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using MySql.Data.MySqlClient;
using System.Net;
using System.Net.Mail;
using System.IO;


//clave hilo:ev99S6mrBAsr7rL9
//clave smsmail:YwzrdqMfUbXvvwRf
namespace Control_Domótico
{
    public partial class Sensores : Form
    {
        public Thread HDaq;
        private MySqlConnection connection;
        private MySqlConnection connectionthread;
        private MySqlConnection connectionsmsmail;

        const int maxsmsdest = 7; //Numero de destinatarios sms permitidos = 6
        const int maxnumberdestmail = 7; //Numero maximo de destinatarios mail=6

        int sendEsclavos = 0;
        string enter = "" + (char)13;
        string valueT1 = "0";
        string valueT2 = "0";
        string valueT3 = "0";
        string valueT4 = "0";
        string valueC1 = "0";
        string valueC2 = "";
        string valueI1 = "";
        string valueI2 = "";
        string valueM1 = "";
```

```
string valueH1 = "";
string valueH2 = "";
string sendBluetooth = "";
string getDaq = "";
string getBluetooth = "";
string auxBluetooth = "";
string receptvalue = "";
string commBluet = "";
string commDAQ = "";
string daqAnswer = "";


bool permiso = true;
bool alarmT1 = false;
bool alarmT2 = false;
bool alarmT3 = false;
bool alarmT4 = false;
bool alarmMaxC1 = false;
bool alarmMaxC2 = false;
bool alarmMinC1 = false;
bool alarmMinC2 = false;
bool alarmI1 = false;
bool alarmI2 = false;
bool alarmM1 = false;
bool alarmH1 = false;
bool alarmH2 = false;
bool alarmErr1T1 = false;
bool alarmErr1T2 = false;
bool alarmErr1T3 = false;
bool alarmErr1T4 = false;
bool alarmErr2T1 = false;
bool alarmErr2T2 = false;
bool alarmErr2T3 = false;
bool alarmErr2T4 = false;
bool alarmErr2C1 = false;
bool alarmErr2C2 = false;
bool alarmErr2I1 = false;
bool alarmErr2I2 = false;
bool cuidadoTaller = false;
bool cuidadoDptoTecnico = false;
bool permisoT1 = true;
bool permisoT2 = true;
bool permisoT3 = true;
bool permisoT4 = true;
bool permisoI1 = true;
bool permisoI2 = true;
bool permisoM1 = false;
bool permisoH1 = true;
bool permisoH2 = true;
bool permisoC1 = true;
bool enableAlarmMagnetic = false;
bool enableTimerMagnetic = true;
bool flagEnableTimerMagnetic = false;
bool banderasalida = false;
bool enableAlarmMessage = true;
bool checkDAQ = true;
bool daqDisconnected = false;
bool nodata = false;
bool enableMensajeNoData = true;
bool firstValues = true;

double alarmvalueT1;
double alarmvalueT2;
double alarmvalueT3;
double alarmvalueT4;
double alarmvalueC1;
double conversionvalueT1;
```

```
        double conversionvalueT2;
        double conversionvalueT3;
        double conversionvalueT4;
        double valueTaller;
        double valueDptoTecnico;
        double auxBluet;
        double aux;
        double TempMaxTaller;
        double TempMaxDptoTecnico;

        int porcentajeBarraProgresoTemp;
        int contadorDaqDesconectada;
        int valueSensor;
        int contadorAlarmaC1;


        public Sensores()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            //Configuracion Puertos COM:
            dterminacionPuertos b = new dterminacionPuertos();
            b.ShowDialog();

                if (b.comDaq != "")
                {
                    srlDAQ.PortName = b.comDaq;
                }

                if (b.comBluetooth != "")
                {
                    srlBluetooth.PortName = b.comBluetooth;
                }


                if (b.comSMS != "")
                {
                    srlSMS.PortName = b.comSMS;
                }

            b.Close();

            this.Location = Screen.PrimaryScreen.WorkingArea.Location;
            iniciarbasedatos();
            iniciarbasedatosSmsMail();

            //iniciar base de datos hilo:
            iniciarbasedatosthread();

            try
            {
                srlDAQ.Open();
                tmrCheckDAQ.Enabled = true;

            }
            catch
            {
                MessageBox.Show("DAQ no encontrada.", "DAQ ERROR",
MessageBoxButtons.OK,MessageBoxIcon.Error);
                pbM1NoData.Visible = true;
                pbC1NoData.Visible = true;
                pbTallerTNoData.Visible = true;
                pbDepTecTNoData.Visible = true;
                pbTallerINoData.Visible = true;
```

```
                pbDepTecINoData.Visible = true;
                pbTallerHNoData.Visible = true;
                pbDepTecHNoData.Visible = true;
                pbSensorTErr1.Visible = true;
                pbSensorTErr2.Visible = true;
                pbSensorTErr3.Visible = true;
                pbSensorTErr4.Visible = true;
                pbSensorCurrentErr.Visible = true;
                pbSensorHumo1Err.Visible = true;
                pbSensorHumo2Err.Visible = true;
                pbSensorIErr.Visible = true;
                pbSensorIErr2.Visible = true;
                pbSensorMErr.Visible = true;
                nodata = true;

            }

            try
            {
                srlSMS.Open();
            }
            catch
            {
                MessageBox.Show("Dispositivo SMS no encontrado.", "SMS ERROR",
MessageBoxButtons.OK);
            }

            try
            {
                HDaq = new Thread(new ThreadStart(lecturaDaq));
                HDaq.Start();
            }
            catch
            {
                MessageBox.Show("No se pudo iniciar el hilo de sensores.",
"THREAD ERROR", MessageBoxButtons.OK);
            }
            tmrSendMaster.Enabled = true;

            try
            {
                srlBluetooth.Open();

            }
            catch
            {
                MessageBox.Show("No se puede habilitar el servicio Bluetooth.",
"BLUETOOTH ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void iniciarbasedatos()
        {
            string server = "localhost";
            string database = "controldomotico";
            string uid = "dsupervisor";
            string password = "2hF3JNzeS47dxK4R";
            string connectionString = "";
            connectionString =
"SERVER="+server+";"+"DATABASE="+database+";"+"UID="+uid+";"+"PASSWORD="+password
;
            connection = new MySqlConnection(connectionString);
regreso:
            try
            {
                connection.Open();
            }
```

```
            catch
            {
                DialogResult errorbase = MessageBox.Show("No se pudo abrir la
base de datos", "Error Base de Datos", MessageBoxButtons.RetryCancel);
                if (errorbase == DialogResult.Retry)
                {
                    goto regreso;
                }
            }
        }

        private void iniciarbasedatosSmsMail()
        {
            string server = "localhost";
            string database = "controldomotico";
            string uid = "dsmsmail";
            string password = "YwzrdqMfUbXvvwRf";
            string connectionString = "";
            connectionString = "SERVER=" + server + ";" + "DATABASE=" + database
+ ";" + "UID=" + uid + ";" + "PASSWORD=" + password;
            connectionsmsmail = new MySqlConnection(connectionString);
        regreso:
            try
            {
                connectionsmsmail.Open();
            }
            catch
            {
                DialogResult errorbase = MessageBox.Show("No se pudo abrir la
base de datos para envio Sms y Mail", "Error Base de Datos",
MessageBoxButtons.RetryCancel);
                if (errorbase == DialogResult.Retry)
                {
                    goto regreso;
                }
            }
        }

        private void iniciarbasedatosthread()
        {
            string server = "localhost";
            string database = "controldomotico";
            string uid = "dthread";
            string password = "ev99S6mrBAsr7rL9";
            string connectionString = "";
            connectionString = "SERVER=" + server + ";" + "DATABASE=" + database
+ ";" + "UID=" + uid + ";" + "PASSWORD=" + password;
            connectionthread = new MySqlConnection(connectionString);

        regreso:
            try
            {
                connectionthread.Open();
            }
            catch
            {
                DialogResult errorbase = MessageBox.Show("No se pudo abrir la
base de datos del hilo", "Error Base de Datos", MessageBoxButtons.RetryCancel);
                if (errorbase == DialogResult.Retry)
                {
                    goto regreso;
                }
            }
        }
        private void btnAdmin_Click(object sender, EventArgs e)
        {
            User a =new User();
```

```
            a.ShowDialog();

        }

        public void lecturaDaq()
        {
            string comando = "";
            int num = 0;
            int lazo = 0;
            string codigo = "";
            int limite = 0;
            bool convertir=false;
            int contadorERR1T1=0;
            int contadorERR2T1=0;
            int contadorERR1T2 = 0;
            int contadorERR2T2 = 0;
            int contadorERR1T3 = 0;
            int contadorERR2T3 = 0;
            int contadorERR1T4 = 0;
            int contadorERR2T4 = 0;
            int contadorERR2C1 = 0;
            int contadorERR2C2 = 0;
            int contadorERR2I1 = 0;
            int contadorERR2I2 = 0;
            const int testerror = 10;
            string sensorValueTaller = "";
            string sensorValueTecnico="";
            string auxSensorValue = "";

          while(lazo==0)
          {
              if (permiso == true & srlDAQ.IsOpen==true)
              {
                  //VALIDACION DE DATOS:
                  if (comando == "1SV0001" + enter)
                  {
                      if (getDaq != "")
                      {
                          daqAnswer = getDaq.Substring(0, 2);
                      }

                      if (daqAnswer != "OK" )
                      {
                          contadorDaqDesconectada++;
                        if (contadorDaqDesconectada==3&enableMensajeNoData==true)
                          {
                              MessageBox.Show("Tarjeta de adquisición de datos
desconectada", "ALERTA", MessageBoxButtons.OK, MessageBoxIcon.Error);
                              daqDisconnected = true;
                              nodata = true;
                              enableMensajeNoData = false;
                              contadorDaqDesconectada = 0;
                          }
                      }
                      else if (daqAnswer=="OK")
                      {
                          daqDisconnected = false;
                          nodata = false;
                          enableMensajeNoData = true;
                          contadorDaqDesconectada = 0;

                      }
                  }

                  if (getDaq.Length == 9)
                  {
                      codigo = getDaq.Substring(0, 3);
```

```
                    if (codigo == "1ST")
                    {
                        receptvalue=getDaq.Substring(3, 4);
                        if (receptvalue !="")
                        {
                              try
                              {
                                valueSensor=Convert.ToInt16(receptvalue);
                                if ((valueSensor>Convert.ToInt16(valueT1)/2 &
valueSensor<Convert.ToInt16(valueT1)*2) || firstValues==true)
                                {
                                    valueT1 = receptvalue;
                                }

                                contadorERR1T1 = 0;
                                contadorERR2T1 = 0;
                              }
                              catch
                              {
                                  if (receptvalue=="ERR1")
                                  {
                                      contadorERR1T1++ ;
                                      if (contadorERR1T1>=testerror-1)
                                      {
                                          valueT1 = receptvalue;
                                      }
                                  }
                                  if (receptvalue == "ERR2")
                                  {
                                      contadorERR2T1++;
                                     if (contadorERR2T1 >= testerror-1)
                                      {
                                          valueT1 = receptvalue;
                                      }
                                  }
                              }
                          }
                      }
                    else
                    {
                        if (codigo == "2ST")
                        {
                            receptvalue = getDaq.Substring(3, 4);
                            if (receptvalue != "")
                            {
                                    try
                                    {

                                    valueSensor=Convert.ToInt16(receptvalue);

                                        if ((valueSensor >
                                        Convert.ToInt16(valueT2) / 2 &
                                        valueSensor < Convert.ToInt16(valueT2)
                                        * 2) || firstValues == true)
                                        {
                                            valueT2 = receptvalue;
                                        }
                                        contadorERR1T2 = 0;
                                        contadorERR2T2 = 0;
                                    }
                                    catch
                                    {
                                        if (receptvalue == "ERR1")
                                        {
                                            contadorERR1T2++;
                                        if (contadorERR1T2 >=  testerror-1)
                                            {
```

```
                                valueT2 = receptvalue;
                            }
                        }
                        if (receptvalue == "ERR2")
                        {
                            contadorERR2T2++;
                            if (contadorERR2T2 >= testerror-1)
                            {
                                valueT2 = receptvalue;
                            }
                        }
                    }
                }
            }
        else
        {
            if (codigo == "3ST")
            {
                receptvalue = getDaq.Substring(3, 4);
                if (receptvalue != "")
                {
                        try
                        {

                     valueSensor=Convert.ToInt16(receptvalue);
                            if ((valueSensor >
                            Convert.ToInt16(valueT3) / 2 &
                            valueSensor <
                            Convert.ToInt16(valueT3) * 2) ||
                            firstValues == true)
                            {
                                valueT3 = receptvalue;
                            }

                            contadorERR1T3 = 0;
                            contadorERR2T3 = 0;
                        }
                        catch
                        {
                            if (receptvalue == "ERR1")
                            {
                                contadorERR1T3++;
                                if (contadorERR1T3 >=
                                testerror-1)
                                {
                                    valueT3 = receptvalue;
                                }
                            }
                            if (receptvalue == "ERR2")
                            {
                                contadorERR2T3++;

                                if (contadorERR2T3 >=
                                testerror-1)
                                {
                                    valueT3 = receptvalue;
                                }
                            }
                        }
                    }
            }
            else
            {
                if (codigo == "4ST")
                {
                    receptvalue = getDaq.Substring(3, 4);
                    if (receptvalue != "")
```

```
    {
            try
            {

     valueSensor=Convert.ToInt16(receptvalue);

                if ((valueSensor >
                Convert.ToInt16(valueT4) / 2 &
                valueSensor <
                Convert.ToInt16(valueT4) * 2)
                || firstValues == true)
                {
                    valueT4 = receptvalue;
                }

                contadorERR1T4 = 0;
                contadorERR2T4 = 0;
            }
            catch
            {
                if (receptvalue == "ERR1")
                {
                    contadorERR1T4++;

                    if (contadorERR1T4 >=
                    testerror-1)
                    {
                        valueT4 = receptvalue;
                    }
                }
                if (receptvalue == "ERR2")
                {
                    contadorERR2T4++;

                    if (contadorERR2T4 >=
                    testerror-1)
                    {
                        valueT4 = receptvalue;
                    }
                }
            }
    }
}
else
{
    if (codigo == "1SC")
    {
        receptvalue = getDaq.Substring(3, 4);
        if (receptvalue != "")
        {
                try
                {

     valueSensor=Convert.ToInt16(receptvalue);

                    if ((valueSensor >
                    Convert.ToInt16(valueC1) /
                    2 & valueSensor <
                    Convert.ToInt16(valueC1) *
                    2) || firstValues == true
                    || contadorAlarmaC1==3)
                    {
                        valueC1 = receptvalue;
                        contadorAlarmaC1=0;
                    }
                    else
                    {
```

```
                        contadorAlarmaC1++;
                    }
                    contadorERR2C1 = 0;
                }
                catch
                {

                    if (receptvalue == "ERR2")
                    {
                        contadorERR2C1++;
                        if (contadorERR2C1 >=
                        testerror-1)
                        {
                            valueC1 =
                        receptvalue;
                        }
                    }
                }
            }
        }
        else
        {
            if (codigo == "2SC")
            {
               receptvalue = getDaq.Substring(3,4);
                if (receptvalue != "")
                {

                        try
                        {

                     Convert.ToInt16(receptvalue);
                            valueC2 = receptvalue;
                            contadorERR2C2 = 0;
                        }
                        catch
                        {
                            if (receptvalue ==
                            "ERR2")
                            {
                                contadorERR2C2++;
                            if (contadorERR2C2 >=
                            testerror-1)
                                {
                             valueC2 = receptvalue;
                                }
                            }
                        }
                }
            }
            else
            {
                if (codigo == "1SI")
                {
                 receptvalue=getDaq.Substring(3,4);
                    if (receptvalue != "")
                    {
                            try
                            {

                     Convert.ToInt16(receptvalue);
                      valueI1 = receptvalue;
                      contadorERR2I1 = 0;
                            }
                            catch
                            {
                        if (receptvalue == "ERR2")
```

```
                {

                contadorERR2I1++;
                 if (contadorERR2I1 >=
                 testerror-1)
                 {
                  valueI1 = receptvalue;
                  }
                }
              }
            }
        }
        else
        {
            if (codigo == "2SI")
            {
                receptvalue =
                getDaq.Substring(3, 4);
                if (receptvalue != "")
                {
                  try
                  {
                   Convert.ToInt16(receptva
                   lue);
                    valueI2 = receptvalue;
                    contadorERR2I2 = 0;
                    }
                    catch
                    {
                    if (receptvalue=="ERR2")
                    {
                     contadorERR2I2++;

                      if (contadorERR2I2 >=
                      testerror-1)
                      {
                      valueI2 = receptvalue;
                      }
                     }
                  }
                }
            }
            else
            {
                if (codigo == "1SM")
                {
                    receptvalue =
                    getDaq.Substring(3, 4);
                    if (receptvalue != "")
                    {
                      try
                      {
                       Convert.ToInt16(rece
                       ptvalue);
                      valueM1 = receptvalue;

                        }
                        catch
                        {
                        }
                    }
                }
                else
                {
                    if (codigo == "1SH")
                    {
receptvalue = getDaq.Substring(3, 4);
```

```
                                                    if (receptvalue != "")
                                                        {
                                                        try
                                                        {
                                                Convert.ToInt16(receptvalue);
                                                 valueH1 = receptvalue;

                                                        }
                                                        catch
                                                        {

                                                        }
                                                        }
                                                    }
                                                    else
                                                    {
                                                      if (codigo == "2SH")
                                                        {
                                            receptvalue = getDaq.Substring(3, 4);
                                                       if (receptvalue != "")
                                                       {
                                                       try
                                                       {

                                                Convert.ToInt16(receptvalue);
                                                valueH2 = receptvalue;

                                                        }
                                                        catch
                                                                {

                                                                }
                                                               }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }

if (valueT1!="0" & valueT2!="0" & valueT3!="0" & valueT4!="0" & valueC1!="0")
                        {
                            firstValues = false;
                        }
                    }


                    //LECTURA SENSORES:
                    sendEsclavos++;


                    if (sendEsclavos <= 4)
                    {
                        //PETICION VALOR TEMPERATURA
                        comando = sendEsclavos.ToString() + "ST" + enter;

                    }
                    else
                    {
                        if (sendEsclavos > 4 & sendEsclavos <= 6)
                        {
                            //PETICION VALOR INUNDACION
```

```
                num = sendEsclavos - 4;
                comando = num.ToString() + "SI" + enter;

            }
            else
            {
                if (sendEsclavos > 6 & sendEsclavos <= 8)
                {
                    //PETICION VALOR HUMO
                    num = sendEsclavos - 6;
                    comando = num.ToString() + "SH" + enter;

                }
                else
                {
                    if (sendEsclavos > 8 & sendEsclavos <= 10)
                    {
                        //PETICION VALOR CORRIENTE
                        num = sendEsclavos - 8;
                        comando = num.ToString() + "SC" + enter;
                    }
                    else
                    {
                        if (sendEsclavos == 11)
                        {
                            //PETICION ESTADO MAGNETICO
                            comando = "1SM" + enter;

                        }
                        else
                        {
                            if (sendEsclavos==12 & checkDAQ==true)
                            {
                               comando= "1SV0001"+enter;
                               checkDAQ=false;
                            }
                            //else
                            //{
                            //    if (sendEsclavos == 13)
                            //    {
                            //        //PETICION ESTADO HUMEDAD
                            //        comando = "1SU" + enter;
                            //    }
                            //}
                            sendEsclavos = 0;
                        }
                    }
                }
            }
        }

            srlDAQ.WriteLine(comando);
            Thread.Sleep(100);

    }

     // Procesamiento de comandos enviados por Bluetooth:
     if (getBluetooth.Length > 4)
     {
         commBluet = getBluetooth.Substring(2);
         if (commBluet == "1AP0001" & srlDAQ.IsOpen == true)
         {
             permisoM1 = false;
             commDAQ = "1AP0001";
             srlDAQ.WriteLine(commDAQ);

         }
```

```
                else if ((commBluet ==
"UDCST"||commBluet=="UDCSI"||commBluet=="UDCSH")& srlBluetooth.IsOpen == true)
                {
                    if (commBluet=="UDCST")
                    {
                        try
                        {
                            if
((valueT1!="ERR1"&valueT2!="ERR1")&(valueT1!="ERR2"&valueT2!="ERR2"))
                            {
                                auxBluet = Convert.ToInt16(valueT2) +
Convert.ToInt16(valueT1);
                                auxBluet = auxBluet / 2;
                                sensorValueTecnico = auxBluet.ToString();
                            }
                            else if (valueT1=="ERR1"||valueT1=="ERR2")
                            {
                                auxBluet = Convert.ToInt16(valueT2);
                                sensorValueTecnico = auxBluet.ToString();
                            }
                            else if (valueT2 == "ERR1" || valueT2 == "ERR2")
                            {
                                auxBluet = Convert.ToInt16(valueT1);
                                sensorValueTecnico = auxBluet.ToString();
                            }
                        }
                        catch
                        {
                            sensorValueTecnico = "0000";
                        }

                        try
                        {
                            if ((valueT3 != "ERR1" & valueT4 != "ERR1") &
(valueT3 != "ERR2" & valueT4 != "ERR2"))
                            {
                                auxBluet = Convert.ToInt16(valueT3) +
Convert.ToInt16(valueT4);
                                auxBluet = auxBluet / 2;
                                sensorValueTaller = auxBluet.ToString();
                            }
                            else if (valueT3 == "ERR1" || valueT3 == "ERR2")
                            {
                                auxBluet = Convert.ToInt16(valueT4) ;
                                sensorValueTaller = auxBluet.ToString();
                            }
                            else if (valueT4 == "ERR1" || valueT4 == "ERR2")
                            {
                                auxBluet = Convert.ToInt16(valueT3);
                                sensorValueTaller = auxBluet.ToString();
                            }

                        }
                        catch
                        {
                            sensorValueTaller = "0000";
                        }

                    }else if (commBluet == "UDCSI")
                    {
                        if (valueI1 != "")
                        {
                            sensorValueTaller = valueI1;
                        }
                        else
                        {
                            sensorValueTaller = "0100";
```

```
                    }

                    if (valueI2 != "")
                    {
                        sensorValueTecnico = valueI2;
                    }
                    else
                    {
                        sensorValueTecnico = "0100";
                    }
                }else if (commBluet == "UDCSH")
                {
                    if (valueH1 != "")
                    {
                        sensorValueTaller = valueH1;
                    }
                    else
                    {
                        sensorValueTaller = "0100";
                    }

                    if (valueH2 != "")
                    {
                        sensorValueTecnico = valueH2;
                    }
                    else
                    {
                        sensorValueTecnico = "0100";
                    }
                }

                for (int i = 0; i < 4; i++)
                {
               Thread.Sleep(10);
               srlBluetooth.Write("000"+sensorValueTaller.Substring(i, 1));
                }

                for (int j = 0; j < 4; j++)
                {
                    Thread.Sleep(10);
                    srlBluetooth.Write("000" +
sensorValueTecnico.Substring(j, 1));
                }
            }
            else if ((commBluet == "UDCSC" || commBluet == "UDCSM") &
srlBluetooth.IsOpen == true)
            {
                if (commBluet == "UDCSC")
                {
                    if (valueC1 != "")
                    {
                        sensorValueTaller = valueC1;
                    }
                    else
                    {
                        sensorValueTaller = "0000";
                    }

                }
                else if (commBluet == "UDCSM")
                {
                    if (valueM1 != "")
                    {
                        sensorValueTaller = valueM1;
                    }
                    else
                    {
```

```
                            sensorValueTaller = "0100";
                        }

                    }

                    for (int k = 0; k < 4; k++)
                    {
                        Thread.Sleep(10);
                        srlBluetooth.Write("000" +
sensorValueTaller.Substring(k, 1));
                    }
                }


                commBluet = "";
                getBluetooth = "";

            }

          //DETERMINACIÓN DE ALARMAS:

            //Temperatura:
            //T1:
            if (valueT1 != "ERR1" & valueT1 != "ERR2")
            {
                alarmErr1T1 = false;
                alarmErr2T1 = false;
                try
                {
                    Convert.ToInt16(valueT1);
                    convertir = true;
                }
                catch
                {
                    convertir = false;
                }

                if (convertir == true)
                {
                    limite = lecturaparametrossensores("maxT1");
                    TempMaxDptoTecnico = limite*100;
                    conversionvalueT1 = Convert.ToInt16(valueT1);
                    if (Convert.ToInt16(valueT1) > limite*100)
                    {
                        alarmT1 = true;
                        alarmvalueT1 = Convert.ToInt16(valueT1);
                    }
                    else
                    {
                        alarmT1 = false;
                    }
                }
                else
                {
                    conversionvalueT1 = 0;
                }

            }
            else
            {
                if (valueT1 == "ERR1")
                {
                    alarmErr1T1 = true;
                    alarmErr2T1 = false;
                }
                else
                {
```

```
            alarmErr1T1 = false;
            if (valueT1 == "ERR2")
            {
                alarmErr2T1 = true;
            }
            else
            {
                alarmErr2T1 = false;
            }
        }
    }

    //T2:
    if (valueT2 != "ERR1" & valueT2 != "ERR2")
    {
        alarmErr1T2 = false;
        alarmErr2T2 = false;
        try
        {
            Convert.ToInt16(valueT2);
            convertir = true;
        }
        catch
        {
            convertir = false;
        }

        if (convertir == true)
        {
            limite = lecturaparametrossensores("maxT2");
            TempMaxDptoTecnico = limite * 100;
            conversionvalueT2 = Convert.ToInt16(valueT2);

            if (Convert.ToInt16(valueT2) > limite*100)
            {
                alarmT2 = true;
                alarmvalueT2 = Convert.ToInt16(valueT2);
            }
            else
            {
                alarmT2 = false;
            }
        }
        else
        {
            conversionvalueT2 = 0;
        }
    }
    else
    {
        if (valueT2 == "ERR1")
        {
            alarmErr1T2 = true;
            alarmErr2T2 = false;
        }
        else
        {
            alarmErr1T2 = false;
            if (valueT2 == "ERR2")
            {
                alarmErr2T2 = true;
            }
            else
            {
                alarmErr2T2 = false;
            }
        }
```

```
                }
                //Promedio T1 T2:
                if ((conversionvalueT1 != 0 & conversionvalueT2 != 0) ||
(conversionvalueT1 == 0 & conversionvalueT2 == 0))
                {
                    valueDptoTecnico = (conversionvalueT1 + conversionvalueT2) / 2;
                }
                else if (conversionvalueT1==0 & conversionvalueT2!=0)
                {
                    valueDptoTecnico = conversionvalueT2;
                }
                else if (conversionvalueT2 == 0 & conversionvalueT1 != 0)
                {
                    valueDptoTecnico = conversionvalueT1;
                }

                //T3:
                if (valueT3 != "ERR1" & valueT3 != "ERR2")
                {
                    alarmErr1T3 = false;
                    alarmErr2T3 = false;
                    try
                    {
                        Convert.ToInt16(valueT3);
                        convertir = true;
                    }
                    catch
                    {
                        convertir = false;
                    }

                    if (convertir == true)
                    {
                        limite = lecturaparametrossensores("maxT3");
                        TempMaxTaller = limite * 100;
                        conversionvalueT3= Convert.ToInt16(valueT3);

                        if (Convert.ToInt16(valueT3) > limite*100)
                        {
                            alarmT3 = true;
                            alarmvalueT3 = Convert.ToInt16(valueT3);
                        }
                        else
                        {
                            alarmT3 = false;
                        }
                    }
                    else
                    {
                        conversionvalueT3 = 0;
                    }
                }
                else
                {
                    if (valueT3 == "ERR1")
                    {
                        alarmErr1T3 = true;
                        alarmErr2T3 = false;
                    }
                    else
                    {
                        alarmErr1T3 = false;
                        if (valueT3 == "ERR2")
                        {
                            alarmErr2T3 = true;
                        }
                        else
```

```
            {
                alarmErr2T3 = false;
            }

        }
    }

    //T4:
    if (valueT4 != "ERR1" & valueT4 != "ERR2")
    {
        alarmErr1T4 = false;
        alarmErr2T4 = false;

        try
        {
            Convert.ToInt16(valueT4);
            convertir = true;
        }
        catch
        {
            convertir = false;
        }

        if (convertir == true)
        {
            limite = lecturaparametrossensores("maxT4");
            TempMaxTaller = limite * 100;
            conversionvalueT4 = Convert.ToInt16(valueT4);

            if (Convert.ToInt16(valueT4) > limite*100)
            {
                alarmT4 = true;
                alarmvalueT4 = Convert.ToInt16(valueT4);
            }
            else
            {
                alarmT4 = false;
            }
        }
        else
        {
            conversionvalueT4 = 0;
        }
    }
    else
    {
        if (valueT4 == "ERR1")
        {
            alarmErr1T4 = true;
            alarmErr2T4 = false;
        }
        else
        {
            alarmErr1T4 = false;
            if (valueT4 == "ERR2")
            {
                alarmErr2T4 = true;
            }
            else
            {
                alarmErr2T4 = false;
            }

        }
    }

    //Promedio T3 T4:
```

```
            if ((conversionvalueT3 != 0 & conversionvalueT4 != 0) ||
(conversionvalueT3 == 0 & conversionvalueT4 == 0))
            {
                valueTaller = (conversionvalueT3 + conversionvalueT4) / 2;
            }
            else if (conversionvalueT3 == 0 & conversionvalueT4 != 0)
            {
                valueTaller = conversionvalueT4;
            }
            else if (conversionvalueT4 == 0 & conversionvalueT3 != 0)
            {
                valueTaller = conversionvalueT3;
            }

               //Corriente:
            //C1:

            if (valueC1 == "ERR2")
            {
                alarmErr2C1 = true;
            }
            else
            {
                alarmErr2C1 = false;

                try
                {
                    Convert.ToInt16(valueC1);
                    convertir = true;
                }
                catch
                {
                    convertir = false;
                }

                if (convertir == true)
                {
                    //limite máximo:
                    limite = lecturaparametrossensores("maxC1");

                    if (Convert.ToInt16(valueC1) > limite*100)
                    {
                        alarmMaxC1 = true;
                        alarmvalueC1 = Convert.ToInt16(valueC1);
                    }
                    else
                    {
                        alarmMaxC1 = false;
                    }

                    //límite mínimo:
                    limite = lecturaparametrossensores("minC1");

                    if (Convert.ToInt16(valueC1) < limite*100)
                    {
                        alarmMinC1 = true;
                        alarmvalueC1 = Convert.ToInt16(valueC1);
                    }
                    else
                    {
                        alarmMinC1 = false;
                    }
                }

            }

            //C2:
```

```
if (valueC2 == "ERR2")
{
    alarmErr2C2 = true;
}
else
{
    alarmErr2C2 = false;
    try
    {
        Convert.ToInt16(valueC2);
        convertir = true;
    }
    catch
    {
        convertir = false;
    }

    if (convertir == true)
    {
        //límite máximo:
        limite = lecturaparametrossensores("maxC2");
        if (Convert.ToInt16(valueC2) > limite*100)
        {
            alarmMaxC2 = true;
        }
        else
        {
            alarmMaxC2 = false;
        }

        //límite mínimo:
        limite = lecturaparametrossensores("minC2");
        if (Convert.ToInt16(valueC2) < limite*100)
        {
            alarmMinC2 = true;
        }
        else
        {
            alarmMinC2 = false;
        }
    }
}

   //Humo:
  //H1:
    try
    {
        Convert.ToInt16(valueH1);
        convertir = true;
    }
    catch
    {
        convertir = false;
    }

    if (convertir == true)
    {
        if (Convert.ToInt16(valueH1) == 1)
        {
            alarmH1 = true;
        }
        else
        {
            alarmH1 = false;
        }
    }
```

```
    //H2:
    try
    {
        Convert.ToInt16(valueH2);
        convertir = true;
    }
    catch
    {
        convertir = false;
    }

    if (convertir == true)
    {
        if (Convert.ToInt16(valueH2) == 1)
        {
            alarmH2 = true;
        }
        else
        {
            alarmH2 = false;
        }
    }

  //Magnético:
    try
    {
        Convert.ToInt16(valueM1);
        convertir = true;
    }
    catch
    {
        convertir = false;
    }

    if (convertir == true)
    {
        limite = lecturaparametrossensores("magnetico");
        if (Convert.ToInt16(valueM1) == 1)
        {
            alarmM1 = true;
            if (limite == 1)
            {
                enableAlarmMagnetic = true;

            }
            else
            {
                enableAlarmMagnetic = false;
            }
        }
        else
        {
            alarmM1 = false;
        }

        if (limite == 1 & enableTimerMagnetic==true)
        {
            flagEnableTimerMagnetic = true;
            enableTimerMagnetic = false;
        }
        else if (limite==0)
        {
            enableTimerMagnetic = true;
        }

    }
```

```
//Inundación:
 //I1:
     try
     {
         Convert.ToInt16(valueI1);
         convertir = true;
     }
     catch
     {
         convertir = false;
     }

     if (convertir == true)
     {
         if (Convert.ToInt16(valueI1) == 1)
         {
             alarmI1 = true;
             alarmErr2I1 = false;
         }
         else if (Convert.ToInt16(valueI1) == 0)
         {
             alarmI1 = false;
             alarmErr2I1 = false;
         }

     }
     else
     {
         if (valueI1=="ERR2")
         {
             alarmErr2I1 = true;

         }
     }
 //I2:
     try
     {
         Convert.ToInt16(valueI2);
         convertir = true;
     }
     catch
     {
         convertir = false;
     }

     if (convertir == true)
     {
         if (Convert.ToInt16(valueI2) == 1)
         {
             alarmI2 = true;
             alarmErr2I2 = false;
         }
         else if (Convert.ToInt16(valueI2) == 0)
         {
             alarmI2 = false;
             alarmErr2I2 = false;
         }

     }
     else
     {
         if (valueI2 == "ERR2")
         {
             alarmErr2I2 = true;

         }
     }
```

```
            }
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            banderasalida = true;
            tmrSendMaster.Enabled = false;
            try
            {
                HDaq.Abort();
                HDaq.Join();
            }
            catch
            {
                MessageBox.Show("Error al cerrar el hilo.", "Error",
MessageBoxButtons.OK);
            }
            try
            {
                connection.Close();
                connectionthread.Close();
            }
            catch
            {
                MessageBox.Show("Error al cerrar base de
datos.","Error",MessageBoxButtons.OK);
            }

            if (srlDAQ.IsOpen == true)
            {
                try
                {
                    srlDAQ.Close();
                }
                catch
                {
                    MessageBox.Show("No se puede cerrar el puerto", "ERROR AL
SALIR", MessageBoxButtons.OK);
                }
            }

            try
            {
                Application.Exit();
            }
            catch
            {

            }

        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            //Dato enviado por bluetooth
            if (auxBluetooth.Length>4)
            {
               textBox1.Text = auxBluetooth.Substring(2);

            }


            try
            {
                aux = Convert.ToInt16(valueT1);
                txtT1.Text = Convert.ToString(aux/100);
            }
```

```
catch
{
    txtT1.Text = valueT1;
}

try
{
    aux = Convert.ToInt16(valueT2);
    txtT2.Text = Convert.ToString(aux/100);
}
catch
{
    txtT2.Text = valueT2;
}

try
{
    aux = Convert.ToInt16(valueT3);
    txtT3.Text = Convert.ToString(aux/100);
}
catch
{
    txtT3.Text = valueT3;
}

try
{
    aux = Convert.ToInt16(valueT4);
    txtT4.Text = Convert.ToString(aux / 100);
}
catch
{
    txtT4.Text = valueT4;
}

try
{
    aux = Convert.ToInt16(valueC1);
    txtC1.Text = Convert.ToString(aux / 100);
    if (nodata==true)
    {
        txtC1.Text = "";
    }
}
catch
{
    txtC1.Text = valueC1;
}

txtC2.Text = valueC2;

if (valueI1=="0001")
{
    txtI1.Text = "Activado";
}
else if (valueI1=="0000")
{
    txtI1.Text = "Desactivado";
}
else if (valueI1 == "ERR2")
{
    txtI1.Text = "Error Tarjeta";
}

if (valueI2 == "0001")
{
    txtI2.Text = "Activado";
```

```
}
else if (valueI2 == "0000")
{
    txtI2.Text = "Desactivado";
}
else if (valueI2 == "ERR2")
{
    txtI2.Text = "Error Tarjeta";
}

if (valueM1 == "0001")
{
    txtM1.Text = "Activado";
}
else if (valueI1 == "0000")
{
    txtM1.Text = "Desactivado";
}

if (valueH1 == "0001")
{
    txtH1.Text = "Activado";
}
else if (valueH1 == "0000")
{
    txtH1.Text = "Desactivado";
}

if (valueH2 == "0001")
{
    txtH2.Text = "Activado";
}
else if (valueH2 == "0000")
{
    txtH2.Text = "Desactivado";
}

string enablesms = "";
string enablemail = "";
string textoadd="";

//ALARMAS:
    //T1:
if (alarmT1 == true)
{
    lblT1.Text = "Temperatura1";
    lblT1.ForeColor = System.Drawing.Color.Red;
    pbSensorTAlarm1.Visible = true;
    pbSensorTErr1.Visible = false;
    pbSensorTNormal.Visible = false;

}
else
{
    if (alarmErr1T1 == true)
    {
        lblT1.Text = "Temperatura1\n(No hay sensor)";
        lblT1.ForeColor = System.Drawing.Color.Black;
        pbSensorTAlarm1.Visible = false;
        pbSensorTErr1.Visible = true;
        pbSensorTNormal.Visible = false;
    }
    else
    {
        if (alarmErr2T1 == true)
        {
            lblT1.Text = "Temperatura1\n(Tarjeta no conectada)";
```

```
                    lblT1.ForeColor = System.Drawing.Color.Black;
                    pbSensorTAlarm1.Visible = false;
                    pbSensorTErr1.Visible = true;
                    pbSensorTNormal.Visible = false;
                }
                else if (nodata == false)
                {
                    lblT1.Text = "Temperatura1";
                    lblT1.ForeColor = System.Drawing.Color.Black;
                    pbSensorTAlarm1.Visible = false;
                    pbSensorTErr1.Visible = false;
                    pbSensorTNormal.Visible = true;
                }

            }
        }

            //T2:
        if (alarmT2 == true)
        {
            lblT2.Text = "Temperatura2";
            lblT2.ForeColor = System.Drawing.Color.Red;
            pbSensorTErr2.Visible = false;
            pbSensorTAlarm2.Visible = true;
            pbSensorTNormal2.Visible = false;
        }
        else
        {
            if (alarmErr1T2 == true)
            {
                lblT2.Text = "Temperatura2\n(No hay sensor)";
                lblT2.ForeColor = System.Drawing.Color.Black;
                pbSensorTErr2.Visible = true;
                pbSensorTAlarm2.Visible = false;
                pbSensorTNormal2.Visible = false;
             }
            else
            {
                if (alarmErr2T2 == true)
                {
                    lblT2.Text = "Temperatura2\n(Tarjeta no conectada)";
                    lblT2.ForeColor = System.Drawing.Color.Black;
                    pbSensorTErr2.Visible = true;
                    pbSensorTAlarm2.Visible = false;
                    pbSensorTNormal2.Visible = false;
                }
                else if (nodata == false)
                {
                    lblT2.Text = "Temperatura2";
                    lblT2.ForeColor = System.Drawing.Color.Black;
                    pbSensorTErr2.Visible = false;
                    pbSensorTAlarm2.Visible = false;
                    pbSensorTNormal2.Visible = true;
                }

            }
        }

        // Promedio T1 T2 (Barra de progreso)
        if (TempMaxDptoTecnico!=0)
        {
            lblDptoTecAverageT.Text = Convert.ToString(valueDptoTecnico/100);
            porcentajeBarraProgresoTemp = Convert.ToInt16((valueDptoTecnico *
100) / TempMaxDptoTecnico);
            if (porcentajeBarraProgresoTemp>100)
            {
                pbTemperatureDptoTecnico.Value = 100;
```

```
                }
                else
                {
                    pbTemperatureDptoTecnico.Value = porcentajeBarraProgresoTemp;
                }
                if (pbTemperatureDptoTecnico.Value <= 33)
                {
                    pbTemperatureDptoTecnico.ForeColor =
System.Drawing.Color.Blue;
                }
                else if (pbTemperatureDptoTecnico.Value > 33 &
pbTemperatureDptoTecnico.Value <= 66)
                {
                    pbTemperatureDptoTecnico.ForeColor =
System.Drawing.Color.Yellow;
                }
                else if (pbTemperatureDptoTecnico.Value > 66 &
pbTemperatureDptoTecnico.Value <= 100)
                {
                    pbTemperatureDptoTecnico.ForeColor =
System.Drawing.Color.Red;
                }
            }

            // Sin Datos Departamento Tecnico Temperatura:
            if (valueDptoTecnico == 0 || nodata==true)
            {
                pbDepTecTNoData.Visible = true;
            }
            else
            {
                pbDepTecTNoData.Visible = false;
            }

            //T3:
            if (alarmT3 == true)
            {
                lblT3.Text = "Temperatura1";
                lblT3.ForeColor = System.Drawing.Color.Red;
                pbSensorTErr3.Visible = false;
                pbSensorTAlarm3.Visible = true;
                pbSensorTNormal3.Visible = false;
            }
            else
            {
                if (alarmErr1T3 == true)
                {
                    lblT3.Text = "Temperatura1\n(No hay sensor)";
                    lblT3.ForeColor = System.Drawing.Color.Black;
                    pbSensorTErr3.Visible = true;
                    pbSensorTAlarm3.Visible = false;
                    pbSensorTNormal3.Visible = false;
                }
                else
                {
                    if (alarmErr2T3 == true)
                    {
                        lblT3.Text = "Temperatura1\n(Tarjeta no conectada)";
                        lblT3.ForeColor = System.Drawing.Color.Black;
                        pbSensorTErr3.Visible = true;
                        pbSensorTAlarm3.Visible = false;
                        pbSensorTNormal3.Visible = false;
                    }
                    else if (nodata == false)
                    {
                        lblT3.Text = "Temperatura1";
                        lblT3.ForeColor = System.Drawing.Color.Black;
```

```
                    pbSensorTErr3.Visible = false;
                    pbSensorTAlarm3.Visible = false;
                    pbSensorTNormal3.Visible = true;
                }

            }
        }

        //T4:
        if (alarmT4 == true)
        {
            lblT4.Text = "Temperatura2";
            lblT4.ForeColor = System.Drawing.Color.Red;
            pbSensorTErr4.Visible = false;
            pbSensorTAlarm4.Visible = true;
            pbSensorTNormal4.Visible = false;
        }
        else
        {
            if (alarmErr1T4 == true)
            {
                lblT4.Text = "Temperatura2\n(No hay sensor)";
                lblT4.ForeColor = System.Drawing.Color.Black;
                pbSensorTErr4.Visible = true;
                pbSensorTAlarm4.Visible = false;
                pbSensorTNormal4.Visible = false;
            }
            else
            {
                if (alarmErr2T4 == true)
                {
                    lblT4.Text = "Temperatura2\n(Tarjeta no conectada)";
                    lblT4.ForeColor = System.Drawing.Color.Black;
                    pbSensorTErr4.Visible = true;
                    pbSensorTAlarm4.Visible = false;
                    pbSensorTNormal4.Visible = false;
                }
                else if (nodata == false)
                {
                    lblT4.Text = "Temperatura2";
                    lblT4.ForeColor = System.Drawing.Color.Black;
                    pbSensorTErr4.Visible = false;
                    pbSensorTAlarm4.Visible = false;
                    pbSensorTNormal4.Visible = true;
                }

            }
        }
        // Promedio T3 T4 (Barra de progreso)
        if (TempMaxTaller!=0)
        {
            lblTallerAverageT.Text = Convert.ToString(valueTaller / 100);
            porcentajeBarraProgresoTemp = Convert.ToInt16((valueTaller * 100)
/ TempMaxTaller);
            if (porcentajeBarraProgresoTemp > 100)
            {
                pbTemperatureTaller.Value = 100;
            }
            else
            {
                pbTemperatureTaller.Value = porcentajeBarraProgresoTemp;
            }
            if (pbTemperatureTaller.Value <= 33)
            {
                pbTemperatureTaller.ForeColor = System.Drawing.Color.Blue;
            }
```

```
                else if (pbTemperatureTaller.Value > 33 &
pbTemperatureTaller.Value <= 66)
                {
                    pbTemperatureTaller.ForeColor = System.Drawing.Color.Yellow;
                }
                else if (pbTemperatureTaller.Value > 66 &
pbTemperatureTaller.Value <= 100)
                {
                    pbTemperatureTaller.ForeColor = System.Drawing.Color.Red;
                }
            }

            // Sin Datos Taller Temperatura:
            if (valueTaller == 0 || nodata==true)
            {
                pbTallerTNoData.Visible = true;
            }
            else
            {
                pbTallerTNoData.Visible = false;
            }

            //C1:
            if (alarmMaxC1 == true)
            {
                lblC1.Text = "Corriente";
                lblC1.ForeColor = System.Drawing.Color.Red;
                pbCurrentHigh.Visible = true;
                pbCurrentLow.Visible = false;
                pbCurrentNormal.Visible = false;
                pbSensorCurrentAlarm.Visible = true;
                pbSensorCurrentErr.Visible = false;
                pbSensorCurrentNormal.Visible = false;
            }
            else
            {
                if (alarmMinC1 == true)
                {
                    lblC1.Text = "Corriente";
                    lblC1.ForeColor = System.Drawing.Color.Blue;
                    pbCurrentHigh.Visible = false;
                    pbCurrentLow.Visible = true;
                    pbCurrentNormal.Visible = false;
                    pbSensorCurrentAlarm.Visible = true;
                    pbSensorCurrentErr.Visible = false;
                    pbSensorCurrentNormal.Visible = false;
                    pbC1NoData.Visible = false;
                }
                else
                {
                    if (alarmErr2C1 == true)
                    {
                        lblC1.Text = "Corriente\n(Tarjeta no conectada)";
                        lblC1.ForeColor = System.Drawing.Color.Black;
                        pbCurrentHigh.Visible = false;
                        pbCurrentLow.Visible = false;
                        pbCurrentNormal.Visible = false;
                        pbSensorCurrentAlarm.Visible = false;
                        pbSensorCurrentErr.Visible = true;
                        pbSensorCurrentNormal.Visible = false;
                        pbC1NoData.Visible = true;
                    }
                    else if(nodata==false)
                    {
                        lblC1.Text = "Corriente";
                        lblC1.ForeColor = System.Drawing.Color.Black;
                        pbCurrentHigh.Visible = false;
```

```
            pbCurrentLow.Visible = false;
            pbCurrentNormal.Visible = true;
            pbSensorCurrentAlarm.Visible = false;
            pbSensorCurrentErr.Visible = false;
            pbSensorCurrentNormal.Visible = true;
            pbC1NoData.Visible = false;
        }
    }
}

//C2:
if (alarmMaxC2 == true)
{
    lblC2.Text = "Corriente2";
    lblC2.ForeColor = System.Drawing.Color.Red;
}
else
{
    if (alarmMinC2 == true)
    {
        lblC2.Text = "Corriente2";
        lblC2.ForeColor = System.Drawing.Color.Blue;
    }
    else
    {
        if (alarmErr2C2 == true)
        {
            lblC2.Text = "Corriente2\n(Tarjeta no conectada)";
            lblC2.ForeColor = System.Drawing.Color.Black;
        }
        else
        {
            lblC2.Text = "Corriente2";
            lblC2.ForeColor = System.Drawing.Color.Black;
        }
    }
}

//H1:
if (alarmH1 == true)
{
    lblH1.Text = "Humo";
    lblH1.ForeColor = System.Drawing.Color.Red;
    pbSmokeOn.Visible = true;
    pbSmokeOff.Visible = false;
    pbSensorHAlarm.Visible = true;
    pbSensorHNormal.Visible = false;
    pbSensorHumo1Err.Visible = false;
    pbTallerHNoData.Visible = false;
}
else if (nodata == false)
{
        lblH1.Text = "Humo";
        lblH1.ForeColor = System.Drawing.Color.Black;
        pbSmokeOn.Visible = false;
        pbSmokeOff.Visible = true;
        pbSensorHAlarm.Visible = false;
        pbSensorHNormal.Visible = true;
        pbSensorHumo1Err.Visible = false;
        pbTallerHNoData.Visible = false;
 }

//H2:
if (alarmH2 == true)
{
    lblH2.Text = "Humo";
    lblH2.ForeColor = System.Drawing.Color.Red;
```

```
    pbSmokeOn2.Visible = true;
    pbSmokeOff2.Visible = false;
    pbSensorHAlarm2.Visible = true;
    pbSensorHumoNormal2.Visible = false;
    pbSensorHumo2Err.Visible = false;
    pbDepTecHNoData.Visible = false;
}
else if (nodata == false)
{
        lblH2.Text = "Humo";
        lblH2.ForeColor = System.Drawing.Color.Black;
        pbSmokeOn2.Visible = false;
        pbSmokeOff2.Visible = true;
        pbSensorHAlarm2.Visible = false;
        pbSensorHumoNormal2.Visible = true;
        pbSensorHumo2Err.Visible = false;
        pbDepTecHNoData.Visible = false;
}

//M1:
if (alarmM1 == true)
{
    lblM1.Text = "Magnético";
    lblM1.ForeColor = System.Drawing.Color.Red;
    pbDoorOpened.Visible = true;
    pbDoorClosed.Visible = false;
    pbSensorMAlarm.Visible = true;
    pbSensorMNormal.Visible = false;
    pbSensorMErr.Visible = false;
    pbM1NoData.Visible = false;
}
else if (nodata == false)
{
    lblM1.Text = "Magnético";
    lblM1.ForeColor = System.Drawing.Color.Black;
    pbDoorOpened.Visible = false;
    pbDoorClosed.Visible = true;
    pbSensorMAlarm.Visible = false;
    pbSensorMNormal.Visible = true;
    pbM1NoData.Visible = false;
    pbSensorMErr.Visible = false;
}

//I1:
if (alarmI1 == true)
{
    lblI1.Text = "Inundación";
    lblI1.ForeColor = System.Drawing.Color.Red;
    pbWaterOn.Visible = true;
    pbWaterOff1.Visible = false;
    pbSensorIAlarm.Visible = true;
    pbSensorINormal.Visible = false;
    pbSensorIErr.Visible = false;
    pbTallerINoData.Visible = false;
}
else
{
    if (alarmErr2I1 == true)
    {
        lblI1.Text = "Inundación\n(Tarjeta no conectada)";
        lblI1.ForeColor = System.Drawing.Color.Black;
        pbWaterOn.Visible = false;
        pbWaterOff1.Visible = false;
        pbSensorIAlarm.Visible = false;
        pbSensorINormal.Visible = false;
        pbSensorIErr.Visible = true;
        pbTallerINoData.Visible = true;
```

```
        }
        else if (nodata == false)
        {
            lblI1.Text = "Inundación";
            lblI1.ForeColor = System.Drawing.Color.Black;
            pbWaterOn.Visible = false;
            pbWaterOff1.Visible = true;
            pbSensorIAlarm.Visible = false;
            pbSensorINormal.Visible = true;
            pbSensorIErr.Visible = false;
            pbTallerINoData.Visible = false;
        }
    }

    //I2:
    if (alarmI2 == true)
    {
        lblI2.Text = "Inundación";
        lblI2.ForeColor = System.Drawing.Color.Red;
        pbWaterOn2.Visible = true;
        pbWaterOff2.Visible = false;
        pbSensorIAlarm2.Visible = true;
        pbSensorINormal2.Visible = false;
        pbSensorIErr2.Visible = false;
        pbDepTecINoData.Visible = false;
    }
    else
    {
        if (alarmErr2I2 == true)
        {

            lblI2.Text = "Inundación\n(Tarjeta no conectada)";
            lblI2.ForeColor = System.Drawing.Color.Black;
            pbWaterOn2.Visible = false;
            pbWaterOff2.Visible = false;
            pbSensorIAlarm2.Visible = false;
            pbSensorINormal2.Visible = false;
            pbSensorIErr2.Visible = true;
            pbDepTecINoData.Visible = true;

        }
        else if (nodata == false)
        {
            lblI2.Text = "Inundación";
            lblI2.ForeColor = System.Drawing.Color.Black;
            pbWaterOn2.Visible = false;
            pbWaterOff2.Visible = true;
            pbSensorIAlarm2.Visible = false;
            pbSensorINormal2.Visible = true;
            pbSensorIErr2.Visible = false;
            pbDepTecINoData.Visible = false;
        }
    }

    if (daqDisconnected == true)
    {
        //Sensores desconectados:
        pbDepTecTNoData.Visible = true;
        //T1:
        pbSensorTAlarm1.Visible = false;
        pbSensorTErr1.Visible = true;
        pbSensorTNormal.Visible = false;
        txtT1.Text = "";
        //T2:
        pbSensorTErr2.Visible = true;
        pbSensorTAlarm2.Visible = false;
```

```
                pbSensorTNormal2.Visible = false;
                txtT2.Text = "";

                pbTallerTNoData.Visible = true;
                //T3:
                pbSensorTErr3.Visible = true;
                pbSensorTAlarm3.Visible = false;
                pbSensorTNormal3.Visible = false;
                txtT3.Text = "";
                //T4:
                pbSensorTErr4.Visible = true;
                pbSensorTAlarm4.Visible = false;
                pbSensorTNormal4.Visible = false;
                txtT4.Text = "";
                //C1:
                pbSensorCurrentAlarm.Visible = false;
                pbSensorCurrentErr.Visible = true;
                pbSensorCurrentNormal.Visible = false;
                pbC1NoData.Visible = true;
                txtC1.Text = "";
                //H1:
                pbSensorHAlarm.Visible = false;
                pbSensorHNormal.Visible = false;
                pbSensorHumo1Err.Visible = true;
                pbTallerHNoData.Visible = true;
                txtH1.Text = "";
                //H2:
                pbSensorHAlarm2.Visible = false;
                pbSensorHumoNormal2.Visible = false;
                pbSensorHumo2Err.Visible = true;
                pbDepTecHNoData.Visible = true;
                txtH2.Text = "";
                //I1:
                pbSensorIAlarm.Visible = false;
                pbSensorINormal.Visible = false;
                pbSensorIErr.Visible = true;
                pbTallerINoData.Visible = true;
                txtI1.Text = "";
                //I2:
                pbSensorIAlarm2.Visible = false;
                pbSensorINormal2.Visible = false;
                pbSensorIErr2.Visible = true;
                pbDepTecINoData.Visible = true;
                txtI2.Text = "";
                //M1:
                pbSensorMAlarm.Visible = false;
                pbSensorMNormal.Visible = false;
                pbSensorMErr.Visible = true;
                pbM1NoData.Visible = true;
                txtM1.Text = "";
            }

            //Habilitacion Mensaje de Alarma de sensores:
           if ((alarmH1==true || alarmH2==true || alarmI1==true || alarmI2==true
|| alarmMaxC1==true || alarmMinC1==true || alarmT1==true || alarmT2==true ||
alarmT3==true || alarmT4==true) & (enableAlarmMessage==true &
tmrAlarmMessage.Enabled==false))
            {
               enableAlarmMessage = false;
               DialogResult aceptar = MessageBox.Show("ALARMA", "PELIGRO",
MessageBoxButtons.OK,MessageBoxIcon.Warning);
               if (aceptar == DialogResult.OK)
               {
                   tmrAlarmMessage.Enabled = true;
               }
            }
```

```
            else if (alarmH1==false & alarmH2==false & alarmI1==false &
alarmI2==false & alarmMaxC1==false & alarmMinC1==false & alarmT1==false &
alarmT2==false & alarmT3==false & alarmT4==false)
            {
                enableAlarmMessage = true;
                tmrAlarmMessage.Enabled = false;
            }

             //Habilitacion timer para envio de alarmas del sensor magnetico:
             if (flagEnableTimerMagnetic==true)
             {
                tmrMagnetic.Enabled = true;
             }

             //ENVIO SMS y MAIL

             if ((alarmT1 == true & permisoT1 == true) || (alarmT2 == true &
permisoT2 == true) || (alarmT3 == true & permisoT3 == true) || (alarmT4 == true &
permisoT4 == true) || (alarmMaxC1 == true & permisoC1 == true) || (alarmMinC1 ==
true & permisoC1 == true) || (alarmI1 == true & permisoI1 == true) || (alarmI2 ==
true & permisoI2 == true) || (alarmH1 == true & permisoH1 == true) || (alarmH2 ==
true & permisoH2 == true) || (alarmM1 == true & permisoM1 == true))
             {

                 MySqlCommand mySqlreadenablesmscommand = new MySqlCommand("SELECT
`conectividad`.`Value` FROM `controldomotico`.`conectividad` WHERE (`Servicio` =
'Sms')", connectionsmsmail);
                 MySqlDataReader readenablesms =
mySqlreadenablesmscommand.ExecuteReader();
                 while (readenablesms.Read())
                 {
                     enablesms = readenablesms[0].ToString();
                 }
                 readenablesms.Close();

                 MySqlCommand mySqlreadenablemailcommand = new
MySqlCommand("SELECT `conectividad`.`Value` FROM `controldomotico`.`conectividad`
WHERE (`Servicio` = 'Mail')", connectionsmsmail);
                 MySqlDataReader readenablemail =
mySqlreadenablemailcommand.ExecuteReader();
                 while (readenablemail.Read())
                 {
                     enablemail = readenablemail[0].ToString();
                 }
                 readenablemail.Close();

                 if (srlSMS.IsOpen == true & enablesms=="1")
                 {
                     string telefono = "";
                     string texto = "";
                     bool bandera = false;

                     for (int i = 1; i <maxsmsdest; i++)
                     {
                         MySqlCommand mySqlreaddestinatariocommand = new
MySqlCommand("SELECT `destinatariossms`.`Check` FROM
`controldomotico`.`destinatariossms` WHERE `id` = " + i.ToString() + " LIMIT 0 ,
30", connectionsmsmail);
                         MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();
                         while (readdestinatario.Read())
                         {
                             if (readdestinatario[0].ToString() == "1")
                             {
                                 bandera = true;
                             }
                         }
```

```
                        readdestinatario.Close();

                        if (bandera == true)
                        {
                            bandera = false;
                            MySqlCommand mySqlreaddestinatariophoncommand = new
MySqlCommand("SELECT `destinatariossms`.`Phone` FROM
`controldomotico`.`destinatariossms` WHERE `id` = " + i.ToString() + " LIMIT 0 ,
30", connectionsmsmail);
                            MySqlDataReader readdestinatariophon =
mySqlreaddestinatariophoncommand.ExecuteReader();
                            while (readdestinatariophon.Read())
                            {
                                telefono = readdestinatariophon[0].ToString();
                            }
                            readdestinatariophon.Close();

                            MySqlCommand mySqlreaddestinatariomessagecommand =
new MySqlCommand("SELECT `mensajesmsmail`.`Message` FROM
`controldomotico`.`mensajesmsmail` WHERE `Service` = 'Sms' LIMIT 0 , 30",
connectionsmsmail);
                            MySqlDataReader readdestinatariomessage =
mySqlreaddestinatariomessagecommand.ExecuteReader();
                            while (readdestinatariomessage.Read())
                            {
                                texto = readdestinatariomessage[0].ToString();
                            }
                            readdestinatariomessage.Close();

                            //Edicion de mensajes:
                            textoadd=edicionmensajes();
                            texto = texto + textoadd;
                            smsSend(telefono, texto);
                        }

                    }
                }

                if (enablemail=="1")
                {
                    string mail = "";
                    string texto = "";
                    bool banderamail = false;

                    for (int i = 1; i < maxnumberdestmail; i++)
                    {
                        MySqlCommand mySqlreaddestinatariomailcommand = new
MySqlCommand("SELECT `destinatariosmail`.`Check` FROM
`controldomotico`.`destinatariosmail` WHERE `id` = " + i.ToString() + " LIMIT 0 ,
30", connectionsmsmail);
                        MySqlDataReader readdestinatariomail =
mySqlreaddestinatariomailcommand.ExecuteReader();
                        while (readdestinatariomail.Read())
                        {
                            if (readdestinatariomail[0].ToString() == "1")
                            {
                                banderamail = true;
                            }
                        }
                        readdestinatariomail.Close();

                        if (banderamail == true)
                        {
                            banderamail = false;
                            MySqlCommand mySqlreaddestinatarioaddcommand = new
MySqlCommand("SELECT `destinatariosmail`.`Mail` FROM
```

```
`controldomotico`.`destinatariosmail` WHERE `id` = " + i.ToString() + " LIMIT 0 ,
30", connectionsmsmail);
                            MySqlDataReader readdestinatarioadd =
mySqlreaddestinatarioaddcommand.ExecuteReader();
                            while (readdestinatarioadd.Read())
                            {
                                mail = readdestinatarioadd[0].ToString();
                            }
                            readdestinatarioadd.Close();

                            MySqlCommand mySqlreaddestinatariotextcommand = new
MySqlCommand("SELECT `mensajesmsmail`.`Message` FROM
`controldomotico`.`mensajesmsmail` WHERE `Service` = 'Mail' LIMIT 0 , 30",
connectionsmsmail);
                            MySqlDataReader readdestinatariotext =
mySqlreaddestinatariotextcommand.ExecuteReader();
                            while (readdestinatariotext.Read())
                            {
                                texto = readdestinatariotext[0].ToString();
                            }
                            readdestinatariotext.Close();

                            //Edicion de mensajes:
                            textoadd = edicionmensajes();
                            texto = texto + textoadd;
                            sendmail(mail, texto);
                        }

                    }
                }
                tmrAlarmConfirm.Enabled = false;
                tmrAlarmConfirm.Enabled = true;
            }

        }

        private void srlDAQ_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            //RECEPCION DATOS:
            permiso = false;
            Thread.Sleep(200);
            try
            {
                if (srlDAQ.IsOpen == true)
                {

                    getDaq = srlDAQ.ReadExisting();
                }
                permiso = true;
            }
            catch
            {

            }
        }
        private void smsSend(string telefono, string mensaje)
        {
            string com = "";
            string enter = "" + (char)13;
            string comilla = "" + (char)34;
            string ctrlz = "" + (char)26;
            com = "ATE0" + enter;
            srlSMS.WriteLine(com);
            Thread.Sleep(100);
            com = "AT+CPMS=" + comilla + "SM" + comilla + enter;
            srlSMS.WriteLine(com);
```

```
            Thread.Sleep(100);
            com = "AT+CMGF=1" + enter;
            srlSMS.WriteLine(com);
            Thread.Sleep(100);
            com = "AT+CNMI=2,1,2,0,0" + enter;
            srlSMS.WriteLine(com);
            Thread.Sleep(100);
            com = "AT+CMGS=" + comilla + telefono + comilla + enter;
            srlSMS.WriteLine(com);
            Thread.Sleep(100);
            com = mensaje;
            srlSMS.WriteLine(com);
            Thread.Sleep(100);
            com = ctrlz;
            srlSMS.WriteLine(com);
            Thread.Sleep(100);
        }

        private int lecturaparametrossensores(string sensor)
        {
            int limite = 35;
            try
            {
            MySqlCommand mySqlreadparametroscommand = new MySqlCommand("SELECT
`sensores`.`Value` FROM `controldomotico`.`sensores` WHERE (`Parameter` = '" +
sensor + "')LIMIT 0 , 30", connectionthread);

                MySqlDataReader readparametros =
mySqlreadparametroscommand.ExecuteReader();
                while (readparametros.Read())
                {
                    limite =  Convert.ToInt16(readparametros[0].ToString());
                }
                readparametros.Close();

            }
            catch
            {
                if (banderasalida==false)
                {
                    MessageBox.Show("No se puede leer la base de datos de
sensores", "Lectura Base de Datos", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                }
            }
            return limite;

        }

        private void tmrAlarmConfirm_Tick(object sender, EventArgs e)
        {
            permisoT1 = true;
            permisoT2 = true;
            permisoT3 = true;
            permisoT4 = true;
            permisoC1 = true;
            permisoI1 = true;
            permisoI2 = true;
            permisoH2 = true;
            permisoH1 = true;
            if (enableAlarmMagnetic==true)
            {
                permisoM1 = true;
            }
            tmrAlarmConfirm.Enabled = false;
        }
```

```
        private string edicionmensajes()
        {
            string textoadd="";
            string grados = "" + (char)42;
            textoadd = "";
            if (alarmT1==true)
            {
                textoadd = textoadd + "\n" + "Temp1 Dpto.Tecnico=" +
Convert.ToString(alarmvalueT1 / 100) + grados + "C";
                permisoT1 = false;
            }
            if (alarmT2 == true)
            {
                textoadd = textoadd + "\n" + "Temp2 Dpto.Tecnico=" +
Convert.ToString(alarmvalueT2 / 100) + grados + "C";
                permisoT2 = false;
            }
            if (alarmT3 == true)
            {
                textoadd = textoadd + "\n" + "Temp1 Taller=" +
Convert.ToString(alarmvalueT3 / 100) + grados + "C";
                permisoT3 = false;
            }
            if (alarmT4 == true)
            {
                textoadd = textoadd + "\n" + "Temp2 Taller=" +
Convert.ToString(alarmvalueT4 / 100) + grados + "C";
                permisoT4 = false;
            }
            if (alarmI1 == true)
            {
                textoadd = textoadd + "\n" + "Taller: Presencia de agua";
                permisoI1 = false;
            }
            if (alarmI2 == true)
            {
                textoadd = textoadd + "\n" + "Dpto. Tecnico: Presencia de agua";
                permisoI2 = false;
            }
            if (alarmH1 == true)
            {
                textoadd = textoadd + "\n" + "Taller: Presencia de Humo";
                permisoH1 = false;
            }
            if (alarmH2 == true)
            {
                textoadd = textoadd + "\n" + "Dpto. Tecnico: Presencia de Humo";
                permisoH2 = false;
            }
            if (alarmMinC1 == true)
            {
                textoadd = textoadd + "\n" + "Instalaciones: Subcorriente";
                permisoC1 = false;
            }
            if (alarmMaxC1 == true)
            {
                textoadd = textoadd + "\n" + "Instalaciones: Sobrecorriente";
                permisoC1 = false;
            }
            if (alarmM1 == true & permisoM1==true)
            {
                textoadd = textoadd + "\n" + "Puerta: Abierta";
                permisoM1 = false;
            }
            return textoadd;
        }
```

```
        private void btnAp_Click(object sender, EventArgs e)
        {
            if (srlDAQ.IsOpen==true)
            {
                try
                {
                    string comando = "";
                    comando = "1AP0001";
                    Thread.Sleep(10);
                    srlDAQ.WriteLine(comando);
                    Thread.Sleep(10);
                }
                catch
                {

                }
            }

        }

        private void sendmail(string receiver,string text)
        {
            parametrosMail.parametrosmail pmail = new
parametrosMail.parametrosmail();

            MailMessage msg = new MailMessage();
            msg.To.Add(new MailAddress(receiver));
            msg.From = new MailAddress("javier.prado@updatecom.com.ec","SECURE
SYS");
            msg.Subject = "Alarma SECURE SYS";
            msg.Body = text;

            SmtpClient clienteSmtp = new SmtpClient(pmail.dominio, 26);
            clienteSmtp.Credentials = new
NetworkCredential(pmail.usuario,pmail.clave);

            try
            {
                clienteSmtp.SendAsync(msg, msg.To);
            }
            catch
            {
                MessageBox.Show("No se pudo enviar el mail", "Error Envio Mail",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void srlBluetooth_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            Thread.Sleep(200);
            if (srlBluetooth.IsOpen==true)
            {
                getBluetooth = srlBluetooth.ReadExisting();
                auxBluetooth = getBluetooth;
            }

        }

        private void tmrMagnetic_Tick(object sender, EventArgs e)
        {
            permisoM1 = true;
            tmrMagnetic.Enabled = false;
        }


        private void tmrCheckDAQ_Tick(object sender, EventArgs e)
```

```
        {
               if (srlDAQ.IsOpen==true)
                {
                    checkDAQ = true;
                }
        }

        private void tmrAlarmMessage_Tick(object sender, EventArgs e)
        {
            tmrAlarmMessage.Enabled = false;
            if (alarmH1 == true || alarmH2 == true || alarmI1 == true || alarmI2
== true || alarmMaxC1 == true || alarmMinC1 == true || alarmT1 == true || alarmT2
== true || alarmT3 == true || alarmT4 == true)
            {
                DialogResult aceptar = MessageBox.Show("ALARMA", "PELIGRO",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                if (aceptar == DialogResult.OK)
                {
                      tmrAlarmMessage.Enabled = true;
                }
            }
        }
    }
}
```

FORM ADMINISTRAR:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

//2hF3JNzeS47dxK4R


namespace Control_Domótico
{

    public partial class Administrar : Form
    {

        private MySqlConnection connection;
        const int maxUsers = 11; //Numero de usuarios maximo=10
        string usuario = "";
        bool banderasms = false;
        bool banderamail = false;
char[] caracter = { '!', '@', '#', '$', '%', '^', '&', ')', '(', '-',
'+', '=', '[', ']', '{', '}', '"', ':', ';', '?', '/', '<', '>', '|',
'~', '`', ' '};

        public Administrar(string user)
        {
            InitializeComponent();
```

```
            usuario=user;

        }

        private void Administrar_Load(object sender, EventArgs e)
        {
            string privilegio = "";
            //tooltip

toolTip1.SetToolTip(txtNewUser, "Escriba el nuevo nombre de usuario.\nSe
permiten letras (aA-zZ), números (0-9) y puntos (.)");
toolTip1.SetToolTip(txtNewPass, "Escriba una contraseña.\nSe permiten
letras (aA-zZ), números (0-9) y puntos (.)");
toolTip1.SetToolTip(txtNewPassConfirm, "Reescriba la contraseña.");
toolTip1.SetToolTip(dudPrivilegeLevel, "Seleccione el nivel de
privilegio.");
toolTip1.SetToolTip(cbxDelUser, "Seleccione el nombre del usuario a
eliminar.");
toolTip1.SetToolTip(tbxSmsMessage, "Ingrese el texto de encabezado\nde
los mensajes sms.");
toolTip1.SetToolTip(tbxMailMessage, "Ingrese el texto de encabezado de
los\nmensajes de correo electrónico.");
toolTip1.SetToolTip(ckbxSmsSend, "Seleccione los destinatarios a los
que\ndesea enviar los mensajes de texto.");
toolTip1.SetToolTip(ckbxMailSend, "Seleccione los destinatarios a los que
desea\nenviar los mensajes de correo electrónico.");


            lecturabasededatos();

            //Determinacion Privilegios:
            privilegio=privilege(usuario);

            if (privilegio=="Alto")
            {
                gbxNewUser.Enabled = true;
                gbxDelUser.Enabled = true;
                gbxSmsEnable.Enabled = true;
                gbxMailEnable.Enabled = true;
                gbxBluetoothEnable.Enabled = true;
                tbxSmsMessage.Enabled = true;
                tbxMailMessage.Enabled = true;
                gbxSmsSend.Enabled = true;
                gbxMailSend.Enabled = true;
                gbxConfigTemp.Enabled = true;
                gbxConfigCurrent.Enabled = true;
            }
            else
            {
                if (privilegio=="Bajo")
                {
                    gbxNewUser.Enabled = false;
                    gbxDelUser.Enabled = false;
                    gbxSmsEnable.Enabled = false;
                    gbxMailEnable.Enabled = true;
                    gbxBluetoothEnable.Enabled = false;
                    tbxSmsMessage.Enabled = true;
                    tbxMailMessage.Enabled = true;
                    gbxSmsSend.Enabled = true;
                    gbxMailSend.Enabled = true;
```

```
                    gbxConfigTemp.Enabled = false;
                    gbxConfigCurrent.Enabled = false;
                }
            }

        }

        private void btnExitAdministrar_Click(object sender, EventArgs e)
        {
            string parametro = "";
            string newvalue = "";
DialogResult seleccion= MessageBox.Show("¿Desea guardar los cambios?",
"Guardar Cambios",
MessageBoxButtons.YesNoCancel,MessageBoxIcon.Exclamation);

           if (seleccion == DialogResult.Yes)
           {
               //Guardar datos en base de datos
               //Guardar parametros:
               for (int contador = 1; contador < 10; contador++)
               {
                   if (contador == 1)
                   {
                       parametro = "maxT1";
                       newvalue = Convert.ToString(numConfigT1.Value);
                   }
                   else
                   {
                       if (contador == 2)
                       {
                           parametro = "maxT2";
                          newvalue = Convert.ToString(numConfigT2.Value);
                       }
                       else
                       {
                           if (contador == 3)
                           {
                               parametro = "maxT3";
                          newvalue = Convert.ToString(numConfigT3.Value);
                           }
                           else
                           {
                               if (contador == 4)
                               {
                                   parametro = "maxT4";
                          newvalue = Convert.ToString(numConfigT4.Value);
                               }
                               else
                               {
                                   if (contador == 5)
                                   {
                                       parametro = "maxC1";
                       newvalue = Convert.ToString(numConfigMaxC1.Value);
                                   }
                                   else
                                   {
                                       if (contador == 6)
                                       {
                                           parametro = "minC1";
                       newvalue = Convert.ToString(numConfigMinC1.Value);
```

```
                                    }
                                    else
                                    {
                                        if (contador == 7)
                                        {
                                            parametro = "maxC2";
                        newvalue = Convert.ToString(numConfigMaxC2.Value);
                                        }
                                        else
                                        {
                                            if (contador == 8)
                                            {
                                                parametro = "minC2";
                        newvalue = Convert.ToString(numConfigMinC2.Value);
                                            }
                                            else
                                            {
                                                if (contador == 9)
                                                {
                                                  parametro = "magnetico";
                                if (cbxEnableAlarmMagnetic.Checked == true)
                                                    {
                                                     newvalue = "1";
                                                    }
                                                    else
                                                    {
                                                     newvalue = "0";
                                                    }

                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }

MySqlCommand mySqlwritesensorescommand = new MySqlCommand("UPDATE
`controldomotico`.`sensores` SET `Value` = '" + newvalue + "' WHERE
(`Parameter` = '" + parametro + "');", connection);
                    mySqlwritesensorescommand.ExecuteNonQuery();
                }
                //Guardar conectividad:
                for (int i = 1; i < 4; i++)
                {
                    if (i==1)
                    {
                        parametro = "Sms";
                        if (rbtnSmsEnable.Checked==true)
                        {
                            newvalue = "1";
                        }
                        else
                        {
                            if (rbtnSmsDisable.Checked == true)
                            {
                                newvalue = "0";
                            }
```

```
                    }
                }

                if (i == 2)
                {
                    parametro = "Mail";
                    if (rbtnMailEnable.Checked == true)
                    {
                        newvalue = "1";
                    }
                    else
                    {
                        if (rbtnMailDisable.Checked == true)
                        {
                            newvalue = "0";
                        }
                    }
                }

                if (i == 3)
                {
                    parametro = "Bluetooth";
                    if (rbtnBluetoothEnable.Checked == true)
                    {
                        newvalue = "1";
                    }
                    else
                    {
                        if (rbtnBluetoothDisable.Checked == true)
                        {
                            newvalue = "0";
                        }
                    }
                }

MySqlCommand mySqlwriteconectividadcommand = new MySqlCommand("UPDATE
`controldomotico`.`conectividad` SET `Value` = '" + newvalue + "' WHERE
(`Servicio` = '" + parametro + "');", connection);
                    mySqlwriteconectividadcommand.ExecuteNonQuery();
                }

                //Guardar mensajes:

                newvalue = tbxSmsMessage.Text;
MySqlCommand mySqlwritemessagesmscommand = new MySqlCommand("UPDATE
`controldomotico`.`mensajesmsmail` SET `Message` = '" + newvalue + "'
WHERE (`Service` = 'Sms');", connection);
                mySqlwritemessagesmscommand.ExecuteNonQuery();

                newvalue = tbxMailMessage.Text;
MySqlCommand mySqlwritemessagemailcommand = new MySqlCommand("UPDATE
`controldomotico`.`mensajesmsmail` SET `Message` = '" + newvalue + "'
WHERE (`Service` = 'Mail');", connection);
                mySqlwritemessagemailcommand.ExecuteNonQuery();

                //Guardar checked destinatariossms:
                int numberdestinatarios = 0;
                numberdestinatarios = ckbxSmsSend.Items.Count-1;

                for (int i = 0; i <= numberdestinatarios; i++)
```

```
                {
                    parametro = ckbxSmsSend.Items[i].ToString();
                    newvalue =
Convert.ToString(Convert.ToInt16(ckbxSmsSend.GetItemChecked(i)));

MySqlCommand mySqlwritedestinatariocommand = new MySqlCommand("UPDATE
`controldomotico`.`destinatariossms` SET `Check` = '" +newvalue+ "' WHERE
(`Phone` = '" + parametro + "');", connection);
                    mySqlwritedestinatariocommand.ExecuteNonQuery();

                }

                //Guardar checked destinatariosmail:
                numberdestinatarios = ckbxMailSend.Items.Count - 1;

                for (int i = 0; i <= numberdestinatarios; i++)
                {
                    parametro = ckbxMailSend.Items[i].ToString();
                    newvalue =
Convert.ToString(Convert.ToInt16(ckbxMailSend.GetItemChecked(i)));

MySqlCommand mySqlwritedestinatariocommand = new MySqlCommand("UPDATE
`controldomotico`.`destinatariosmail` SET `Check` = '" + newvalue + "'
WHERE (`Mail` = '" + parametro + "');", connection);
                    mySqlwritedestinatariocommand.ExecuteNonQuery();

                }

                this.Close();

            }
            else
            {
                if (seleccion == DialogResult.No)
                {
                    this.Close();
                }
            }
        }
        private void iniciarbasedatos()
        {
            string server = "localhost";
            string database = "controldomotico";
            string uid = "dsupervisor";
            string password = "2hF3JNzeS47dxK4R";
            string connectionString = "";
            connectionString = "SERVER=" + server + ";" + "DATABASE=" +
database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password;
            connection = new MySqlConnection(connectionString);
        regreso:
            try
            {
                connection.Open();
            }
            catch
            {
DialogResult errorbase = MessageBox.Show("No se pudo abrir la base de
datos", "Error Base de Datos", MessageBoxButtons.RetryCancel);
                if (errorbase == DialogResult.Retry)
                {
```

```
                    goto regreso;
                }
            }
        }

        private void btnNewUser_Click(object sender, EventArgs e)
        {
            errAdministrar.Clear();
            bool banderaLogin = false;
            int indexerror;
            int id;

            if (txtNewUser.Text == "" || txtNewPassConfirm.Text == "" ||
txtNewPass.Text == "")
            {
MessageBox.Show("Complete los campos requeridos", "Nuevo Usuario",
MessageBoxButtons.OK,MessageBoxIcon.Exclamation);
            }
            else
            {
                txtNewUser.Text=txtNewUser.Text.Trim();
                indexerror=txtNewUser.Text.IndexOfAny(caracter);
                if (indexerror != -1)
                {
                    errAdministrar.SetError(txtNewUser, "El nombre
contiene caracteres inválidos.");
                }
                else
                {
                    indexerror = txtNewPass.Text.IndexOfAny(caracter);
                    if (indexerror != -1)
                    {
                        errAdministrar.SetError(txtNewPass, "La
contraseña contiene caracteres inválidos.");
                    }
                    else
                    {
                        if (txtNewPass.Text == txtNewPassConfirm.Text)
                        {
MySqlCommand mySqlreadlogincommand = new MySqlCommand("SELECT
`login`.`User` FROM `controldomotico`.`login` WHERE (`User` = '" +
txtNewUser.Text + "')", connection);
MySqlDataReader readlogin = mySqlreadlogincommand.ExecuteReader();
                            while (readlogin.Read())
                            {
                                if
(txtNewUser.Text==readlogin[0].ToString())
                                {
                                    banderaLogin = true;
                                }
                            }
                            readlogin.Close();

                            id = ultimoid("login");
                            if (id == maxUsers)
                            {
MessageBox.Show("Ha superado el número de usuarios permitidos", "Nuevo
Usuario", MessageBoxButtons.OK, MessageBoxIcon.Information);
                            }
                            else
```

```
                                {
                                    if (banderaLogin == false)
                                    {
MySqlCommand mySqlwritelogincommand = new MySqlCommand("INSERT INTO
`controldomotico`.`login` (`id` ,`User` ,`Password`,`Privilege`) VALUES
(" + id.ToString() + ", '" + txtNewUser.Text + "', '" + txtNewPass.Text +
"','" + dudPrivilegeLevel.Text + "');", connection);

mySqlwritelogincommand.ExecuteNonQuery();
MessageBox.Show("Ingreso Exitoso", "Nuevo Usuario",
MessageBoxButtons.OK);
                                        readusers();
                                    }
                                    else
                                    {
MessageBox.Show("Este nombre de usuario ya existe", "Nuevo Usuario",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                                    }
                                }

                                txtNewUser.Text = "";
                                txtNewPass.Text = "";
                                txtNewPassConfirm.Text = "";
                                dudPrivilegeLevel.Text = "Bajo";
                            }
                            else
                            {
MessageBox.Show("Las contraseñas introducidas no coinciden.", "Error
Contraseña", MessageBoxButtons.OK, MessageBoxIcon.Error);
                                txtNewPass.Text = "";
                                txtNewPassConfirm.Text = "";
                            }
                        }
                    }
                }

            }

            private void readusers()
            {
                int id;
                //lectura base de datos: usuarios
                cbxDelUser.Items.Clear();
                id = ultimoid("login");

                for (int i = 1; i < maxUsers; i++)
                {
MySqlCommand mySqlreadlogincommand = new MySqlCommand("SELECT
`login`.`User` FROM `controldomotico`.`login` WHERE `id` =" +
i.ToString() + " LIMIT 0 , 30", connection);
MySqlDataReader readlogin = mySqlreadlogincommand.ExecuteReader();

                    while (readlogin.Read())
                    {
                        cbxDelUser.Items.Add(readlogin[0].ToString());
                    }
                    readlogin.Close();
                }

            }
```

```
        private string privilege(string user)
        {
            string nivelprivilegio = "";
MySqlCommand mySqlreadprivilegecommand = new MySqlCommand("SELECT
`login`.`Privilege` FROM `controldomotico`.`login` WHERE (`User` =
'"+user+"')LIMIT 0 , 30", connection);
MySqlDataReader readprivilege =
mySqlreadprivilegecommand.ExecuteReader();

            while (readprivilege.Read())
            {
                nivelprivilegio=readprivilege[0].ToString();
            }
            readprivilege.Close();
            return nivelprivilegio;
        }

        private int ultimoid(string table)
        {
            bool banderalogin = false;
            int id=0;
            do
            {
                id++;
                banderalogin=false;
MySqlCommand mySqlreadidcommand = new MySqlCommand("SELECT * FROM
`"+table+"`WHERE `id` =" + id.ToString() + " LIMIT 0 , 30", connection);
MySqlDataReader readid = mySqlreadidcommand.ExecuteReader();
                while (readid.Read())
                  {
                      banderalogin=true;
                  }
                readid.Close();
            } while (banderalogin==true);

            return id;
        }

        private void btnDelUser_Click(object sender, EventArgs e)
        {
            string id="";
            bool banderauser = false;
            if (cbxDelUser.Text == "")
            {
MessageBox.Show("Por favor seleccione el usuario que desea eliminar",
"Eliminar Usuario", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            else
            {
                for (int i = 1; i < maxUsers; i++)
                {
MySqlCommand mySqlreadidcommand = new MySqlCommand("SELECT `login`.`User`
FROM `controldomotico`.`login` WHERE (`id` = '" + i.ToString() + "');",
connection);
MySqlDataReader readid = mySqlreadidcommand.ExecuteReader();

                    while (readid.Read())
                    {
                        if (cbxDelUser.Text==readid[0].ToString())
```

```
                        {
                          id = i.ToString();
                          i = maxUsers;
                          banderauser = true;
                        }

                    }
                    readid.Close();
                }

                if (banderauser==true)
                {
                    DialogResult seleccion = MessageBox.Show("¿Está
seguro que desea borrar el usuario seleccionado?", "Eliminar Usuario",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                    if (seleccion == DialogResult.Yes)
                    {
                        eraseDataBase("login", id);
                        cbxDelUser.Text = "";
                        readusers();
                    }
                }
                else
                {
MessageBox.Show("El destinatario "+cbxDelUser.Text+"no existe","Eliminar
Usuario",MessageBoxButtons.OK,MessageBoxIcon.Error);
                }
            }
        }

        private void eraseDataBase (string table,string field)
        {
MySqlCommand mySqlerasecommand = new MySqlCommand("DELETE FROM
`controldomotico`.`"+table+"` WHERE `"+table+"`.`id` ="+field,
connection);
            mySqlerasecommand.ExecuteNonQuery();
        }

        private void btnEditSmsDest_Click(object sender, EventArgs e)
        {
            banderasms = true;
    EditarDestinatarioSms destinatariosms = new EditarDestinatarioSms();
            destinatariosms.ShowDialog();
        }

        private void btnEditMailDest_Click(object sender, EventArgs e)
        {
            banderamail = true;
Editar_DestinatarioMail destinatariomail = new Editar_DestinatarioMail();
            destinatariomail.ShowDialog();
        }

        private void readdestinatariossms()
        {
            //lectura base de datos: usuarios
            ckbxSmsSend.Items.Clear();

            for (int i = 1; i < 10; i++)
            {
```

```
MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT
`destinatariossms`.`Phone` FROM `controldomotico`.`destinatariossms`
WHERE `id` = " + i.ToString() + " LIMIT 0 , 30", connection);
MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();

                while (readdestinatario.Read())
                {

ckbxSmsSend.Items.Add(readdestinatario[0].ToString());

                }
                readdestinatario.Close();
            }

        }

        private void readdestinatariosmail()
        {
            //lectura base de datos: usuarios
            ckbxMailSend.Items.Clear();

            for (int i = 1; i < 10; i++)
            {
MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT
`destinatariosmail`.`Mail` FROM `controldomotico`.`destinatariosmail`
WHERE `id` = " + i.ToString() + " LIMIT 0 , 30", connection);
MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();

                while (readdestinatario.Read())
                {

ckbxMailSend.Items.Add(readdestinatario[0].ToString());

                }
                readdestinatario.Close();
            }

        }

        private void Administrar_Activated(object sender, EventArgs e)
        {
            if (banderasms==true)
            {
                banderasms = false;
                readdestinatariossms();
                readdestinatarios("destinatariossms", "Sms");
            }
            if (banderamail == true)
            {
                banderamail = false;
                readdestinatariosmail();
                readdestinatarios("destinatariosmail", "Mail");
            }

        }
        private void readdestinatarios(string table,string service)
        {
            string parametro="";
```

```
            string servicio = "";
            int numberdestinatarios = 0;

            if (service=="Sms")
                {
                    numberdestinatarios = ckbxSmsSend.Items.Count - 1;
                    servicio = "Phone";
                }
                else
                {
                    if (service=="Mail")
                    {
                     numberdestinatarios = ckbxMailSend.Items.Count - 1;
                        servicio = "Mail";
                    }
                }
            //lectura parametros:
            for (int i = 0; i <= numberdestinatarios; i++)
            {
                if (service=="Sms")
                {
                    parametro = ckbxSmsSend.Items[i].ToString();
                }
                else
                {
                    if (service=="Mail")
                    {
                        parametro = ckbxMailSend.Items[i].ToString();
                    }
                }
MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT `"+
table +"`.`Check` FROM `controldomotico`.`"+table+"` WHERE
(`"+servicio+"` = '" + parametro + "') LIMIT 0 , 30", connection);
MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();

                while (readdestinatario.Read())
                {
                    if (service=="Sms")
                    {
ckbxSmsSend.SetItemChecked(i,Convert.ToBoolean(readdestinatario[0]));
                    }
                    else
                    {
                        if (service=="Mail")
                        {

ckbxMailSend.SetItemChecked(i,Convert.ToBoolean(readdestinatario[0]));
                        }
                    }

                }
                readdestinatario.Close();
            }

        }

        private void lecturabasededatos()
        {
            //conexion base de datos:
```

```
iniciarbasedatos();

//lectura base de datos:usuarios
readusers();
//lectura base de datos:destinatarios
readdestinatariossms();
readdestinatariosmail();
//inicializacion destinatarios checked:
readdestinatarios("destinatariossms", "Sms");
readdestinatarios("destinatariosmail", "Mail");

//lectura base de datos: parametros de sensores
string parametro = "";

for (int contador = 1; contador < 10; contador++)
{
    if (contador == 1)
    {
        parametro = "maxT1";
    }
    else
    {
        if (contador == 2)
        {
            parametro = "maxT2";
        }
        else
        {
            if (contador == 3)
            {
                parametro = "maxT3";
            }
            else
            {
                if (contador == 4)
                {
                    parametro = "maxT4";
                }
                else
                {
                    if (contador == 5)
                    {
                        parametro = "maxC1";
                    }
                    else
                    {
                        if (contador == 6)
                        {
                            parametro = "minC1";
                        }
                        else
                        {
                            if (contador == 7)
                            {
                                parametro = "maxC2";
                            }
                            else
                            {
                                if (contador == 8)
                                {
```

```
                                                parametro = "minC2";
                                            }
                                            else
                                            {
                                                if (contador == 9)
                                                {
                                               parametro = "magnetico";
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
MySqlCommand mySqlreadsensorescommand = new MySqlCommand("SELECT
`sensores`.`value` FROM `controldomotico`.`sensores` WHERE (`Parameter` =
'" + parametro + "')", connection);
MySqlDataReader readsensores = mySqlreadsensorescommand.ExecuteReader();

                while (readsensores.Read())
                {
                    if (contador == 1)
                    {
        numConfigT1.Value = Convert.ToInt16(readsensores[0].ToString());
                    }
                    else
                    {
                        if (contador == 2)
                        {
        numConfigT2.Value = Convert.ToInt16(readsensores[0].ToString());
                        }
                        else
                        {
                            if (contador == 3)
                            {
        numConfigT3.Value = Convert.ToInt16(readsensores[0].ToString());
                            }
                            else
                            {
                                if (contador == 4)
                                {
         numConfigT4.Value = Convert.ToInt16(readsensores[0].ToString());
                                }
                                else
                                {
                                    if (contador == 5)
                                    {
       numConfigMaxC1.Value = Convert.ToInt16(readsensores[0].ToString());
                                    }
                                    else
                                    {
                                        if (contador == 6)
                                        {
       numConfigMinC1.Value = Convert.ToInt16(readsensores[0].ToString());
                                        }
                                        else
                                        {
                                            if (contador == 7)
```

```
                                                {
        numConfigMaxC2.Value = Convert.ToInt16(readsensores[0].ToString());
                                                }
                                                else
                                                {
                                                    if (contador == 8)
                                                    {
        numConfigMinC2.Value = Convert.ToInt16(readsensores[0].ToString());
                                                    }
                                                    else
                                                    {
                                                        if (contador == 9)
                                                        {

cbxEnableAlarmMagnetic.Checked =
Convert.ToBoolean(Convert.ToInt16(readsensores[0].ToString()));
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }

                        }
                        readsensores.Close();
                    }

                    //lectura base de datos: conectividad
                    for (int i = 1; i < 4; i++)
                    {
                        if (i == 1)
                        {
                            parametro = "Sms";
                        }
                        else
                        {
                            if (i == 2)
                            {
                                parametro = "Mail";
                            }
                            else
                            {
                                if (i == 3)
                                {
                                    parametro = "Bluetooth";
                                }
                            }
                        }
MySqlCommand mySqlreadconectividadcommand = new MySqlCommand("SELECT
`conectividad`.`value` FROM `controldomotico`.`conectividad` WHERE
(`Servicio` = '" + parametro + "')", connection);
MySqlDataReader readconectividad =
mySqlreadconectividadcommand.ExecuteReader();
                        while (readconectividad.Read())
                        {
                            if (i == 1)
                            {
```

```
                        if (readconectividad[0].ToString() == "1")
                        {
                            rbtnSmsEnable.Checked = true;
                        }
                        else
                        {
                            if (readconectividad[0].ToString() == "0")
                            {
                                rbtnSmsDisable.Checked = true;
                            }
                        }
                    }

                    if (i == 2)
                    {
                        if (readconectividad[0].ToString() == "1")
                        {
                            rbtnMailEnable.Checked = true;
                        }
                        else
                        {
                            if (readconectividad[0].ToString() == "0")
                            {
                                rbtnMailDisable.Checked = true;
                            }
                        }
                    }

                    if (i == 3)
                    {
                        if (readconectividad[0].ToString() == "1")
                        {
                            rbtnBluetoothEnable.Checked = true;
                        }
                        else
                        {
                            if (readconectividad[0].ToString() == "0")
                            {
                                rbtnBluetoothDisable.Checked = true;
                            }
                        }
                    }
                }
                readconectividad.Close();
            }
            //lectura mensaje sms:
            MySqlCommand mySqlreadmessagesmscommand = new
MySqlCommand("SELECT `mensajesmsmail`.`message` FROM
`controldomotico`.`mensajesmsmail` WHERE (`Service` = 'Sms')",
connection);
            MySqlDataReader readmessagesms =
mySqlreadmessagesmscommand.ExecuteReader();
            while (readmessagesms.Read())
            {
                tbxSmsMessage.Text = readmessagesms[0].ToString();
            }
            readmessagesms.Close();

            //lectura mensaje mail:
```

```
            MySqlCommand mySqlreadmessagemailcommand = new
MySqlCommand("SELECT `mensajesmsmail`.`message` FROM
`controldomotico`.`mensajesmsmail` WHERE (`Service` = 'Mail')",
connection);
            MySqlDataReader readmessagemail =
mySqlreadmessagemailcommand.ExecuteReader();
            while (readmessagemail.Read())
            {
                tbxMailMessage.Text = readmessagemail[0].ToString();
            }
            readmessagemail.Close();
        }

        private void numConfigT1_ValueChanged(object sender, EventArgs e)
        {
            numConfigT2.Value = numConfigT1.Value;
        }

        private void numConfigT2_ValueChanged(object sender, EventArgs e)
        {
            numConfigT1.Value = numConfigT2.Value;
        }

        private void numConfigT3_ValueChanged(object sender, EventArgs e)
        {
            numConfigT4.Value = numConfigT3.Value;
        }

        private void numConfigT4_ValueChanged(object sender, EventArgs e)
        {
            numConfigT3.Value = numConfigT4.Value;
        }
    }
}
```

FORM DETERMINACIÓN DE PUERTOS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;

namespace Control_Domótico
{
    public partial class dterminacionPuertos : Form
    {
       public string comDaq = "";
       public string comSMS="";
       public string comBluetooth="";
       //public bool allowConfPorts = false;
```

```
        public bool regreso = false;

         public dterminacionPuertos()
         {
             InitializeComponent();
         }

         private void dterminacionPuertos_Load(object sender, EventArgs e)
         {
             string[] ports = SerialPort.GetPortNames();

             foreach (string port in ports)
             {
                 dudSrlDaq.Items.Add(port);
                 dudSrlSMS.Items.Add(port);
                 dudSrlBluetooth.Items.Add(port);
             }
         }

        private void btnConfigPortAccept_Click(object sender, EventArgs e)
         {

             comDaq= dudSrlDaq.Text;
             comSMS= dudSrlSMS.Text;
             comBluetooth= dudSrlBluetooth.Text;

             if (comSMS == "" || comDaq == "" || comBluetooth == "")
             {
MessageBox.Show("Determine todos los puertos", "Error Puertos",
MessageBoxButtons.OK, MessageBoxIcon.Warning);

             }
             else
             {
                 this.Visible = false;
             }
         }
    }
}
```

FORM EDITAR DESTINATATIO MAIL:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Control_Domótico
{
    public partial class Editar_DestinatarioMail : Form
```

```
    {
char[] caracter = { '!', '@', '#', '$', '%', '^', '&', ')', '(', '-',
'+', '=', '[', ']', '{', '}', '"', ':', ';', '?', '/', '<', '>', '|',
'~', '`'};
char[] caracteraddress = { '!', '@', '#', '$', '%', '^', '&', ')', '(',
'-', '+', '=', '[', ']', '{', '}', '"', ':', ';', '?', '/', '<', '>',
'|', '~', '`',' ','A', 'B', 'C', 'D', 'F',
'G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','
Y','Z'};
const int maxnumberdestmail = 7; //Numero maximo de destinatarios mail=6
        private MySqlConnection connection;

        public Editar_DestinatarioMail()
        {
            InitializeComponent();
        }

        private void Editar_DestinatarioMail_Load(object sender,
EventArgs e)
        {
            //ToolTip:
toolTip1.SetToolTip(txtMailNewDestName, "Ingrese el nombre del nuevo
destinatario\npara envio de correo electrónico.\nSe permiten letras (aA-
zZ), números (0-9), puntos (.) y espacios en blanco");
toolTip1.SetToolTip(txtMailNewDestAddressHeader, "Ingrese la dirección de
correo electrónico.\nSe permiten letras (a-z), números (0-9) y puntos
(.)");
toolTip1.SetToolTip(cbxMailDelDestName, "Seleccione el nombre del
destinatario que desea eliminar.");
            toolTip1.SetToolTip(cbxMailDelDestAddress, "Seleccione la
dirección de correo electrónico que desea eliminar.");

            //conexion base de datos:
            iniciarbasedatos();

            //Lectura base de datos: destinatarios
            readdestinatarios();


        }
        private void btnMailNewDest_Click(object sender, EventArgs e)
        {
            int id;
            bool banderadestinatario = false;
            int indexerror;
            int indexerror2;

            errMailDest.Clear();

            if (txtMailNewDestName.Text == "" ||
txtMailNewDestAddressHeader.Text == "" ||
txtMailNewDestAddressTrailer.Text=="")
            {
                MessageBox.Show("Complete los campos requeridos.", "Nuevo
Destinatario Mail", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
            else
            {
                txtMailNewDestName.Text = txtMailNewDestName.Text.Trim();
```

```
                indexerror =
txtMailNewDestName.Text.IndexOfAny(caracter);

                if (indexerror != -1)
                {
errMailDest.SetError(txtMailNewDestName, "El nombre contiene caracteres
especiales.");
                }
                else
                {
                    txtMailNewDestAddressHeader.Text =
txtMailNewDestAddressHeader.Text.Trim();
                    indexerror =
txtMailNewDestAddressHeader.Text.IndexOfAny(caracteraddress);
                    txtMailNewDestAddressTrailer.Text =
txtMailNewDestAddressTrailer.Text.Trim();
                    indexerror2 =
txtMailNewDestAddressTrailer.Text.IndexOfAny(caracteraddress);
                    if (indexerror != -1 || indexerror2 != -1)
                    {

errMailDest.SetError(txtMailNewDestAddressTrailer, "La dirección de
correo electrónico contiene\ncaracteres especiales, espacios en blanco\no
letras en mayúscula.");
                    }
                    else
                    {
MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT
`destinatariosmail`.`Mail` FROM `controldomotico`.`destinatariosmail`
WHERE (`Mail` = '" + txtMailNewDestAddressHeader.Text + "@" +
txtMailNewDestAddressTrailer.Text + "')", connection);
                        MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();
                        while (readdestinatario.Read())
                        {
                            banderadestinatario = true;
                        }
                        readdestinatario.Close();

                        id = ultimoid("destinatariosmail");
                        if (id == maxnumberdestmail)
                        {
MessageBox.Show("Ha excedido el número de destinatarios permitidos.",
"Nuevo Destinatario Mail", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                        }
                        else
                        {

                            if (banderadestinatario == false)
                            {
MySqlCommand mySqlwritedestinatariocommand = new MySqlCommand("INSERT
INTO `controldomotico`.`destinatariosmail` (`id` ,`Name` ,`Mail`) VALUES
(" + id.ToString() + ", '" + txtMailNewDestName.Text + "', '" +
txtMailNewDestAddressHeader.Text + "@" +
txtMailNewDestAddressTrailer.Text + "');", connection);

mySqlwritedestinatariocommand.ExecuteNonQuery();
MessageBox.Show("Ingreso Exitoso.", "Nuevo Destinatario Mail",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
                                readdestinatarios();
```

```
                                }
                                else
                                {
MessageBox.Show("Esta dirección de correo electrónico ya existe.", "Nuevo
Destinatario Mail", MessageBoxButtons.OK, MessageBoxIcon.Hand);
                                }

                                txtMailNewDestName.Text = "";
                                txtMailNewDestAddressHeader.Text = "";
                                txtMailNewDestAddressTrailer.Text = "";
                            }
                        }
                    }
                }
            }

            private int ultimoid(string table)
            {
                bool bandera = false;
                int id=0;
                do
                {
                    id++;
                    bandera=false;
MySqlCommand mySqlreadidcommand = new MySqlCommand("SELECT * FROM
`"+table+"`WHERE `id` =" + id.ToString() + " LIMIT 0 , 30", connection);
                    MySqlDataReader readid =
mySqlreadidcommand.ExecuteReader();
                    while (readid.Read())
                      {
                          bandera=true;
                      }
                    readid.Close();
                } while (bandera==true);

                return id;
            }

                private void readdestinatarios()
               {

                    //lectura base de datos: destinatariossms
                    cbxMailDelDestName.Items.Clear();

                    for (int i = 1; i < maxnumberdestmail; i++)
                    {
MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT
`destinatariosmail`.`Name` FROM `controldomotico`.`destinatariosmail`
WHERE `id` = " + i.ToString() + " LIMIT 0 , 30", connection);
MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();

                        while (readdestinatario.Read())
                        {
                            if
(cbxMailDelDestName.Items.Contains(readdestinatario[0].ToString()) ==
false)
                            {

cbxMailDelDestName.Items.Add(readdestinatario[0].ToString());
```

```
                    }
                }
                readdestinatario.Close();
            }

        }
         private void iniciarbasedatos()
         {
             string server = "localhost";
             string database = "controldomotico";
             string uid = "dsupervisor";
             string password = "2hF3JNzeS47dxK4R";
             string connectionString = "";
connectionString = "SERVER=" + server + ";" + "DATABASE=" + database +
";" + "UID=" + uid + ";" + "PASSWORD=" + password;
             connection = new MySqlConnection(connectionString);
         regreso:
             try
             {
                 connection.Open();
             }
             catch
             {
DialogResult errorbase = MessageBox.Show("No se pudo abrir la base de
datos.", "Error Base de Datos",
MessageBoxButtons.RetryCancel,MessageBoxIcon.Error);
                 if (errorbase == DialogResult.Retry)
                 {
                     goto regreso;
                 }
             }
         }



         private void btnMailDelDest_Click(object sender, EventArgs e)
         {
             string id = "";
             if (cbxMailDelDestName.Text=="")
             {
MessageBox.Show("No ha seleccionado un destinatario.\nPor favor
seleccione el destinatario que desea elminar e intente de
nuevo.","Eliminar Destinatario
Mail",MessageBoxButtons.OK,MessageBoxIcon.Information);
             }
             else
             {
                 if (cbxMailDelDestAddress.Text=="")
                 {
MessageBox.Show("Seleccione la dirección de correo\nelectrónico que desea
eleminar.", "Eliminar Destinatario Mail", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                 }
                 else
                 {
MySqlCommand mySqlreadidcommand = new MySqlCommand("SELECT
`destinatariosmail`.`id` FROM `controldomotico`.`destinatariosmail` WHERE
(`Mail` = '" + cbxMailDelDestAddress.Text + "');", connection);
                     MySqlDataReader readid =
mySqlreadidcommand.ExecuteReader();
```

```
                        while (readid.Read())
                        {
                            id = readid[0].ToString();
                        }
                        readid.Close();
                        if (id != "")
                        {
DialogResult seleccion = MessageBox.Show("¿Está seguro que desea borrar
el destinatario seleccionado?", "Eliminar Destinatario Mail",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                            if (seleccion == DialogResult.Yes)
                            {
                                eraseDataBase("destinatariosmail", id);
                                cbxMailDelDestName.Text = "";
                                cbxMailDelDestAddress.Text = "";
                                cbxMailDelDestAddress.Items.Clear();
                                readdestinatarios();
                            }
                        }
                        else
                        {
MessageBox.Show("Dirección de correo invalida.\n Por favor, seleccione
una de las direcciones en la lista.", "Eliminar Destinatario Mail",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                            cbxMailDelDestAddress.Text = "";
                        }
                    }
                }
            }
            private void eraseDataBase(string table, string field)
            {
                try
                {
MySqlCommand mySqlerasecommand = new MySqlCommand("DELETE FROM
`controldomotico`.`" + table + "` WHERE `" + table + "`.`id` =" + field,
connection);
                    mySqlerasecommand.ExecuteNonQuery();
                }
                catch
                {
MessageBox.Show("No se pudo borrar el destinatario.","Eliminar
Destinatario Mail",MessageBoxButtons.OK,MessageBoxIcon.Exclamation);
                }

            }

private void cbxMailDelDestName_SelectedIndexChanged(object sender,
EventArgs e)
            {
                bool bandera = false;
                cbxMailDelDestAddress.Items.Clear();

                for (int i = 1; i < maxnumberdestmail; i++)
                {
MySqlCommand mySqlreaddestinatarionamecommand = new MySqlCommand("SELECT
`destinatariosmail`.`Name` FROM `controldomotico`.`destinatariosmail`
WHERE (`id` = '" + i.ToString() + "')", connection);
MySqlDataReader readdestinatarioname =
mySqlreaddestinatarionamecommand.ExecuteReader();
```

```
                    while (readdestinatarioname.Read())
                    {
if (cbxMailDelDestName.Text == readdestinatarioname[0].ToString())
                        {
                            bandera = true;
                        }
                    }
                    readdestinatarioname.Close();

                    if (bandera==true)
                    {
                        bandera = false;
MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT
`destinatariosmail`.`Mail` FROM `controldomotico`.`destinatariosmail`
WHERE (`id` = '" + i.ToString() + "')", connection);
MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();

                        while (readdestinatario.Read())
                        {

cbxMailDelDestAddress.Items.Add(readdestinatario[0].ToString());
                        }
                        readdestinatario.Close();
                    }
                }
            }
    }
}
```

FORM EDITAR DESTINATATIO SMS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Control_Domótico
{
    public partial class EditarDestinatarioSms : Form
    {
const int maxsmsdest = 7; //Numero de destinatarios sms permitidos = 6
char[] charactername = { '!', '@', '#', '$', '%', '^', '&', ')', '(', '-',
'+', '=', '[', ']', '{', '}', '"', ':', ';', '?', '/', '<', '>', '|',
'~', '`' };
char[] characternumber = { '!', '@', '#', '$', '%', '^', '&', ')', '(', '-
', '+', '=', '[', ']', '{', '}', '"', ':', ';', '?', '/', '<', '>', '|',
'~', '`', ' ', 'A', 'B', 'C', 'D', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y',
```

```
'Z','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','
r','s','t','u','v','w','y','z'};

        private MySqlConnection connection;

        public EditarDestinatarioSms()
        {
            InitializeComponent();
        }

      private void EditarDestinatarioSms_Load(object sender, EventArgs e)
        {
            //tooltip:
toolTip1.SetToolTip(txtSmsNewDestName, "Ingrese el nombre del nuevo
destinatario\npara envio de mensajes de texto.\nSe permiten letras (aA-
zZ), números (0-9), puntos (.) y espacios en blanco");
toolTip1.SetToolTip(txtSmsNewDestNumber, "Ingrese el número de teléfono
móvil del destinatario.\nSe permiten números (0-9)");
toolTip1.SetToolTip(cbxSmsDelDestName, "Seleccione el destinario Sms que
desea eliminar.");
toolTip1.SetToolTip(cbxSmsDelDestNumber, "Escoja el número de teléfono
que desea\neliminar del destinatario seleccionado.");

            //conexion base de datos:
            iniciarbasedatos();

            //Lectura base de datos: destinatarios
            readdestinatarios();

        }

        private void btnSmsNewDest_Click(object sender, EventArgs e)
        {
            int id;
            bool banderadestinatario = false;
            int indexerrorname;
            int indexerrornumber;

            errorSmsDest.Clear();

      if (txtSmsNewDestName.Text == "" || txtSmsNewDestNumber.Text == "")
            {
MessageBox.Show("Complete los campos requeridos", "Nuevo Destinatario
Sms", MessageBoxButtons.OK,MessageBoxIcon.Exclamation);
            }
            else
            {

MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT
`destinatariossms`.`Phone` FROM `controldomotico`.`destinatariossms`
WHERE (`Phone` ='" + txtSmsNewDestNumber.Text + "')", connection);
                    MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();
                    while (readdestinatario.Read())
                    {
                        banderadestinatario = true;
                    }
                    readdestinatario.Close();

                    id =ultimoid("destinatariossms");
```

```
                    if (id == maxsmsdest)
                    {
MessageBox.Show("Ha excedido el número de destinatarios permitidos.",
"Nuevo Destinatario Sms", MessageBoxButtons.OK,MessageBoxIcon.Hand);
                    }
                    else
                    {

                        if (banderadestinatario == false)
                        {
                            //Validacion del nombre de destinatario:
txtSmsNewDestName.Text = txtSmsNewDestName.Text.Trim();
                            indexerrorname =
txtSmsNewDestName.Text.IndexOfAny(caractername);
                            if (indexerrorname != -1)
                            {

errorSmsDest.SetError(txtSmsNewDestName,"El nombre del destinatario
contiene caracteres especiales");
                            }

//Validacion de los caracteres del numero telefonico:
txtSmsNewDestNumber.Text = txtSmsNewDestNumber.Text.Trim();
indexerrornumber = txtSmsNewDestNumber.Text.IndexOfAny(caracternumber);
                            if (indexerrornumber!= -1)
                            {

errorSmsDest.SetError(txtSmsNewDestNumber, "Existen caracteres
inválidos");
                            }

                           if (indexerrorname==-1 & indexerrornumber==-1)
                            {
//Verificacion de la longitud del numero de telefono:
if (txtSmsNewDestNumber.Text.Length != 9 ||
txtSmsNewDestNumber.Text.StartsWith("0") == false)
                                {

errorSmsDest.SetError(txtSmsNewDestNumber, "El número de teléfono es
inválido");
                                }
                                else
                                {
MySqlCommand mySqlwritedestinatariocommand = new MySqlCommand("INSERT
INTO `controldomotico`.`destinatariossms` (`id` ,`Name` ,`Phone`) VALUES
(" + id.ToString() + ", '" + txtSmsNewDestName.Text + "', '" +
txtSmsNewDestNumber.Text + "');", connection);

mySqlwritedestinatariocommand.ExecuteNonQuery();
MessageBox.Show("Ingreso Exitoso", "Nuevo Destinatario Sms",
MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
                                    readdestinatarios();
                                    txtSmsNewDestName.Text = "";
                                    txtSmsNewDestNumber.Text = "";
                                }
                            }

                        }
                        else
                        {
```

```
MessageBox.Show("Este número telefónico ya existe.", "Nuevo Usuario Sms",
MessageBoxButtons.OK,MessageBoxIcon.Hand);
                        }


                    }
                }

            }

           private int ultimoid(string table)
        {
            bool bandera = false;
            int id=0;
            do
            {
                id++;
                bandera=false;
MySqlCommand mySqlreadidcommand = new MySqlCommand("SELECT * FROM
`"+table+"`WHERE `id` =" + id.ToString() + " LIMIT 0 , 30", connection);
MySqlDataReader readid = mySqlreadidcommand.ExecuteReader();
                while (readid.Read())
                  {
                      bandera=true;
                  }
                readid.Close();
            } while (bandera==true);

            return id;
        }

            private void readdestinatarios()
           {
               //lectura base de datos: destinatariossms
               cbxSmsDelDestName.Items.Clear();

               for (int i = 1; i < maxsmsdest; i++)
               {
MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT
`destinatariossms`.`Name` FROM `controldomotico`.`destinatariossms` WHERE
`id` = " + i.ToString() + " LIMIT 0 , 30", connection);
MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();

                   while (readdestinatario.Read())
                   {
                       if
(cbxSmsDelDestName.Items.Contains(readdestinatario[0].ToString())==false)
                         {

cbxSmsDelDestName.Items.Add(readdestinatario[0].ToString());
                         }

                   }
                   readdestinatario.Close();
               }

           }
            private void iniciarbasedatos()
            {
```

```
                string server = "localhost";
                string database = "controldomotico";
                string uid = "dsupervisor";
                string password = "2hF3JNzeS47dxK4R";
                string connectionString = "";
                connectionString = "SERVER=" + server + ";" + "DATABASE="
+ database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password;
                connection = new MySqlConnection(connectionString);
            regreso:
                try
                {
                    connection.Open();
                }
                catch
                {
DialogResult errorbase = MessageBox.Show("No se pudo abrir la base de
datos.", "Error Base de Datos",
MessageBoxButtons.RetryCancel,MessageBoxIcon.Error);
                    if (errorbase == DialogResult.Retry)
                    {
                        goto regreso;
                    }
                }
            }

private void cbxSmsDelDestName_SelectedIndexChanged(object sender,
EventArgs e)
            {
                bool bandera = false;

                cbxSmsDelDestNumber.Items.Clear();
                cbxSmsDelDestNumber.Text = "";

                for (int i = 1; i < maxsmsdest; i++)
                {
MySqlCommand mySqlreaddestinatarionamecommand = new MySqlCommand("SELECT
`destinatariossms`.`Name` FROM `controldomotico`.`destinatariossms` WHERE
(`id` ='" + i.ToString() + "') LIMIT 0 , 30", connection);
MySqlDataReader readdestinatarioname =
mySqlreaddestinatarionamecommand.ExecuteReader();

                    while (readdestinatarioname.Read())
                    {
                        if
(cbxSmsDelDestName.Text==readdestinatarioname[0].ToString())
                        {
                            bandera = true;
                        }
                    }
                    readdestinatarioname.Close();

                    if (bandera==true)
                    {
                        bandera = false;
MySqlCommand mySqlreaddestinatariocommand = new MySqlCommand("SELECT
`destinatariossms`.`Phone` FROM `controldomotico`.`destinatariossms`
WHERE (`id` ='" + i.ToString()+ "') LIMIT 0 , 30", connection);
MySqlDataReader readdestinatario =
mySqlreaddestinatariocommand.ExecuteReader();
```

```
                        while (readdestinatario.Read())
                        {

cbxSmsDelDestNumber.Items.Add(readdestinatario[0].ToString());
                        }
                        readdestinatario.Close();
                    }
                }
            }

            private void btnSmsDelDest_Click(object sender, EventArgs e)
            {
                string id = "";
                if (cbxSmsDelDestName.Text=="")
                {
MessageBox.Show("No ha seleccionado un destinatario.\nPor favor
seleccione el destinatario que desea elminar y vuelva a intentarlo.",
"Eliminar Destinatario Sms", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                }
                else
                {
                    if (cbxSmsDelDestNumber.Text=="")
                    {
MessageBox.Show("Seleccione el número de teléfono que desea
eliminar.","Eliminar Destinatario Sms", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    }
                    else
                    {
MySqlCommand mySqlreadidcommand = new MySqlCommand("SELECT
`destinatariossms`.`id` FROM `controldomotico`.`destinatariossms` WHERE
(`Phone` ='" + cbxSmsDelDestNumber.Text + "');", connection);
MySqlDataReader readid = mySqlreadidcommand.ExecuteReader();

                        while (readid.Read())
                        {
                            id = readid[0].ToString();
                        }
                        readid.Close();

                        if (id == "")
                        {
MessageBox.Show("Número invalido. Por favor, seleccione uno de los
números en la lista.", "Eliminar Destinatario Sms", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                        }
                        else
                        {
DialogResult seleccion = MessageBox.Show("¿Está seguro que desea borrar
el destinatario seleccionado?", "Eliminar Destinatario Sms",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                            if (seleccion == DialogResult.Yes)
                            {
                                eraseDataBase("destinatariossms", id);
                                cbxSmsDelDestName.Text = "";
                                cbxSmsDelDestNumber.Text = "";
                                cbxSmsDelDestNumber.Items.Clear();
                                readdestinatarios();
                            }
```

```
                        }
                    }
                }

            }
            private void eraseDataBase(string table, string field)
            {
                try
                {
MySqlCommand mySqlerasecommand = new MySqlCommand("DELETE FROM
`controldomotico`.`" + table + "` WHERE `" + table + "`.`id` =" + field,
connection);
                    mySqlerasecommand.ExecuteNonQuery();
                }
                catch
                {
MessageBox.Show("No se pudo eliminar el destinatario seleccionado.\n Por
favor intente de nuevo", "Eliminar Destinatario Sms",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                }
            }
    }
}
```

FORM EDITAR DESTINATATIO SMS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Control_Domótico
{
    public partial class User : Form
    {
        public string logPass;
        public string logUser;
        const int maxusers = 11; //Numero de usuarios maximo=10

        private MySqlConnection connection;

        public User()
        {
            InitializeComponent();

        }

        private void btnAceptuser_Click(object sender, EventArgs e)
        {
            bool banderauser=false;
```

```
            string id="";

            logPass = txtLogPass.Text;
            logUser = txtLogUser.Text;

            for (int i = 1; i < maxusers; i++)
            {
MySqlCommand mySqlreadusercommand = new MySqlCommand("SELECT
`login`.`User` FROM `controldomotico`.`login` WHERE (`id` = '" +
i.ToString() + "')", connection);
MySqlDataReader readuser = mySqlreadusercommand.ExecuteReader();

                while (readuser.Read())
                {
                    if (logUser == readuser[0].ToString())
                    {
                        banderauser = true;
                        id = i.ToString();
                        i = maxusers;
                    }
                }
                readuser.Close();
            }

            if (banderauser==true)
              {
MySqlCommand mySqlreadpasswordcommand = new MySqlCommand("SELECT
`login`.`Password` FROM `controldomotico`.`login` WHERE (`id` = '" + id +
"')", connection);
MySqlDataReader readpassword = mySqlreadpasswordcommand.ExecuteReader();

                while (readpassword.Read())
                {
                    if (logPass == readpassword[0].ToString())
                    {
                        Administrar admin = new Administrar(logUser);
                        this.Hide();
                        admin.ShowDialog();
                        this.Close();
                    }
                    else
                    {
MessageBox.Show("Usuario y/o Contraseña Incorrecta", "Error",
MessageBoxButtons.OK);
                        txtLogUser.Text = "";
                        txtLogPass.Text = "";
                    }
                }
                readpassword.Close();
            }
            else
            {
MessageBox.Show("Usuario y/o Contraseña Incorrecta", "Error",
MessageBoxButtons.OK);
                txtLogUser.Text = "";
                txtLogPass.Text = "";
            }
        }

        private void User_Load(object sender, EventArgs e)
```

```
        {
            iniciarbasedatos();

toolTip1.SetToolTip(txtLogUser, "Escriba el nombre de usuario.");
toolTip1.SetToolTip(txtLogPass, "Ingrese su contraseña.");

        }

        private void btnCancelUser_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void iniciarbasedatos()
        {
            string server = "localhost";
            string database = "controldomotico";
            string uid = "dsupervisor";
            string password = "2hF3JNzeS47dxK4R";
            string connectionString = "";
connectionString = "SERVER=" + server + ";" + "DATABASE=" + database +
";" + "UID=" + uid + ";" + "PASSWORD=" + password;
connection = new MySqlConnection(connectionString);
        regreso:
            try
            {
                connection.Open();
            }
            catch
            {
DialogResult errorbase = MessageBox.Show("No se pudo abrir la base de
datos", "Error Base de Datos", MessageBoxButtons.RetryCancel);
                if (errorbase == DialogResult.Retry)
                {
                    goto regreso;
                }
            }
        }
    }
}
```

CODIGO FUENTE INTERFAZ MÓVIL:

CLASE MIDLET SECURESYS:

```
package SECSYSControl;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import java.util.Vector;
import javax.bluetooth.RemoteDevice;
import org.netbeans.microedition.lcdui.SplashScreen;
```

```
public class SECURESYS extends MIDlet implements CommandListener {

    private boolean pausarMidlet = false;
    //Login de inicio
    private Form loginForm;
    private TextField userField;
    private TextField passwordField;
    private ImageItem imageItem;
    private Image image1;
    private Command login;
    //Splash de inicio:
    protected SplashScreen splashInicio;
    protected Image image;
    //Variables:
    private String temperaturataller = "";
    private String temperaturadeptec = "";
    private String corrientedomotico = "";
    private String inundaciontaller = "";
    private String inundaciondeptec = "";
    private String humotaller = "";
    private String humodeptec = "";
    private String magneticoestado = "";
    protected String url;
    protected boolean conectar = false;
    // declaracion de FORMS:
    protected Form FormSeleccion;
    protected Form FormTemperatura;
    protected Form FormCorriente;
    protected Form FormHumedad;
    protected Form FormInundacion;
    protected Form FormMagnetico;
    protected Form FormHumo;
    public Form FormConectividad;
    //declaracion COMANDOS:
    private Command CommSeleccionar;
    private Command CommSalir;
    private Command CommTempBack;
    private Command CommCorrienteBack;
    private Command CommHumedadBack;
    private Command CommInundacionBack;
    private Command CommMagneticoBack;
    private Command CommHumoBack;
    protected Command CommBuscar;
    protected Command CommParar;
    private Command CommConectividadBack;
    private Command CommConectividad;
    private Command CommAbrirPuerta;
    //declarcion CHOICEGROUP:
    protected ChoiceGroup ChgSensores;
    protected ChoiceGroup ListConectividad;
    //clase para Bluetooth:
    private DispositivosBluetooth dispositivosBluet;
    private EnvioBluetooth envioBluet;
    protected Vector Dispositivos;
    protected String message = "";
    protected String message2 = "";
    protected RemoteDevice otherDevice;

    public SECURESYS() {
```

```
    Dispositivos = new Vector();
}

public void startApp() {
    if (pausarMidlet) {
        resumeMIDlet();
    } else {
        initialize();
        startMIDlet();
    }
    pausarMidlet = false;
}

public void initialize() {

    dispositivosBluet = new DispositivosBluetooth(this);
    envioBluet = new EnvioBluetooth(this);
}

public void startMIDlet() {
    switchDisplayable(null, getSplashScreen());

}

public void pauseApp() {
    pausarMidlet = true;
}

public void resumeMIDlet() {
}

public void destroyApp(boolean unconditional) {
}

public void commandAction(Command command, Displayable displayable) {
    message = "";
    message2 = "";

    if (displayable == splashInicio) {
        if (command == SplashScreen.DISMISS_COMMAND) {

            switchDisplayable(null, getLoginForm());

        }
    } else if (displayable == loginForm) {
        if (command == login) {
                if (userField.getString().trim().equals("Updatecom") &
                passwordField.getString().trim().equals("remote")) {
                switchDisplayable(null, getFormConectividad());

                int err = 0;
                err = dispositivosBluet.initLocalDevice();
                if (err != 0) {
                    return;
                }
                err = dispositivosBluet.startDeviceSearch();
                if (err != 0) {
                    return;
                }
```

```
            FormConectividad.removeCommand(CommBuscar);
            FormConectividad.addCommand(CommParar);

        } else {
            userField.setString("");
            passwordField.setString("");
        }

    } else if (command == CommSalir) {
        exitMIDlet();
    }
} else if (displayable == FormSeleccion) {

    if (command == CommSalir) {
        exitMIDlet();
    } else if (command == CommSeleccionar) {
        if (ChgSensores.getSelectedIndex() == 0) {
            int error = 0;
            error = envioBluet.initLocalDevice();
            if (error == 0) {
                message = "UDCST";
                message2 = "OK";
                envioBluet.runn("Temperatura");
            }
            switchDisplayable(null, getFormTemperatura());
            FormTemperatura.deleteAll();

        } else if (ChgSensores.getSelectedIndex() == 1) {
            int error = 0;
            error = envioBluet.initLocalDevice();
            if (error == 0) {
                message = "UDCSC";
                envioBluet.runn("Corriente");
            }
            switchDisplayable(null, getFormCorriente());
            FormCorriente.deleteAll();
        } else if (ChgSensores.getSelectedIndex() == 2) {
            int error = 0;
            error = envioBluet.initLocalDevice();
            if (error == 0) {
                message = "UDCSI";
                message2 = "OK";
                envioBluet.runn("Inundacion");
            }
            switchDisplayable(null, getFormInundacion());
            FormInundacion.deleteAll();
        } else if (ChgSensores.getSelectedIndex() == 3) {
            int error = 0;
            error = envioBluet.initLocalDevice();
            if (error == 0) {
                message = "UDCSH";
                message2 = "OK";
                envioBluet.runn("Humo");
            }
            switchDisplayable(null, getFormHumo());
            FormHumo.deleteAll();
        } else if (ChgSensores.getSelectedIndex() == 4) {
            int error = 0;
            error = envioBluet.initLocalDevice();
```

```
            if (error == 0) {
                message = "UDCSM";
                envioBluet.runn("Magnetico");
            }
            switchDisplayable(null, getFormMagnetico());
            FormMagnetico.deleteAll();
        }
    } else if (command == CommConectividad) {
        switchDisplayable(null, getFormConectividad());
    }
} else if (displayable == FormTemperatura) {
    if (command == CommTempBack) {
        FormTemperatura.deleteAll();
        switchDisplayable(null, getFormSeleccion());

    }
} else if (displayable == FormCorriente) {
    if (command == CommCorrienteBack) {
        FormCorriente.deleteAll();
        switchDisplayable(null, getFormSeleccion());
    }
} else if (displayable == FormInundacion) {
    if (command == CommInundacionBack) {
        FormInundacion.deleteAll();
        switchDisplayable(null, getFormSeleccion());
    }
} else if (displayable == FormHumo) {
    if (command == CommHumoBack) {
        FormHumo.deleteAll();
        switchDisplayable(null, getFormSeleccion());
    }
} else if (displayable == FormMagnetico) {
    if (command == CommMagneticoBack) {
        FormMagnetico.deleteAll();
        switchDisplayable(null, getFormSeleccion());
    } else if (command == CommAbrirPuerta) {
        int error = 0;
        error = envioBluet.initLocalDevice();
        if (error == 0) {
            message = "1AP0001";
            envioBluet.runn("Puerta");
        }
    }
} else if (displayable == FormConectividad) {
    if (command == CommBuscar) {
        int err = 0;
        err = dispositivosBluet.initLocalDevice();
        if (err != 0) {
            return;
        }
        err = dispositivosBluet.startDeviceSearch();
        if (err != 0) {
            return;
        }
        FormConectividad.removeCommand(CommBuscar);
        FormConectividad.addCommand(CommParar);
    } else if (command == CommParar) {
        dispositivosBluet.stopDiscover();
        FormConectividad.removeCommand(CommParar);
```

```
            FormConectividad.addCommand(CommBuscar);
        } else if (command == CommConectividadBack) {
            FormConectividad.removeCommand(CommParar);
            FormConectividad.addCommand(CommBuscar);
 otherDevice = (RemoteDevice) Dispositivos.elementAt(ListConectividad.getSelectedIndex());
            dispositivosBluet.stopDiscover();
            switchDisplayable(null, getFormSeleccion());

            if (envioBluet.desconectar == true) {
                try {
                    envioBluet.conn.close();

                } catch (Exception e) {
                }
            }
            conectar = true;

        } else if (command == CommSalir) {
            exitMIDlet();
        }
    }
  }

  public Form getFormCorriente() {
    if (FormCorriente == null) {
        FormCorriente = new Form("Sensores de corriente:", new Item[]{});
        FormCorriente.addCommand(getCommCorrienteBack());
//Para que escuche el comando de este form y vaya al metodo commandAction cuando se aprete
un boton:
        FormCorriente.setCommandListener(this);
    }
    return FormCorriente;
  }

  public Form getFormHumedad() {
    if (FormHumedad == null) {
        FormHumedad = new Form("Sensores de humedad:", new Item[]{});
        FormHumedad.addCommand(getCommHumedadBack());
//Para que escuche el comando de este form y vaya al metodo commandAction cuando se aprete
un boton:
        FormHumedad.setCommandListener(this);
    }
    return FormHumedad;
  }

  public Form getFormInundacion() {
    if (FormInundacion == null) {
        FormInundacion = new Form("Sensores Inundacion:", new Item[]{});
        FormInundacion.addCommand(getCommInundacionBack());
 //Para que escuche el comando de este form y vaya al metodo commandAction cuando se aprete
un boton:
        FormInundacion.setCommandListener(this);
    }
    return FormInundacion;
  }

  public Form getFormMagnetico() {
    if (FormMagnetico == null) {
        FormMagnetico = new Form("Puerta:", new Item[]{});
```

```
            FormMagnetico.addCommand(getCommMagneticoBack());
            FormMagnetico.addCommand(getCommAbrirPuerta());
//Para que escuche el comando de este form y vaya al metodo commandAction cuando se aprete
un boton:
            FormMagnetico.setCommandListener(this);
        }
        return FormMagnetico;
    }

    public Form getFormSeleccion() {
        if (FormSeleccion == null) {
            FormSeleccion = new Form("Sensores", new Item[]{getChgSensores()});
            FormSeleccion.addCommand(getCommSalir());
            FormSeleccion.addCommand(getCommConectividad());
            FormSeleccion.addCommand(getCommSeleccionar());
//Para que escuche el comando de este form y vaya al metodo commandAction cuando se aprete
un boton:
            FormSeleccion.setCommandListener(this);
        }
        return FormSeleccion;
    }

    public Form getFormTemperatura() {
        if (FormTemperatura == null) {
            FormTemperatura = new Form("Temperatura");
            FormTemperatura.addCommand(getCommTempBack());
 //Para que escuche el comando de este form y vaya al metodo commandAction cuando se aprete
un boton:
            FormTemperatura.setCommandListener(this);
        }
        return FormTemperatura;
    }

    public Form getFormConectividad() {
        if (FormConectividad == null) {
            FormConectividad = new Form("Conectividad");
            FormConectividad.addCommand(getCommBuscar());
            FormConectividad.addCommand(getCommParar());
            FormConectividad.addCommand(getCommConectividadBack());
            FormConectividad.removeCommand(CommParar);
            FormConectividad.addCommand(CommBuscar);
            FormConectividad.addCommand(getCommSalir());
            FormConectividad.setCommandListener(this);
//Para que escuche el comando de este form y vaya al metodo commandAction cuando se apriete
un boton:
            FormConectividad.append(getListConectividad());

        }
        return FormConectividad;
    }

    public Form getFormHumo() {
        if (FormHumo == null) {
            FormHumo = new Form("Presencia de humo:", new Item[]{});
            FormHumo.addCommand(getCommHumoBack());
 //Para que escuche el comando de este form y vaya al metodo commandAction cuando se apriete
un boton:
            FormHumo.setCommandListener(this);
        }
```

```
    return FormHumo;
}

public Command getCommCorrienteBack() {
    if (CommCorrienteBack == null) {
        CommCorrienteBack = new Command("Regresar", Command.BACK, 0);

    }
    return CommCorrienteBack;
}

public Command getCommHumedadBack() {
    if (CommHumedadBack == null) {
        CommHumedadBack = new Command("Regresar", Command.BACK, 0);
    }
    return CommHumedadBack;
}

public Command getCommHumoBack() {
    if (CommHumoBack == null) {
        CommHumoBack = new Command("Regresar", Command.BACK, 0);
    }
    return CommHumoBack;
}

public Command getCommInundacionBack() {
    if (CommInundacionBack == null) {
        CommInundacionBack = new Command("Regresar", Command.BACK, 0);
    }
    return CommInundacionBack;
}

public Command getCommMagneticoBack() {
    if (CommMagneticoBack == null) {
        CommMagneticoBack = new Command("Regresar", Command.BACK, 0);
    }
    return CommMagneticoBack;
}

public Command getCommSalir() {
    if (CommSalir == null) {
        CommSalir = new Command("Salir", Command.OK, 0);
    }
    return CommSalir;
}

public Command getCommSeleccionar() {
    if (CommSeleccionar == null) {
        CommSeleccionar = new Command("Aceptar", Command.BACK, 0);
    }
    return CommSeleccionar;
}

public Command getCommConectividad() {
    if (CommConectividad == null) {
        CommConectividad = new Command("Conectividad", Command.OK, 0);
    }
    return CommConectividad;
}
```

```
public Command getCommTempBack() {
    if (CommTempBack == null) {
        CommTempBack = new Command("Regresar", Command.BACK, 0);
    }
    return CommTempBack;
}

public Command getCommBuscar() {
    if (CommBuscar == null) {
        CommBuscar = new Command("Actualizar", Command.OK, 1);
    }
    return CommBuscar;
}

public Command getCommConectividadBack() {
    if (CommConectividadBack == null) {
        CommConectividadBack = new Command("Aceptar", Command.BACK, 0);

    }
    return CommConectividadBack;
}

public Command getCommParar() {
    if (CommParar == null) {
        CommParar = new Command("Parar", Command.OK, 1);
    }
    return CommParar;
}

public Command getCommAbrirPuerta() {
    if (CommAbrirPuerta == null) {
        CommAbrirPuerta = new Command("Abrir", Command.OK, 1);
    }
    return CommAbrirPuerta;
}

public Command getLoginCommand() {
    if (login == null) {

        login = new Command("Login", Command.BACK, 1);

    }
    return login;
}

public ChoiceGroup getListConectividad() {
    if (ListConectividad == null) {
        ListConectividad = new ChoiceGroup(null, Choice.EXCLUSIVE);
    }
    return ListConectividad;
}

public ChoiceGroup getChgSensores() {
    if (ChgSensores == null) {
        ChgSensores = new ChoiceGroup("Sensores:", Choice.EXCLUSIVE);
        ChgSensores.append("Temperatura", null);
        ChgSensores.append("Corriente", null);
        ChgSensores.append("Inundacion", null);
        ChgSensores.append("Humo", null);
```

```
            ChgSensores.append("Magnetico", null);
            ChgSensores.setFitPolicy(Choice.TEXT_WRAP_OFF);
            ChgSensores.setSelectedFlags(new boolean[]{false, false, false, false, false});
        }

        return ChgSensores;
    }

    public void switchDisplayable(Alert alert, Displayable nextDisplayable) {

        Display display = getDisplay();
        if (alert == null) {
            display.setCurrent(nextDisplayable);
        } else {
            display.setCurrent(alert, nextDisplayable);
        }
    }

    public Display getDisplay() {
        return Display.getDisplay(this);
    }

    public void exitMIDlet() {
        switchDisplayable(null, null);
        destroyApp(true);
        notifyDestroyed();
        try {
            envioBluet.conn.close();

        } catch (Exception e) {
        }
    }

    public Form getLoginForm() {
        if (loginForm == null) {

            loginForm = new Form("Login", new Item[]{getImageItem(), getuserField(),
getpasswordField()});
            loginForm.addCommand(getLoginCommand());
            loginForm.addCommand(getCommSalir());
            loginForm.setCommandListener(this);

        }
        return loginForm;
    }

    public TextField getuserField() {
        if (userField == null) {

            userField = new TextField("Username", null, 32, TextField.ANY);
            userField.setInitialInputMode("UCB_BASIC_LATIN");

        }
        return userField;
    }

    public TextField getpasswordField() {
        if (passwordField == null) {
```

```
passwordField = new TextField("Password", null, 32, TextField.ANY | TextField.PASSWORD);
passwordField.setInitialInputMode("UCB_BASIC_LATIN");

        }
        return passwordField;
    }

    public ImageItem getImageItem() {
        if (imageItem == null) {

imageItem = new ImageItem("", getImage1(), ImageItem.LAYOUT_CENTER | Item.LAYOUT_TOP
| Item.LAYOUT_BOTTOM | Item.LAYOUT_VCENTER | Item.LAYOUT_SHRINK | Item.LAYOUT_2,
"<Missing Image>", Item.PLAIN);
            imageItem.setPreferredSize(-1, -1);

        }
        return imageItem;
    }

    public Image getImage1() {
        if (image1 == null) {

            try {
                image1 = Image.createImage("/lock.jpg");
            } catch (java.io.IOException e) {
                e.printStackTrace();
            }

        }
        return image1;
    }

    public SplashScreen getSplashScreen() {
        if (splashInicio == null) {
            splashInicio = new SplashScreen(getDisplay());
            splashInicio.setTimeout(2000);
            splashInicio.setAllowTimeoutInterrupt(false);
            splashInicio.setTitle("Bienvenido");
            splashInicio.setCommandListener(this);
            splashInicio.setImage(getImage());
            splashInicio.setFullScreenMode(false);

        }
        return splashInicio;
    }

    public Image getImage() {
        if (image == null) {
            try {
                image = Image.createImage("/icono screen.jpg");
            } catch (java.io.IOException e) {
                e.printStackTrace();
            }
        }
        return image;
    }
}
```

CLASE ENVÍO BLUETOOTH:

```
package SECSYSControl;

import javax.bluetooth.DiscoveryListener;
import javax.bluetooth.LocalDevice;
import javax.bluetooth.RemoteDevice;
import javax.bluetooth.DeviceClass;
import javax.bluetooth.ServiceRecord;
import javax.bluetooth.BluetoothStateException;
import javax.bluetooth.DiscoveryAgent;
import java.io.DataOutputStream;
import java.io.DataInputStream;
import java.io.InputStream;
import java.io.IOException;
import javax.bluetooth.UUID;
import javax.microedition.io.Connector;
import javax.microedition.io.StreamConnection;
import javax.microedition.io.StreamConnectionNotifier;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import java.util.Vector;

public class EnvioBluetooth implements DiscoveryListener {
private SECURESYS domotico;

    private StreamConnectionNotifier con;
    private LocalDevice devLocal;
    private DiscoveryAgent discoverAgent;
    private Display display;
    private ServiceRecord service;
    private UUID uuid;
    DataOutputStream output;
    DataInputStream input;
    private String url;
    protected String saveurl;
    protected int answerint;
    protected int int1;
    protected int int2;
    protected int int3;
    protected double int4;
    protected String answer;
    protected String answer2;
    StreamConnection conn;
    protected Vector inputs;
    protected boolean bandera;
    protected String despliegue;
    protected boolean desconectar=false;
    private boolean abrirpuerta=false;

    protected int initLocalDevice() {
        try {
        devLocal = LocalDevice.getLocalDevice();
        discoverAgent = devLocal.getDiscoveryAgent();
        } catch (BluetoothStateException e) {
        showErrAlert( "Error getting local device!" );
        return -1;
```

```
    }
    return 0;
}

public EnvioBluetooth(SECURESYS domotico){
     this.domotico=domotico;

}

public void deviceDiscovered(RemoteDevice remoteDev, DeviceClass devClass) {

}

public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
    service = servRecord[0];
}

public void serviceSearchCompleted(int transID, int respCode) {
    switch (respCode) {

                case SERVICE_SEARCH_COMPLETED:

                        try {

                  createConnection();

                  //domotico.FormInundacion.append("Paso1");
                  output.writeUTF(domotico.message);
                  //domotico.FormInundacion.append("Paso2");
                  output.flush();
                  output.close();
                  input=conn.openDataInputStream();
                  conn.close();
                  domotico.conectar=true;
                  desconectar=false;
                  if (despliegue!="Puerta") {

                  answerint=input.readInt();
                  answerint=answerint-808464432;
                  int1=answerint*1000;

                  answerint=input.readInt();
                  answerint=answerint-808464432;
                  int2=answerint*100;

                  answerint=input.readInt();
                  answerint=answerint-808464432;
                  int3=answerint*10;

                  answerint=input.readInt();
                  answerint=answerint-808464432;
                  int4=answerint;

                  answer=String.valueOf((int4+int3+int2+int1)/100);

                     if (domotico.message2.equals("OK")) {
                     answerint=input.readInt();
                     answerint=answerint-808464432;
                     int1=answerint*1000;
```

```
                    answerint=input.readInt();
                    answerint=answerint-808464432;
                    int2=answerint*100;

                    answerint=input.readInt();
                    answerint=answerint-808464432;
                    int3=answerint*10;

                    answerint=input.readInt();
                    answerint=answerint-808464432;
                    int4=answerint;

                    answer2=String.valueOf((int4+int3+int2+int1)/100);
                    }
                  desplegarform();
                  input.close();
                 }
                  resetvar();
                  domotico.conectar=true;

                        } catch (Exception e) {
                  error();
                        }
                        break;
                }
    }

    public void inquiryCompleted(int discType) {

    }

    protected void showErrAlert( String strError ) {
        Alert alertErr = new Alert("Error", strError, null, AlertType.ERROR );
        display.setCurrent( alertErr );
    }


    protected void stopDiscover() {

    }

    public void runn(String desplegar) {
        despliegue=desplegar;
        try {
                    uuid = new UUID(0x0003);
                    int transationID = discoverAgent.searchServices(
                            new int[] { 0x0100 }, new UUID[] { uuid }, domotico.otherDevice,this);

        }  catch (BluetoothStateException e) {
                            e.printStackTrace();
            } catch (IOException f) {
                            f.printStackTrace();
            }
    }

    private void createConnection() {
                     try {
url = service.getConnectionURL(ServiceRecord.NOAUTHENTICATE_NOENCRYPT, false);
```

```
                if (domotico.conectar==true) {
                conn = (StreamConnection) Connector.open(url);
                }
                saveurl=url;
                output = conn.openDataOutputStream();
                        domotico.conectar=false;
                desconectar=true;

                 } catch (Exception e) {


                 }
}


 private void error(){
    try{
domotico.FormTemperatura.append("Error en al comunicacion.\nIntente nuevamente.");
domotico.FormCorriente.append("Error en al comunicacion.\nIntente nuevamente.");
domotico.FormHumo.append("Error en al comunicacion.\nIntente nuevamente.");
domotico.FormMagnetico.append("Error en al comunicacion.\nIntente nuevamente.");
domotico.FormInundacion.append("Error en al comunicacion.\nIntente nuevamente.");
    }
    catch(Exception e){
    }

 }
 private void desplegarform(){
    if (despliegue.equals("Temperatura")) {
       domotico.FormTemperatura.deleteAll();
       domotico.FormTemperatura.append("Temp. Taller: "+answer+" °C"+"\n");
       domotico.FormTemperatura.append("Temp. Dpto. Tecn.: "+answer2+" °C"+"\n");

    }else if (despliegue.equals("Corriente")){
        domotico.FormCorriente.deleteAll();
        domotico.FormCorriente.append("Corriente: "+answer+"A"+"\n");

    }else if (despliegue.equals("Inundacion")){
        domotico.FormInundacion.deleteAll();

        if (answer.equals("0.0")) {
           answer="Libre";
        }else if(answer.equals("0.01")){
           answer="Presente";
        }else{
            answer="(no data)";
        }

         if (answer2.equals("0.0")) {
           answer2="Libre";
        }else if(answer.equals("0.01")){
           answer2="Presente";
        }else{
            answer2="(no data)";
        }

        domotico.FormInundacion.append("H2O Taller: "+answer+"\n");
        domotico.FormInundacion.append("H2O Dpto. Tecnico: "+answer2+"\n");
```

```
        }else if (despliegue.equals("Humo")){
            domotico.FormHumo.deleteAll();
            if (answer.equals("0.0")) {
              answer="Libre";
            }else if(answer.equals("0.01")){
              answer="Presente";
            }else{
               answer="(no data)";
            }

             if (answer2.equals("0.0")) {
              answer2="Libre";
            }else if(answer.equals("0.01")){
              answer2="Presente";
            }else{
               answer2="(no data)";
            }

            domotico.FormHumo.append("Humo Taller: "+answer+"\n");
            domotico.FormHumo.append("Humo Dpto. Tecnico: "+answer2+"\n");

        }else if (despliegue.equals("Magnetico")){
            domotico.FormMagnetico.deleteAll();
            if (answer.equals("0.0")) {
              answer="Cerrada";
              abrirpuerta=true;
            }else if(answer.equals("0.01")){
              answer="Abierta";
            }else{
               answer="(no data)";
            }
            domotico.FormMagnetico.append("Puerta: "+answer+"\n");
        }
    }

    protected void resetvar(){
        int1=0;
        int2=0;
        int3=0;
        int4=0;
        try{
            input.close();
        }catch(Exception e){

        }
    }

}
```

CLASE DISPOSITIVOS BLUETOOTH:

```
package SECSYSControl;

//Librerias importadas para bluetooth:

import javax.bluetooth.DiscoveryListener;
import javax.bluetooth.LocalDevice;
```

```
import javax.bluetooth.RemoteDevice;
import javax.bluetooth.DeviceClass;
import javax.bluetooth.ServiceRecord;
import javax.bluetooth.BluetoothStateException;
import javax.bluetooth.DiscoveryAgent;

//Otras librerias:

import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;

public class DispositivosBluetooth implements DiscoveryListener{

    private SECURESYS domotico;

    private LocalDevice devLocal;
    private DiscoveryAgent discoverAgent;
    private Display display;

    public DispositivosBluetooth(SECURESYS domotico){
     this.domotico=domotico;
    }

    protected int initLocalDevice() {
        try {
        devLocal = LocalDevice.getLocalDevice();
        } catch (BluetoothStateException e) {
        showErrAlert( "Error getting local device!" );
        return -1;
        }
        return 0;
    }

    protected int startDeviceSearch() {
        discoverAgent = devLocal.getDiscoveryAgent();

        try {
        domotico.Dispositivos.removeAllElements();
        domotico.ListConectividad.deleteAll();
        discoverAgent.startInquiry( DiscoveryAgent.GIAC,this );
        } catch( BluetoothStateException e ) {
        showErrAlert( "Error starting device enquiry!" );
        return -1;
        }
        return 0;
    }
    public void deviceDiscovered(RemoteDevice remoteDev, DeviceClass devClass) {
       try {
       domotico.Dispositivos.addElement(remoteDev);
        //add device "Friendly name" to a listbox
       domotico.ListConectividad.append( remoteDev.getFriendlyName( false ),
        null);
        } catch (Exception e ) {

       }
    }

    public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
```

```
    }

    public void serviceSearchCompleted(int transID, int respCode) {

    }

    public void inquiryCompleted(int discType) {
        switch (discType) {
              case INQUIRY_TERMINATED:
                 domotico.Dispositivos.removeAllElements();
                 domotico.ListConectividad.deleteAll();
                              break;
                  case INQUIRY_COMPLETED:

                 domotico.FormConectividad.addCommand( domotico.CommBuscar );
                 domotico.FormConectividad.removeCommand( domotico.CommParar);


                  break;

                  default:

                 break;
        }
    }

    protected void showErrAlert( String strError ) {
        Alert alertErr = new Alert("Error", strError, null, AlertType.ERROR );
        display.setCurrent( alertErr );
    }


    protected void stopDiscover() {
        discoverAgent.cancelInquiry( this );
    }
}
```

ANEXO C

1. HOJA DE DATOS DEL SENSOR DE HUMEDAD HS1101

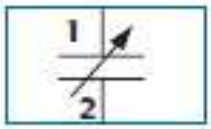
HUMIREL

TECHNICAL DATA

RELATIVE HUMIDITY SENSOR

HS 1100 / HS 1101

Based on a unique capacitive cell, these relative humidity sensors are designed for high volume, cost sensitive applications such as ***office automation, automotive cabin air control, home appliances, and industrial process control systems****. They are also useful in all applications where humidity compensation is needed.*



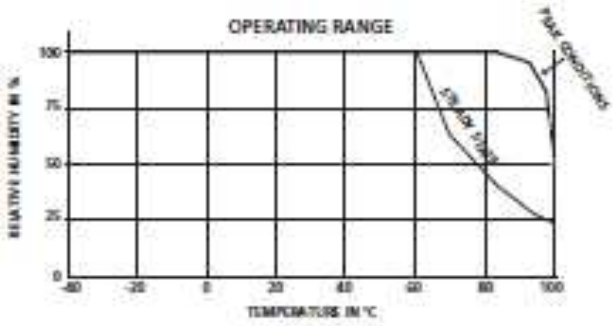
FEATURES

- **Full interchangeability** with no calibration required in standard conditions
- **Instantaneous desaturation** after long periods in saturation phase
- Compatible with automatized assembly processes, **including wave soldering, reflow and water immersion** (1)
- High reliability and long term stability
- Patented solid polymer structure
- Suitable for linear voltage or frequency output circuitry
- Fast response time
- Individual marking for compliance to stringent traceability requirements

(1) soldering temperature profiles available on request

MAXIMUM RATINGS (Ta= 25°C unless otherwise noted)

Ratings	Symbol	Value	Unit
Operating Temperature	Ta	-40 to 100	°C
Storage Temperature	Tstg	-40 to 125	°C
Supply Voltage	Vs	10	Vdc
Humidity Operating Range	RH	0 to 100	% RH
Soldering @ T = 260°C	t	10	s



CHARACTERISTICS

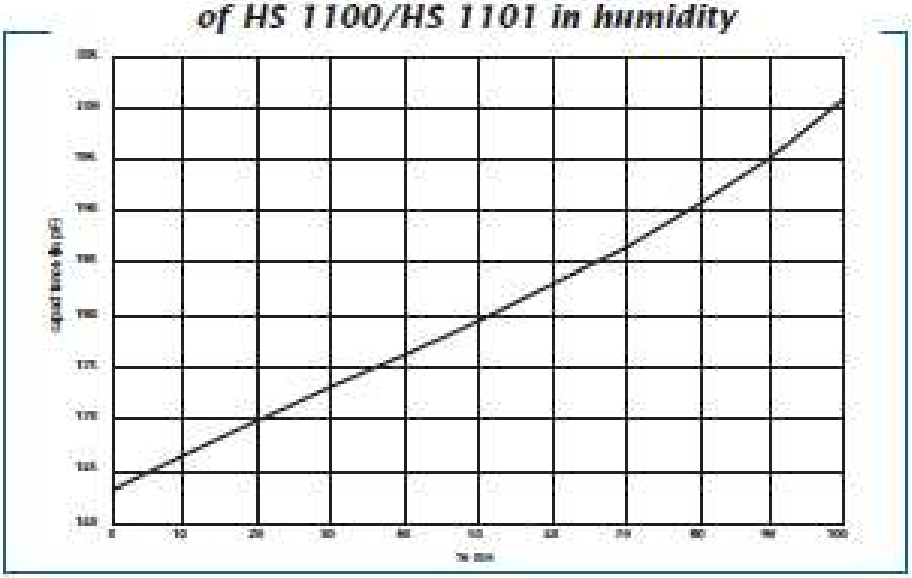
(Ta = 25°C, measurement frequency @ 10kHz unless otherwise noted)

Characteristics	Symbol	Min.	Typ.	Max.	Unit.
Humidity measuring range	RH	1		99	%
Supply voltage	Vs		5	10	V
Nominal capacitance @ 55% RH*	C	177	180	183	pF
Temperature coefficient	Tc		0.04		pF/°C
Averaged Sensitivity from 33% to 75% RH	ΔC/%RH		0.34		pF/%RH
Leakage current (Vcc = 5 Volts)	Ix		1		nA
Recovery time after 150 hours of condensation	tr		10		s
Humidity Hysteresis			+/-1.5		%
Long term stability			0.5		%RH/yr
Response time (33 to 76 % RH, still air @ 63%)	tu		5		s
Deviation to typical response curve (10% to 90% RH)			+/-2		% RH

* Tighter specification available on request

CHARACTERISTICS (CONT'D)

Typical response curve of HS 1100/HS 1101 in humidity



Calibration data are traceable to NIST standards through CETIAT laboratory.

Measurement frequency : 10kHz
Ta = 25°C

Polynomial response : $C(pf)=C@55\%*(1.2*10^{-7}RH^3-1.36*10^{-5}RH^2+2.19*10^{-3}RH+9.010^{-1})$

RH in % RH

Measurement frequency influence

In this data sheet, all capacitance measurements are @ 10kHz. However, the sensor can operate without restriction from 5kHz to 100kHz. To calculate the influence of frequency on capacitance measurements :

$$C@fkHz=C@10kHz(1.027-0.0118*Ln(fkHz))$$

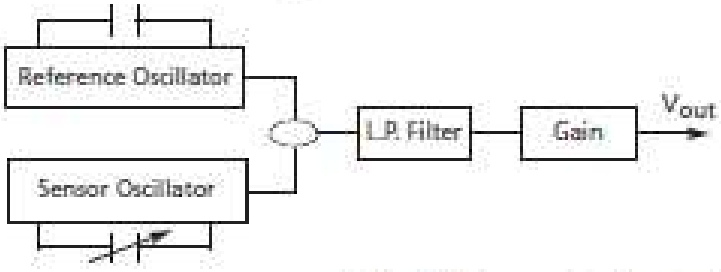
Polarization

In order to get a better reproducibility during measurements, always connect the case of the header (pin 2) to the ground of the circuit.
The case of the header is located on the opposite side of the tab.

Soldering instructions : see the Application Note HPC007

PROPORTIONAL VOLTAGE OUTPUT CIRCUIT

Internal Block Diagram



$$V_{out}=V_{cc}*(0.00474*\%RH+0.234)$$

for 5 - 99% RH

Typical temperature coefficient :
+0.1% RH/°C - From 10 to 60°C

DEMO BOARD AVAILABLE ON REQUEST (REF HMI S10)

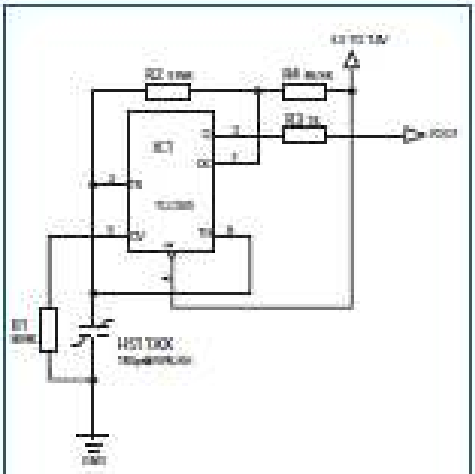
Typical Characteristics for Voltage Output Circuit
At V_{CC} 5V - 25°C

RH	0	10	20	30	40	50	60	70	80	90	100
Voltage (V)	-	1.41	1.65	1.89	2.12	2.36	2.60	2.83	3.07	3.31	3.55

HUMIREL

TECHNICAL 3 DATA

FREQUENCY OUTPUT CIRCUITS



BILL OF MATERIAL AVAILABLE ON REQUEST

COMMENTS

This circuit is the typical astable design for 555. The HS1100/HS1101, used as variable capacitor, is connected to the TRIG and THRES pin. Pin 7 is used as a short circuit pin for resistor R4.
The HS1100/HS1101 equivalent capacitor is charged through R2 and R4 to the threshold voltage (approximately 0.67Vcc) and discharged through R2 only to the trigger level (approximately 0.33Vcc) since R4 is shorten to ground by pin 7.
Since the charge and discharge of the sensor run through different resistors, R2 and R4, the duty cycle is determined by :

$$t_{high} = C@\%RH*(R2+R4)*ln2$$
$$t_{low} = C@\%RH*R2*ln2$$
$$F = 1/(t_{high}+t_{low}) = 1/(C@\%RH*(R4+2*R2)*ln2)$$
$$Output\ duty\ cycle = t_{high}*F = R2/(R4+2*R2)$$

To provide an output duty cycle close to 50%, R4 should be very low compared to R2 but never under a minimum value.
Resistor R3 is a short circuit protection. 555 must be a CMOS version.

REMARK

R1 unbalances the internal temperature compensation schema of the 555 in order to introduce a temperature coefficient that matches the HS1100/HS1101 temperature coefficient. In all cases, R1 should be a 1% resistor with a maximum of 100ppm coefficient temperature like all other R-C timer resistors. Since 555 internal temperature compensation changes from one trademark to one other, R1 value should be adapted to the specific chip. To keep the nominal frequency of 6660Hz at 55%RH, R2 also needs slight adjustment as shown in the table.

555 Type	R1	R2
TLC555 (Texas)	909kΩ	576kΩ
TS555 (SGS)	100nF capacitor	523kΩ
7555 (Harris)	1730kΩ	549kΩ
LMC555 (National)	1238kΩ	562kΩ

For a frequency of 6660Hz at 55%RH

Typical Characteristics for Frequency Output Circuits

REFERENCE POINT AT 6660Hz FOR 55%RH / 25°C

RH	0	10	20	30	40	50	60	70	80	90	100
Frequency	7351	7224	7100	6976	6853	6728	6600	6466	6330	6186	6033

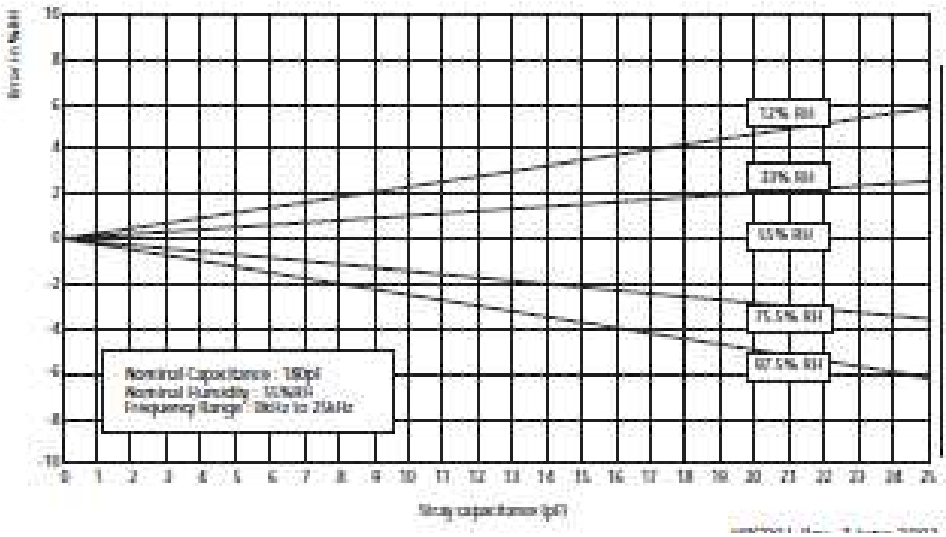
Typical for a 555 Cmos type. TLC555 (RH : Relative Humidity in %, F : Frequency in Hz)

Polynomial response :

$$F_{mes(Hz)} = F55_{(Hz)}(1.1038-1.936810^{-3}*RH+3.011410^{-6}*RH^2-3.440310^{-8}*RH^3)$$

Measurement Error vs Stray Capacitance

A special attention is required in order to minimize stray capacitance in the layout. The added capacitance will act as a parallel capacitance with the sensor and create a measurement error.



2. HOJA DE DATOS DEL SENSOR DE TEMPERATURA DS18S20/18B20

DALLAS MAXIM

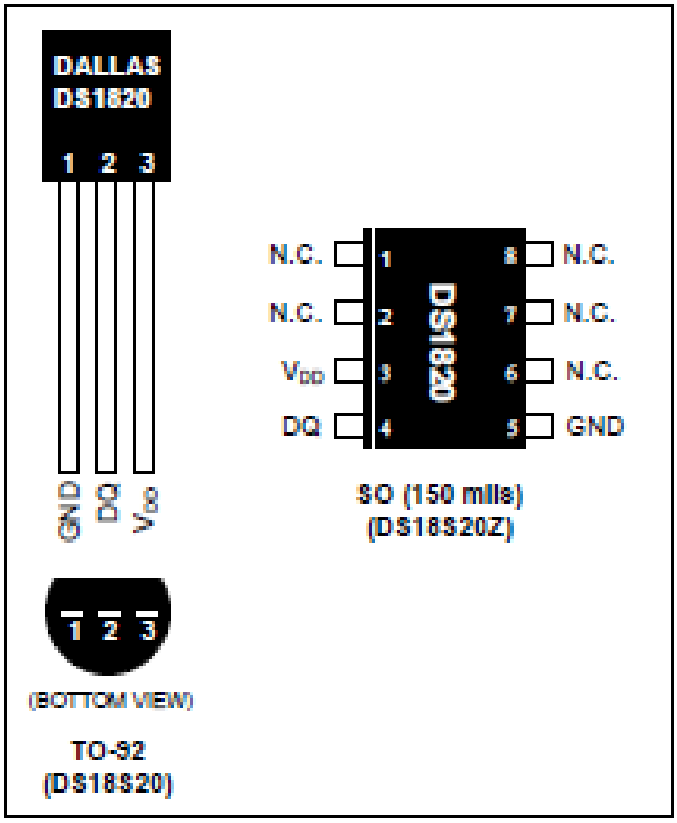
DS18S20
High-Precision 1-Wire Digital Thermometer

www.maxim-ic.com

FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature Sensing Applications
- Requires No External Components
- Can Be Powered From Data Line. Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- 9-Bit Thermometer Resolution
- Converts Temperature in 750ms (max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS



DESCRIPTION

The DS18S20 digital thermometer provides 9-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of –55°C to +125°C and is accurate to ±0.5°C over the range of –10°C to +85°C. In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground.....-0.5V to +6.0V
Operating Temperature Range.....-55°C to +125°C
Storage Temperature Range .....-55°C to +125°C
Solder Temperature .....Refer to the IPC/JEDEC J-STD-020 Specification.
Reflow Oven Temperature .....+220°C

These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD}	Local Power	+3.0		+5.5	V	1
Pullup Supply Voltage	V_{PU}	Parasite Power	+3.0		+5.5	V	1, 2
		Local Power	+3.0		V_{DD}		
Thermometer Error	t_{ERR}	-10°C to +85°C			±0.5	°C	3
		-55°C to +125°C			±2		
Input Logic-Low	V_{IL}		-0.3		+0.8	V	1, 4, 5
Input Logic-High	V_{IH}	Local Power	+2.2		The lower of 5.5 or V_{DD} + 0.3	V	1, 6
		Parasite Power	+3.0				
Sink Current	I_L	V_{IO} = 0.4V	4.0			mA	1
Standby Current	I_{DDS}			750	1000	nA	7, 8
Active Current	I_{DD}	V_{DD} = 5V		1	1.5	mA	9
DQ Input Current	I_{DQ}			5		µA	10
Drift				±0.2		°C	11

NOTES:

1) All voltages are referenced to ground.
2) The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU}. In order to meet the V_{IH} spec of the DS18S20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$.
3) See typical performance curve in Figure 16.
4) Logic-low voltages are specified at a sink current of 4mA.
5) To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
6) Logic-high voltages are specified at a source current of 1mA.
7) Standby current specified up to +70°C. Standby current typically is 3µA at +125°C.
8) To minimize I_{DDS}, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or V_{DD} – 0.3V ≤ DQ ≤ V_{DD}.
9) Active current refers to supply current during active temperature conversions or EEPROM writes.
10) DQ line is high ("high-Z" state).
11) Drift data is based on a 1000-hour stress test at +125°C with V_{DD} = 5.5V.

3. HOJA DE DATOS MAX487

19-1500; Rev 0; 4/00

MAXIM

+3V/+5V, 250MHz, SOT23 ADC Buffer Amplifiers with High-Speed Disable

General Description

The MAX4285/MAX4286 single and MAX4287/MAX4288/MAX4387/MAX4388 dual ADC buffer amplifiers feature high-speed performance and single +3V supply operation. The MAX4285/MAX4286/MAX4288 and MAX4388 offer a disable feature that reduces power-supply current and places the outputs in a high-impedance state. All six devices operate from a +2.85V to +6.5V single supply or from ±1.425V to ±3.25V dual supplies. The common-mode input voltage range extends to the negative power-supply rail (ground in single-supply applications).

These devices require 20mA of quiescent supply current per amplifier while achieving a 250MHz -3dB bandwidth and a 350V/µs slew rate. The combination of an 6ns (to 0.1%) settling time, 88dBc (f = 5MHz) of SFDR, and up to 100mA output drive makes these amplifiers ideal for high-speed ADC drivers for communications and instrumentation applications. In addition, when disabled, their high output impedance makes them ideal for multiplexing applications.

The MAX4285/MAX4286 are available in space-saving 6-pin SOT23 and 8-pin SO packages. The MAX4287/MAX4387 come in 8-pin µMAX and 8-pin SO packages, while the MAX4288/MAX4388 come in 10-pin µMAX and 14-pin SO packages.

Applications

High-Speed ADC Drivers
Communications Equipment
Instrumentation
CCD Imaging Systems
Ultrasound

Typical Operating Circuit appears at end of data sheet.

Pin Configurations appear at end of data sheet.

Features

- High Speed at 3V
  - 250MHz -3dB Bandwidth (MAX4285/87/88)
  - 150MHz -3dB Bandwidth (MAX4286, MAX4387/88)
  - 350V/µs Slew Rate
- +2.85V to +6.5V Single-Supply Operation
- Input Common-Mode Range Extends to V_{EE}
- Low Distortion at 5MHz
  - -88dBc SFDR
- High Output Current Drive: -106mA to +77mA
- 6ns Settling Time to 0.1%
- High-Speed Enable/Disable
  - 40ns Enable Time
  - 50ns Disable Time
  - High Output Impedance
- Space-Saving SOT23 and µMAX Packages

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE	SOT TOP MARK
MAX4285EUT-T	-40°C to +85°C	6 SOT23-6	AABQ
MAX4285ESA	-40°C to +85°C	8 SO	—
MAX4286EUT-T	-40°C to +85°C	6 SOT23-6	AABR
MAX4286ESA	-40°C to +85°C	8 SO	—
MAX4287EUA	-40°C to +85°C	8 µMAX	—
MAX4287ESA	-40°C to +85°C	8 SO	—

Ordering Information continued at end of data sheet.

Selector Guide

PART	OP AMPS PER PKG	MIN GAIN	-3dB BANDWIDTH (AT MIN GAIN)	HIGH-SPEED DISABLE	PIN-PACKAGE
MAX4285	1	1	250MHz	Yes	6-pin SOT23, 8-pin SO
MAX4286	1	5	150MHz	Yes	6-pin SOT23, 8-pin SO
MAX4287	2	1	250MHz	No	8-pin µMAX/SO
MAX4288	2	1	250MHz	Yes	10-pin µMAX/14-pin SO
MAX4387	2	5	150MHz	No	8-pin µMAX/SO
MAX4388	2	5	150MHz	Yes	10-pin µMAX, 14-pin SO

For free samples and the latest literature, visit www.maxim-ic.com or phone 1-800-998-8800. For small orders, phone 1-800-835-8769

+3V/+5V, 250MHz, SOT23 ADC Buffer Amplifiers with High-Speed Disable

ABSOLUTE MAXIMUM RATINGS

Supply Voltage (V_{CC} - V_{EE}) ..... -0.3V to +7.5V
Input Voltage Range (IN+, IN-) ..... (V_{EE} - 0.3V) to (V_{CC} + 0.3V)
Differential Input Voltage ..... -0.3V to +7.5V
Voltage at $\overline{DISABLE}$ ..... (V_{EE} - 0.3V) to (V_{CC} + 0.3V)
Current into IN+, IN-, $\overline{DISABLE}$ ..... ±20mA
Output Short-Circuit Duration ..... Indefinite
Continuous Power Dissipation (T_A = +70°C)
6-Pin SOT23 (derate 7.1mW/°C above +70°C) 571mW
8-Pin µMAX (derate 4.1mW/°C above +70°C) 330mW
10-Pin µMAX (derate 5.6mW/°C above +70°C) 444mW
8-Pin SO (derate 5.88mW/°C above +70°C) 471mW
14-Pin SO (derate 8.3mW/°C above +70°C) 667mW
Operating Temperature Range ..... -40°C to +85°C
Junction Temperature ..... +150°C
Storage Temperature Range ..... -65°C to +150°C
Lead Temperature (soldering, 10s) ..... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = +3V, V_{EE} = 0, $\overline{DISABLE\_}$ = 3V, R_L = ∞, V_{CM} = 1V, and T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Typical values are at T_A = +25°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Operating Supply Voltage Range		Guaranteed by PSRR test		2.85		6.5	V
Input Common-Mode Voltage Range	V_{CM}	Guaranteed by CMRR test		V_{EE}		V_{CC} - 1.25	V
Input Offset Voltage	V_{OS}	MAX4_87EU_/MAX4_88EU_/MAX4_8_ES_			±0.1	±8	mV
		MAX4285EUT-T/MAX4286EUT-T			±1.5	±12	
Input Offset-Voltage Temperature Coefficient	TC_{VOS}				20		µV/°C
Input Offset-Voltage Matching	ΔV_{OS}	MAX4287/88 and MAX4387/88			±0.2		mV
Input Bias Current	I_B				13	35	µA
Input Offset Current	I_{OS}				0.2	8	µA
Input Resistance	R_{IN}	Differential (-10mV ≤ V_{IN} ≤ +10mV)			38		kΩ
		Common mode (V_{EE} ≤ V_{CM} ≤ V_{CC} - 1.25V)			600		
Common-Mode Rejection Ratio	CMRR	V_{EE} ≤ V_{CM} ≤ V_{CC} - 1.25V	MAX4_87EU_/ MAX4_88EU_/ MAX4_8_ES_		50	73	dB
			MAX4285EUT-T/ MAX4286EUT-T		45	68	
Open-Loop Gain	A_{VOL}	VEE + 0.4V ≤ V_{OUT} ≤ V_{CC} - 0.4V	R_L = 2kΩ	75	94		dB
			R_L = 300Ω	65	94		
			R_L = 100Ω	60	85		

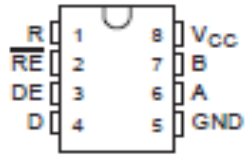
4. HOJA DE DATOS SN75176

SN75176A
DIFFERENTIAL BUS TRANSCEIVER

SLLS100A – JUNE 1984 – REVISED MAY 1995

- Bidirectional Transceiver
- Meets or Exceeds the Requirements of ANSI Standards EIA/TIA-422-B and ITU Recommendation V.11
- Designed for Multipoint Transmission on Long Bus Lines in Noisy Environments
- 3-State Driver and Receiver Outputs
- Individual Driver and Receiver Enables
- Wide Positive and Negative Input/Output Bus Voltage Ranges
- Driver Output Capability . . . ±60 mA Max
- Thermal-Shutdown Protection
- Driver Positive- and Negative-Current Limiting
- Receiver Input Impedance . . . 12 kΩ Min
- Receiver Input Sensitivity . . . ±200 mV
- Receiver Input Hysteresis . . . 50 mV Typ
- Operates From Single 5-V Supply
- Low Power Requirements

D OR P PACKAGE
(TOP VIEW)



description

The SN75176A differential bus transceiver is a monolithic integrated circuit designed for bidirectional data communication on multipoint bus-transmission lines. It is designed for balanced transmission lines and meets ANSI Standard EIA/TIA-422-B and ITU Recommendation V.11.

The SN75176A combines a 3-state differential line driver and a differential input line receiver, both of which operate from a single 5-V power supply. The driver and receiver have active-high and active-low enables, respectively, that can be externally connected together to function as a direction control. The driver differential outputs and the receiver differential inputs are connected internally to form differential input/output (I/O) bus ports that are designed to offer minimum loading to the bus whenever the driver is disabled or V_{CC} = 0. These ports feature wide positive and negative common-mode voltage ranges making the device suitable for party-line applications.

The driver is designed to handle loads up to 60 mA of sink or source current. The driver features positive- and negative-current limiting and thermal shutdown for protection from line fault conditions. Thermal shutdown is designed to occur at a junction temperature of approximately 150°C. The receiver features a minimum input impedance of 12 kΩ, an input sensitivity of ±200 mV, and a typical input hysteresis of 50 mV.

The SN75176A can be used in transmission-line applications employing the SN75172 and SN75174 quadruple differential line drivers and SN75173 and SN75175 quadruple differential line receivers.

The SN75176A is characterized for operation from 0°C to 70°C.

SN75176A
DIFFERENTIAL BUS TRANSCEIVER

SLLS100A – JUNE 1984 – REVISED MAY 1995

recommended operating conditions

		MIN	TYP	MAX	UNIT
Supply voltage, V_{CC}		4.75	5	5.25	V
Voltage at any bus terminal (separately or common mode), V_I or V_{IC}		−7		12	V
High-level input voltage, V_{IH}	D, DE, and $\overline{RE}$	2			V
Low-level input voltage, V_{IL}	D, DE, and $\overline{RE}$			0.8	V
Differential input voltage, V_{ID} (see Note 2)				±12	V
High-level output current, I_{OH}	Driver			−60	mA
	Receiver			−400	µA
Low-level output current, I_{OL}	Driver			60	mA
	Receiver			8	
Operating free-air temperature, T_A		0		70	°C

NOTE 2: Differential-input/output bus voltage is measured at the noninverting terminal A with respect to the inverting terminal B.

Function Tables

DRIVER

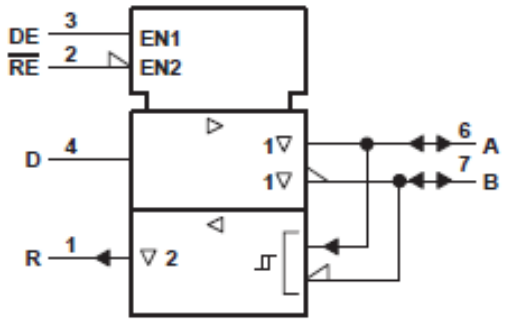
INPUT D	ENABLE DE	OUTPUTS A	OUTPUTS B
H	H	H	L
L	H	L	H
X	L	Z	Z

RECEIVER

DIFFERENTIAL INPUTS A – B	ENABLE $\overline{RE}$	OUTPUT R
$V_{ID} \geq 0.2$ V	L	H
$-0.2\text{ V} < V_{ID} < 0.2$ V	L	?
$V_{ID} \leq -0.2$ V	L	L
X	H	Z
Open	L	?

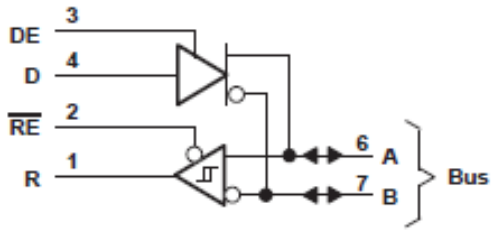
H = high level, L = low level, ? = indeterminate,
X = irrelevant, Z = high impedance (off)

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

logic diagram (positive logic)



5. HOJA DE DATOS SENSOR DE CORRIENTE AKR 50 B420L

AC Current transducer AKR-B420L

I_{PN} = 2..200A

A Split Core transducer for the electronic measurement AC waveforms current, with galvanic isolation between the primary (High power) and the secondary circuits (Electronic circuit). Jumper selectable ranges and True RMS 4-20mA current output.

Electrical data

Primary Nominal Current I_{PN} (A.t.RMS)	Analogue Output Signal[1] I_{OUT} (mA)	Type	RoHS Date Code
2,5	4-20	AKR 6 B420L	JUNE 2006
10,20,50	4-20	AKR 60 B420L	planned
100,150,200	4-20	AKR 200 B420L	planned

Vc	Supply voltage (Loop powered)	24	V DC
R_L	Load resistance	see power supply diagram	
V_e	Rated voltage (CAT III, PD2)	150	V AC
V_d	RMS isolation voltage test; 50 Hz, 1mn	3	kV AC
f	Frequency bandwith	10-400	Hz

Accuracy - Dynamic performance data

x	Accuracy @ I_{PN}, T_A=25°C	± 1	%
t_r	Response time @ 90% of I_{PN}	< 500	mS

General data

T_A	Ambient operating temperature (0-95% RH)	- 20 ..+ 50	°C
T_S	Ambient storage temperature	- 20 ..+ 85	°C
m	Mass	120	g
	Safety	IEC 61010-1	
	EMC	EN 61326	

Note: [1] For 4-20mA output model, no saturation output up to 23 mA.

Selecting the transducer

VFD (Variable Frequency Drive) and SCR (Semi Conductor Rectifier) output waveforms are rough approximations of a sine wave. There are numerous spikes and dips in each cycle. AKR transducers use a mathematical algorithm called "True RMS," which integrates the actual waveform over time. True RMS is the only way to accurately measure distorted AC waveforms. *Select AKR transducers for nonlinear loads or in "noisy" power environments.*

Features

- VFD and SCR waveforms current measurement
- True RMS responding
- Split core box
- Current output
- Loop powered transducers
- Panel mounting
- Jumper selectable ranges

Advantages

- Large aperture
- High isolation between primary and secondary circuits
- Easy to mount

Applications

- VFD Controlled Loads: VFD output indicates how the motor and attached load are operating.
- SCR Controlled Loads: Acurate measurement of phase angle fired or burst fired (time proportioned) SCRs.
- Switching Power Supplies and Electronic Ballasts: True RMS sensing is the most accurate way to measure power supply or ballast input power.

Options on request

- DIN mounting

LEM reserves the right to carry out modifications on its transducers, in order to improve them, without previous notice.

ANEXO D

TIPOS DE ALERTA [60]

- Alerta Verde: Situaciones previsibles que conllevan a riesgo de emergencia colectiva y deben de ser de conocimiento del equipo de urgencias. Ejemplos manifestaciones, concentraciones, paro laboral, demoliciones, etc.

- Alerta Amarilla: Comunicación oficial de que está ocurriendo un evento y es inminente la producción de víctimas. Se activa el comité de emergencias y se pone en práctica el plan de desastres.

- Alerta Roja: Las catástrofes locales o situaciones graves que imprevistamente afecten la seguridad o la vida de gran número de personas o dañen masivamente edificaciones o instalaciones, interrumpiendo la normal actividad de la colectividad y el funcionamiento de las instituciones. Ejemplos terremotos, fenómenos volcánicos, inundaciones, incendios, grave alteración de orden público, enfermedades masivas, etc.