

UNIVERSIDAD TECNOLÓGICA ISRAEL



CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES

**ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PROTOTIPO
DE ALARMA BARRIAL Y SISTEMA DE GRABACIÓN ACTIVADO POR
SMS.**

**TRABAJO DE GRADUACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

AUTOR:

JUAN ROBERTO SANDOVAL PERUGACHI

TUTOR:

ING. MAURICIO ALMINATE

QUITO ECUADOR.

OCTUBRE 2013

UNIVERSIDAD TECNOLÓGICA ISRAEL**APROBACIÓN DEL TUTOR**

En mi calidad de Tutor del Trabajo de Graduación certifico:

Que el trabajo de graduación **“ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PROTOTIPO DE ALARMA BARRIAL Y SISTEMA DE GRABACIÓN ACTIVADO POR SMS.”**, presentado por el Sr. Juan Roberto Sandoval Perugachi, estudiante de la Carrera de Electrónica y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D. M., Octubre de 2013

TUTOR

Ing. Mauricio Alminati V.

UNIVERSIDAD TECNOLÓGICA ISRAEL**AUTORÍA DE TESIS**

El abajo firmante, en calidad de estudiante de la Carrera de Electrónica y Telecomunicaciones, declaro que los contenidos de este Trabajo de Graduación, requisito previo a la obtención del Grado de Ingeniería en Electrónica y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito D.M., Octubre de 2013

Juan Roberto Sandoval Perugachi

CC: 171927064-5

UNIVERSIDAD TECNOLÓGICA ISRAEL
APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado, aprueban la tesis de graduación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

Quito D.M., Octubre de 2013

Para constancia firman:

TRIBUNAL DE GRADO

PRESIDENTE

MIEMBRO 1

MIEMBRO 2

AGRADECIMIENTO

Agradezco a mis padres Antonio y Pastora quienes me supieron apoyar incondicionalmente en mis estudios y han hecho un gran sacrificio para que no me falte nada y pueda estudiar sin preocupaciones, especialmente agradezco a mi madre quien siempre estuvo a mi lado para brindarme su apoyo. Agradezco también a mis hermanas por ser una parte importante de mi vida y representar la unión familiar.

Gracias a todos.

DEDICATORIA

Dedico este proyecto de grado a mis padres, guías primordiales de mi vida. Sin la ayuda de ellos, no hubiese podido conseguir este sueño tan anhelado en mi vida. También dedico este proyecto de grado a mi novia, Deysi, compañera inseparable de mis días, ella representó gran esfuerzo y constancia en momentos de decline y cansancio.

ÍNDICE GENERAL

CAPÍTULO 1 PROBLEMATIZACIÓN

1.1 Antecedentes.....	1
1.2 Problema investigado	2
1.3 Problema principal.....	3
1.4 Problemas secundarios.....	3
1.5 Justificación.....	3
1.6 Objetivos.....	3
1.6.1 Objetivo principal.....	3
1.6.2 Objetivos secundarios.....	4
1.7 Metodología.....	4

CAPÍTULO 2 MARCO DE REFERENCIA

2.1 Marco teórico.....	5
2.1.1 Tecnología GSM.....	5
2.1.1.1 Arquitectura de la red GSM.....	5
2.1.1.2 Características técnicas.....	7
2.1.2 Servicio de mensajes cortos de texto SMS.....	7
2.1.2.1 Formato del SMS.....	8
2.1.2.2 Arquitectura del SMS.....	9
2.1.3 Módulo GSM.....	9
2.1.3.1 Módulo GSM-USB.....	10
2.1.3.2 Módulo GSM RS-232.....	10
2.1.4 Comandos AT.....	11
2.1.4.1 Estructura de los comandos AT.....	11
2.1.4.2 Comandos AT generales.....	12
2.1.5 Protocolo SIRC SONY.....	12
2.1.5.1 Codificación.....	13
2.1.6 Software libre.....	14
2.1.6.1 Hardware libre.....	14
2.2 Marco conceptual.....	15
2.2.1 Especificaciones del módulo GSM.....	15

2.2.2 Comandos AT utilizados para la configuración del modem GSM.....	16
2.2.3 Software Arduino.....	17
2.2.3.1 Entorno Arduino.....	17
2.2.4 Especificaciones de la sirena.....	18
2.2.5 Especificaciones del motor.....	19
2.2.6 Especificaciones del emisor IR.....	19
2.2.7 Especificaciones de la batería.....	20
CAPÍTULO 3 ESTUDIO, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL PROTOTIPO DE ARMA BARRIAL Y SISTEMA DE GRABACIÓN DE VIDEO ACTIVADO POR SMS	
3.1 Estudio de la tecnología Arduino para ser aplicada en Alarmas Barriales.....	21
3.1.1 Arduino.....	21
3.1.2 Hardware Arduino.....	21
3.1.3 Placa electrónica Arduino Uno.....	22
3.2 Diseño del sistema de control prototipo de Alarma Barrial.....	23
3.2.1 Diseño del Hardware.....	25
3.2.1.1 Etapa de Control.....	25
3.2.1.2 Etapa de Alimentación.....	26
3.2.1.2.1 Alimentación Alterna.....	27
3.2.1.2.2 Fuente de Alimentación Continua.....	27
3.2.1.3 Etapa de Activación del Sistema.....	28
3.2.1.3.1 Códigos que la alarma comunitaria interpretará para su ejecución.....	29
3.2.1.4 Etapa de Salida.....	29
3.2.1.4.1 Sirena.....	30
3.2.1.4.2 Motor.....	30
3.2.1.4.3 Cámara de video.....	32
3.2.1.4.4 Teléfono celular.....	33
3.2.2 Diseño del software.....	33
3.2.2.1 Diseño del software del Arduino Uno.....	33
3.2.2.2 Configuración del modem GSM.....	34

3.2.3 Diseño mecánico.....	36
3.2.4 Montaje.....	38
3.2.4.1 Montaje de hardware.....	38
3.2.4.2 Montaje del software.....	41
3.2.4.2.1 Programación con software Arduino.....	41
3.2.4.2.2 Programa para la obtención de los códigos Hexadecimales.....	43
3.2.4.2.3 Diseño de la programación del sistema de control del Prototipo.....	44
3.2.5 Montaje de la parte mecánica.....	51
3.3 Implementación del sistema de control prototipo de Alarma Barrial.....	53
3.3.1 Implementación del hardware.....	53
3.3.2 Implementación del sistema.....	56
 CAPÍTULO 4 RESULTADOS Y COSTOS	
4.1 Pruebas de funcionamiento.....	58
4.1.1 Pruebas de funcionamiento con SIM de diferentes operadoras.....	58
4.1.2 Pruebas del funcionamiento del sistema.....	59
4.2 Análisis de resultados.....	63
4.2.1 Análisis de las pruebas realizadas con SIM diferentes.....	63
4.2.2 Análisis de las pruebas del funcionamiento del sistema.....	64
4.3 Matriz FODA.....	65
4.4 Costos del Proyecto.....	66
 CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES	
5.1 Conclusiones.....	70
5.2 Recomendaciones.....	72
BIBLIOGRAFÍA.....	73
ANEXOS.....	74
 ANEXO 1 Datos técnicos de la placa electrónica Arduino Uno.....	75
ANEXO 2 Datos técnicos del microcontrolador ATmega 328P.....	77
ANEXO 3 Datos técnicos del Modem GSM ZTE MG3006.....	80

ANEXO 4 Filmadora SONY modelo DCR-HC42.....	85
ANEXO 5 Emisor IR TSUS5400.....	87

ÍNDICE DE TABLAS

Tabla 3.1 Lista de comandos	29
Tabla 3.2 Botones del control remoto con su respectivo código hexadecimal.....	32
Tabla 4.1 Configuración del SIM.....	58
Tabla 4.2 Resultados de prueba realizada.....	58
Tabla 4.3 Resultados obtenidos de la prueba de funcionamiento del sistema.....	64
Tabla 4.4 Matriz FODA.....	65
Tabla 4.5 Costo de materiales utilizados en la placa electrónica de Control.....	66
Tabla 4.6 Costo de materiales utilizados en la placa electrónica del motor.....	66
Tabla 4.7 Costo de materiales utilizados en la placa de alimentación.....	67
Tabla 4.8 Costo de materiales utilizados en la placa de control de la sirena.....	67
Tabla 4.9 Costo de materiales utilizados en la elaboración de la caja metálica.....	68
Tabla 4.10 Costo otros materiales utilizados.....	68
Tabla 4.11 Costo de mano de obra.....	68
Tabla 4.12 Costo total del prototipo.....	69

ÍNDICE DE FIGURAS

Figura 2.1 Arquitectura de la red GSM.....	6
Figura 2.2 Servicio SMS.....	8
Figura 2.3 Envío de un SMS entre un MS y un servidor.....	8
Figura 2.4 Arquitectura de la red SMS.....	9
Figura 2.5 Módulo GSM-USB.....	10
Figura 2.6 Módulo GSM con puerto RS-232.....	10
Figura 2.7 Versiones del protocolo SIRC.....	13
Figura 2.8 Codificación de bits en el protocolo SIRC.....	14
Figura 2.9 Modem ZTE MG3006.....	15
Figura 2.10 Entorno Arduino.....	17
Figura 2.11 Sirena.....	18
Figura 2.12 Motor.....	19
Figura 2.13 Emisor IR.....	19
Figura 2.14 Batería de 12 Vdc.....	20
Figura 3.1 Placa Arduino Uno.....	21
Figura 3.2 Descripción gráfica de distribución de elementos en placa electrónica Arduino Uno.....	22
Figura 3.3 Distribución de los pines del microcontrolador ATmega 328.....	23
Figura 3.4 Diagrama de bloques general.....	24
Figura 3.5 Diagrama de Bloques de la Etapa de Hardware.....	25
Figura 3.6 Diagrama circuital de la placa electrónica Arduino Uno.....	25
Figura 3.7 Diagrama circuital de la etapa de alimentación.....	27
Figura 3.8 Diagrama circuital de la etapa de comunicación.....	28
Figura 3.9 Diagrama de bloques de la etapa de salida.....	29
Figura 3.10 Diagrama circuital de la etapa de activación de la sirena.....	30
Figura 3.11 Diagrama circuital para el control del movimiento de la cámara de video.....	31
Figura 3.12 Emisor IR para la activación del sistema de grabación de video.....	32
Figura 3.13 Diagrama de flujo para el diseño del software de la placa electrónica Arduino Uno.....	34
Figura 3.14 Diagrama de flujo de la configuración del modem GSM.....	35

Figura 3.15 Dimensiones de caja contenedora de la alarma.....	36
Figura 3.16 Interior de la caja contenedora.....	37
Figura 3.17 Circuito receptor IR.....	38
Figura 3.18 Circuito general implementado en el protoboard.....	39
Figura 3.19 Diagrama circuital final implementado en ISIS.....	40
Figura 3.20 Selección de placa Arduino Uno.....	41
Figura 3.21 Selección del puerto serial asignado.....	42
Figura 3.22 Librerías necesarias para ejecutar el programa.....	43
Figura 3.23 Programa implementado para la recepción IR.....	43
Figura 3.24 Códigos hexadecimales obtenidos del circuito receptor IR.....	44
Figura 3.25 Parte interior de la caja.....	52
Figura 3.26 Vista frontal de la caja.....	52
Figura 3.27 Circuitos impresos o PCB.....	54
Figura 3.28 Circuitos finales correspondientes a cada etapa del prototipo.....	55
Figura 3.29 Montaje final de los circuitos electrónicos.....	55
Figura 3.30 Montaje del sistema en la caja metálica.....	56
Figura 3.31 Sitio elegido para la implementación del prototipo.....	56
Figura 3.32 Zonas monitoreadas.....	57
Figura 3.33 Amarra metálica.....	57
Figura 3.34 Implementación final del prototipo.....	57
Figura 4.1 Configuración del Modem.....	59
Figura 4.2 Motor moviéndose hacia la izquierda.....	60
Figura 4.3 Motor moviéndose hacia la derecha.....	61
Figura 4.4 Motor moviéndose hacia el centro.....	62
Figura 4.5 SMS generado al desactivar el sistema.....	63

RESUMEN

El presente proyecto consiste en el desarrollo de un sistema prototipo de alarma comunitaria, realizado bajo software libre y con la placa electrónica Arduino Uno; que incorpora una cámara de video con capacidad de grabar en tiempo real acoplada a un motor que se mueve a tres zonas previamente elegidas. Para la activación del sistema se utiliza SMS específicos asignados a cada zona monitoreada por la cámara de video, los mismos que son enviados hacia un modem GSM para luego transmitirlos por medio de un puerto RS-232 hacia la placa electrónica Arduino Uno la misma que los procesa y activa el sistema. Además el sistema informa a los usuarios registrados via SMS que el sistema fue activado.

Este informe técnico consta de 5 capítulos. El Capítulo 1 presenta la problemática, los antecedentes, los objetivos y la metodología utilizada para su desarrollo.

En el capítulo 2 se presentan conceptos y características técnicas relevantes de los diferentes dispositivos que sirven para la elaboración del prototipo.

El capítulo 3 muestra el diseño del proyecto, montaje con las pruebas y la implementación del mismo.

El capítulo 4 muestra las pruebas finales realizadas para verificar el correcto funcionamiento del sistema, contiene una matriz FODA en la que se analizó los pros y contras del proyecto y además se muestran los costos totales del mismo.

El capítulo 5 presenta una serie de conclusiones sacadas en el desarrollo del proyecto y recomendaciones que pueden servir para mejorar el sistema y/o un mejor uso del mismo.

ABSTRACT

This project involves developing a prototype community alarm system, carried out under free software and electronic board Arduino Uno, which incorporates a video camera capable of recording real-time coupled to a motor that moves to three areas previously chosen. To activate SMS system used specific area assigned to each monitored by video camera, they are sent to a GSM modem and then transmit them via an RS-232 port to the Arduino Uno electronic board the same as the processes and activates the system. Moreover, the system informs registered users via SMS that the system was activated.

This technical report consists of five chapters. Chapter 1 introduces the problem, background, objectives and the methodology used for development.

Chapter 2 presents relevant concepts and technical characteristics of the different devices used for prototype development.

Chapter 3 shows the project design, assembly with testing and implementation.

Chapter 4 shows the final tests performed to verify proper operation of the system, contains a SWOT matrix which analyzed the pros and cons of the project and also shows the total cost of the same.

Chapter 5 presents a series of lessons learned in the development of the project and recommendations that can be used to improve the system and / or a better use.

CAPÍTULO 1

PROBLEMATIZACIÓN

1.1 Antecedentes

El sector elegido para la implementación del prototipo se encuentra ubicado al norte de la Ciudad de Quito en el sector de Marianitas de Calderón, antes de su lotización fue un conjunto de terrenos llamados “Compañía Empresarial AASCTICOPE”. En el año de 1994 fue lotizado con el nombre de “Marianitas del Norte”, actualmente lleva el nombre de “Comité Pro Mejoras Jardines de Marianitas”. Está conformado por veinte y cinco hogares unifamiliares y un conjunto multifamiliar construidos en bloque y hormigón, las vías de acceso son empedradas.

Las familias que habitan en el sector son de clase económica media, la mayor parte de los habitantes trabajan y envían a sus hijos a estudiar en la Ciudad de Quito mientras que una reducida parte de la población trabaja en sectores aledaños como Carapungo o Calderón.

Para precautelar la seguridad en los diferentes barrios de la ciudad el Municipio de Quito inició en el año 2003 la instalación de alarmas comunitarias, los primeros sistemas de seguridad instalados utilizaban sensores de movimiento instalados en cada una de las casas y si se detectaba alguna anomalía disparaba la alarma y enviaba una señal de alerta a las computadoras de las centrales de monitoreo, para el año 2005 el sistema mejoró ya que se comenzó a utilizar la tecnología de Radio Frecuencia, cada usuario del sistema tenía un pulsador inalámbrico para su activación ante un hecho delictivo.

A partir del año 2010 los sistemas de alarmas comunitarias migraron a una nueva tecnología y con esto mejoró su radio de cobertura ya que utilizaban la red GSM para su activación a través de SMS, si el usuario es víctima de la delincuencia envía un SMS al sistema, el mismo que se activará automáticamente y simultáneamente enviará la señal de alarma a los celulares de los teléfonos celulares de los Policías de la UPC más cercana.

Producto de la evolución de la tecnología en el campo de la electrónica, hoy en día los sistemas de alarmas comunitarias han mejorado en cuanto a sus prestaciones.

1.2 Problema investigado

Las alarmas comunitarias implementadas en la Ciudad de Quito utilizan pulsadores inalámbricos con capacidad para cubrir a 24 familias; utilizan tecnología de Radiofrecuencia, en caso de requerir su activación se debe tener línea de vista con el sitio donde fue instalada de lo contrario no podrán ser activadas.

En sectores piloto de la ciudad de Quito se está implementando “EL BOTÓN DEL PÁNICO” por parte del Ministerio del Interior, el cual consiste en registrar datos estratégicos del beneficiario como dirección exacta de su domicilio, número de casa, nombres completos de la persona propietaria del número celular en un software instalado en los computadores del UPC (Unidad de Policía Comunitaria) más cercano a su domicilio; en el momento que sea víctima de un hecho delictivo, el usuario marca el número “5” desde su celular y automáticamente se enviará un mensaje de texto con los datos de la persona que activó la alarma a los teléfonos móviles de los policías del UPC, para el auxilio inmediato, el tiempo de respuesta es de 3 minutos. Por el momento este sistema de seguridad se encuentra en fase de pruebas por lo que no todos los ciudadanos pueden acceder a este servicio.

Por limitaciones de carácter técnico en ciertos sectores de la ciudad, el Municipio de Quito ha optado por la implementación de Alarmas Comunitarias que trabajan a través de redes GSM y pueden ser activadas mediante un teléfono celular sin necesidad de tener línea de vista, y debido a que utilizan este tipo de tecnología, estos sistemas tienen un costo de adquisición e instalación elevado.

Con la investigación realizada en el internet se encontró que en Chile se está implementando sistemas de alarmas comunitarias vía mensaje de texto, que permitirán actuar y proteger a las víctimas de delitos de la comuna. En la Ciudad de Bogotá las autoridades locales se encuentran implementando sistemas de seguridad parecidos, los mismos que cubren a 20 casas, en cada domicilio se encuentra instalado un “BOTON DE VIDA”. Dichas alarmas cuentan con un dispositivo que les permitirá a los ciudadanos llamar a la Policía del cuadrante en una situación de emergencia. El sistema está diseñado para enviar un mensaje de texto con la ubicación exacta del ciudadano que está haciendo la solicitud, con lo que se han reducido los índices de criminalidad y ha mejorado la relación comunidad-Policía.

1.3 Problema principal

No existe un Sistema Prototipo de Alarma Barrial con Sistema de Grabación de video en tiempo real incorporado y activado vía SMS en la Ciudad de Quito.

1.4 Problemas secundarios

- No existen estudios en el Ecuador sobre Sistemas de Alarmas Barriales que utilicen tecnología Arduino.
- Existen varios sistemas de alarmas comunitarias que pueden ser activados por medio de teléfonos celulares pero no tienen sistema de grabación de video incorporado.
- Existen varios tipos de tecnología para la implementación de este sistema prototipo.

1.5 Justificación

El trabajo de investigación presentado servirá para estudiar y conocer cada uno de los elementos y dispositivos electrónicos a ser empleados en el presente proyecto y aplicar los conocimientos de electrónica digital, transmisión y recepción de datos, programación con tecnología Arduino, control de periféricos externos y redes GSM que se utilizarán para su diseño e implementación, con lo que se entregará a la comunidad un prototipo de alarma barrial que tenga mayor radio de cobertura y pueda tener un mayor número de usuarios que se beneficien del sistema, el mismo que utilizará la red GSM para su activación y control por medio de SMS. Además incorpora un sistema de grabación en tiempo real que se activará automáticamente, y esto ayudará a la Policía a obtener evidencias físicas del hecho delictivo, por otro lado se trata de un proyecto que reducirá los costos de adquisición e implementación, para brindar mayor seguridad a los usuarios de los sectores donde se instalen este tipo de alarmas comunitarias.

1.6 Objetivos

1.6.1 Objetivo principal

- Estudiar, Diseñar e implementar un sistema prototipo de alarma barrial y sistema de grabación de video activado por SMS.

1.6.2 Objetivos secundarios

- Estudiar la tecnología Arduino para ser aplicada en Alarmas Barriales que incorporen Sistemas de grabación de video en tiempo real.
- Diseñar un sistema de control que cumpla con las especificaciones del prototipo de Alarma Barrial.
- Implementar un sistema de control utilizando la tecnología que requiere el prototipo de Alarma Barrial.

1.7 Metodología

El presente proyecto se dividió en cuatro fases y en cada una de ellas se utilizó un método de investigación.

La primera fase comprendió el reconocimiento, planteamiento del problema y las posibles soluciones para la fabricación del prototipo de alarma barrial por lo que se utilizó el método de análisis y síntesis.

En la segunda fase se desarrolló el diseño del prototipo, en la que se utilizaron los métodos deductivo e inductivo ya que se investigó varios conceptos tecnológicos para llegar a determinar cuál es el tipo de tecnología que cumpla con los requerimientos necesarios para el desarrollo del diseño.

La tercera fase comprendió la implementación del prototipo, el método de investigación utilizado en esta etapa fue la experimentación con la que se procedió a la correcta implementación de la alarma barrial.

En la cuarta etapa se utilizó el método experimental para la validación del prototipo mediante pruebas de funcionamiento y con esto se demostró que trabaja correctamente y sin problemas.

CAPÍTULO 2

MARCO DE REFERENCIA

En el presente capítulo se hace una descripción detallada de las herramientas y tecnologías utilizadas para el desarrollo del prototipo.

2.1 Marco teórico

2.1.1 Tecnología GSM

GSM o *Global System for mobile communications*, (Sistema global para las comunicaciones móviles), es el nombre del grupo de estandarización establecido en 1982 para la creación de un estándar de telefonía móvil europeo que opera a 900 MHz. En 1989 la responsabilidad de GSM se trasladó al ETSI (*European Telecommunications Standards Institute*) y las primeras especificaciones fueron emitidas en 1990.

Este tipo de telefonía móvil es de segunda generación (2G), actualmente es el estándar más popular del mundo, esto quiere decir que los usuarios pueden usar sus teléfonos en todo el mundo, por acuerdos internacionales de *roaming* entre operadores de redes móviles.

El estándar GSM presta los siguientes servicios:

- Transmisión y recepción de voz y datos
- Envío y recepción de mensajes cortos de texto (SMS)
- Servicios complementarios (llamada en espera, multi conferencias, identificador de llamadas, entre otros)

2.1.1.1 Arquitectura de la red GSM¹

Su arquitectura se basa en el reparto del espectro disponible debido a la limitación del rango de frecuencias disponibles, ya que cada conversación requiere un mínimo de ancho de banda. A cada compañía proveedora de servicios de telefonía móvil se le

¹ <http://repositorio.espe.edu.ec/bitstream/21000/3316/1/T-ESPE-031145.pdf>

asigna cierto ancho de banda con frecuencias determinadas para evitar interferencias entre usuarios. En la figura 2.1 se muestra la arquitectura de una red GSM de manera resumida.

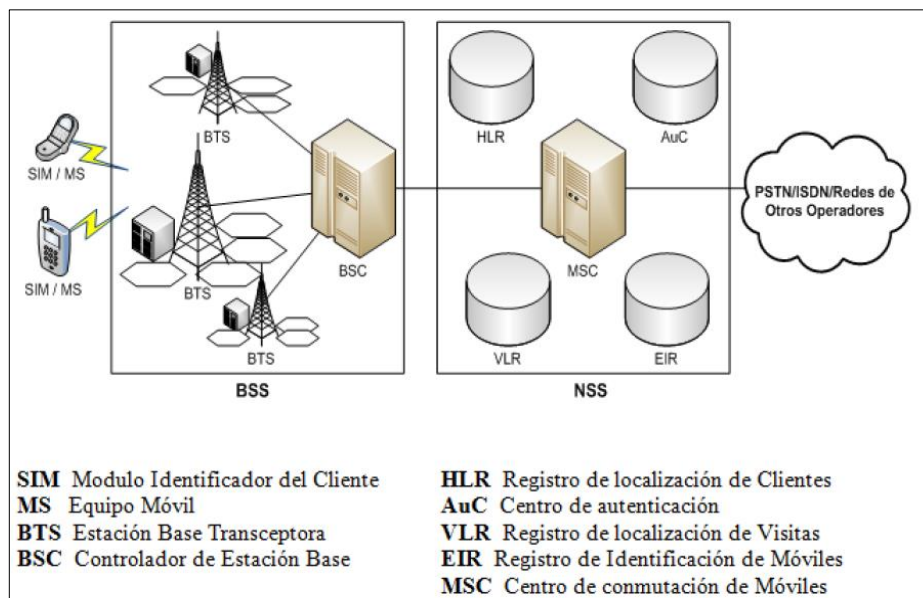


Figura 2.1 Arquitectura de la red GSM

Su arquitectura está basada en capas, las mismas que se detallan a continuación:

1. **Estación Móvil (MS)**, son terminales digitales que pueden ser portables o incluso portátiles, los mismos que incorporan un dispositivo **SIM** (Módulo identificador de cliente).
2. **Subsistema de Estación Base (BSS)**, es una colección de dispositivos que soportan interface de radio de redes de conmutación. Los principales componentes del BSS son:
 - a) **Estación Transceptora de Base (BTS)**, consta de módems, radios y antenas.
 - b) **Controlador de Estación Base (BSC)**, gestiona las operaciones de radio de varias BTS.
3. **Subsistema de Red y Conmutación (NSS)**, suministra la comunicación entre el subsistema GSM, cuyos componentes son:
 - a) **Centro de Conmutación de Servicios Móviles (MSC)**, es el centro de conmutación.
 - b) **Base de Datos.**

4. **Centro de Monitoreo y Operaciones (OCC)**, controla la operación del sistema y la inicialización de la red, también gestiona los equipos móviles.

2.1.1.2 Características técnicas

Las bandas asignadas por parte de la UIT (Unión Internacional de Telecomunicaciones) para el estándar GSM en la mayor parte del mundo está en el rango de: 900MHz y 1800MHz; en el Ecuador se utilizan las bandas de frecuencia de 850MHz y 1900MHz, a partir de estos datos, las principales características técnicas de la red GSM son:

- **Canales:** 124 canales, cada canal da un servicio de 8 a 16 usuarios a la vez.
- **Bandas:** Usa dos bandas de 25 MHz para transmitir y recibir información.
- **Subida:** Usa la banda de 824MHz a 849MHz.
- **Bajada:** Usa la banda de 869MHz a 894MHz.
- **Ancho de banda:** 20KHz.
- **Modulación:** GMSK (Gaussian Minimum Shift Keying).
- **Velocidad máxima de canal de radio:** 270.833Kbps.
- **Duración de bit:** 3,692 ms.
- **Longitud de trama:** 6,615 ms.
- **Longitud de slot de tiempo:** 577us.
- **Codificación de voz:** PRE-LPT 13Kbps.
- **Potencia de salida:** 20mW a 20W.

2.1.2 Servicio de mensajes cortos de texto SMS

El servicio de SMS permite enviar un mensaje de texto entre una Estación Móvil (MS) y otra entidad denominada SME (Entidad de mensajes cortos) por medio de un centro de servicio SMSC (Centro de servicio de mensajes cortos), en la figura 2.2 se muestra el proceso descrito.

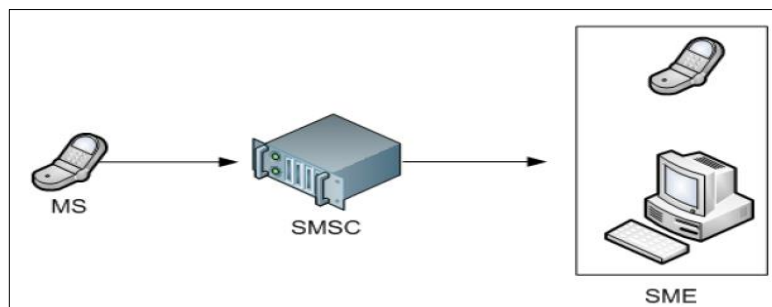


Figura 2.2 Servicio SMS

Como se puede apreciar en la figura 2.2 la entidad que recibe el SMS puede ser otro terminal MS.

Cuando se envía un mensaje para solicitar un servicio, un extremo es un MS y al otro extremo está un servidor que atiende las peticiones de los usuarios como consulta de saldos, activación de servicios, entre otros, como se muestra en la figura 2.3.

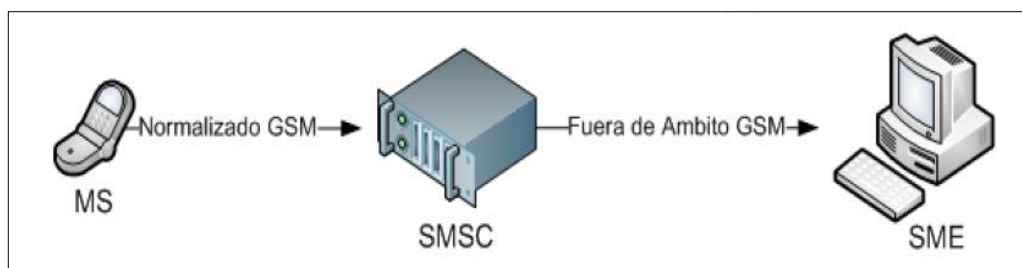


Figura 2.3 Envío de un SMS entre un MS y un servidor

2.1.2.1 Formato del SMS

Dentro de las especificaciones de los SMS existe la posibilidad de realizar el envío de mensajes de dos maneras, las mismas que son:

- a) **Modo PDU**, esta estructura de mensaje lleva consigo bits de información específica, además de funciones de control para presentación del mensaje.
- b) **Modo Texto**, esta estructura está conformada por caracteres de texto, números y símbolos, es un modo de gama media que no se encuentra en todos los terminales MS.

2.1.2.2 Arquitectura del SMS

La red básica SMS se encuentra compuesta por varias capas las mismas que se explican de manera rápida a continuación y se las presentan en la figura 2.4.

- **MS**, Estación Móvil.
- **MSC**, Centro de Conmutación.
- **SMS-GMSC**, es una puerta de enlace para que el MSC resuelva el servicio del SMS.
- **SMS-IWMSC**, (Internetworking MSC for Short Message Service), puerta para hacer la interconexión entre el MSC y resolver el servicio SMS.
- **SMSC**, Centro de servicios.
- **HLR**, Base de datos para registro de la localización de los usuarios.
- **VLR**, Base de datos para localización de visitas.

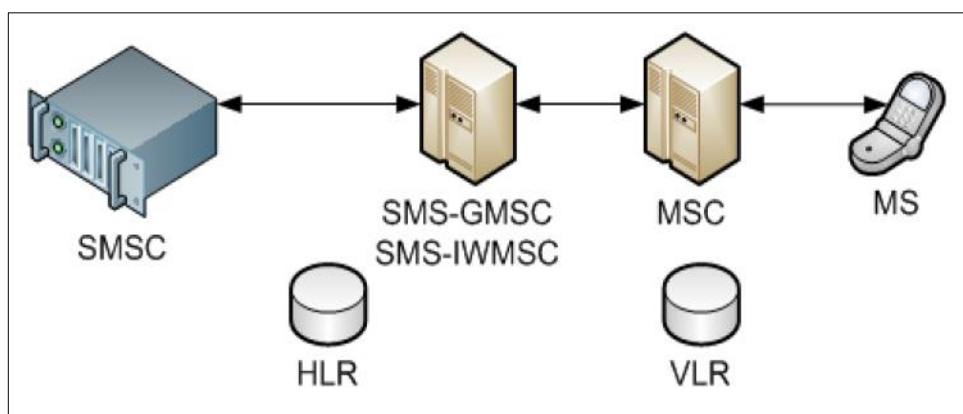


Figura 2.4 Arquitectura de la red SMS

2.1.3 Módulo GSM²

El módulo GSM es un dispositivo móvil que permite a los usuarios acceder a una red de telefonía móvil GSM, con el que se pueden realizar llamadas de voz, envío y recepción de mensajes cortos de texto (SMS). En el mercado existe gran variedad de dispositivos que cumplen con estos requerimientos, los mismos que van desde teléfonos móviles, módulos USB o módulos con puerto serial RS 232 los mismos que pueden ser acoplados a un computador personal para poder ser configurados.

² <http://186.42.96.211:8080/jspui/bitstream/123456789/898/1/bsanguano-ecalvache.pdf>

2.1.3.1 Módulo GSM-USB

Son de pequeño tamaño, por lo que son ideales para proyectos donde se tenga un espacio de instalación reducido, sin embargo, para su correcto funcionamiento se requiere la instalación de un software en una Pc, en la figura 2.5 se muestra un modem GSM-USB.



Figura 2.5 Módulo GSM-USB³

2.1.3.2 Módulo GSM RS-232

Este tipo de módulos poseen una interfaz de comunicación serial, lo cual es ideal para la realización de proyectos, ya que es independiente y no requiere la instalación de ningún tipo de software en las Pcs, sin embargo son de mayor tamaño y la mayor parte de este tipo de módems no poseen interfaz de audio por lo que no permiten la codificación DTMF, en la figura 2.6 se puede apreciar este tipo de módulos.



Figura 2.6 Módulo GSM con puerto RS-232⁴

³ Sony Ericsson MD300 Mobile Broadband USB / User Guide/ 2aec./2008

⁴ <http://www.mc-technologies.net/images/cellular/xt55.gif>

2.1.4 Comandos AT⁵

Los comandos AT son un grupo de instrucciones que conforman un lenguaje de comunicación entre los usuarios y un MODEM.

Desarrollados en 1977 por Dennis Hayes, como un medio de comunicación con un terminal MODEM, para así poder configurarlo suministrándole instrucciones para que sea capaz de marcar un número de teléfono.

Compañías como Microcomm y US Robotics desarrollaron nuevos comandos AT a partir de los ya existentes y con esto ampliaron su gama, a los nuevos comandos los denominaron AT extendidos y comienzan con los caracteres AT+.

2.1.4.1 Estructura de los comandos AT

Los comandos AT se encuentran conformados por cadenas de caracteres ASCII; para su ejecución se debe anteponer el prefijo AT. El prefijo AT proviene de la palabra *Attention* que solicita al modem atender a la solicitud de comando recibida.

La estructura para enviar un comando AT hacia un modem GSM se presenta a continuación:

AT+	CMGF=1	<CR>
PREFIJO	COMANDO	SUFIJO

- El prefijo de los comandos AT debe ser la cadena de caracteres “AT”, el signo “+” también debe ser colocado después de estos caracteres.
- El sufijo de los comandos AT debe ser <CR> (Retorno de carro), que es equivalente a ENTER.
- Los comandos pueden ser escritos con letras mayúsculas o minúsculas.
- Cuando se coloca el signo “=” a un comando, se está configurando un parámetro, cuando se coloca el signo de interrogación “?” se está pidiendo información.

La respuesta del modem ante un comando tiene la siguiente estructura:

⁵ <http://dspace.ups.edu.ec/bitstream/123456789/166/3/Capitulo%202.pdf>

<CR><LF>	OK	<CR><LF>
PREFIJO	COMANDO	SUFIJO

- Los caracteres “OK” corresponden a una conexión exitosa, caso contrario aparecerá “ERROR”.
- El prefijo y sufijo consta de <CR> (Retorno de carro) y <LF> (Salto de línea).

2.1.4.2 Comandos AT generales

A continuación se presenta un resumen de comandos AT que pueden ser utilizados en cualquier modem GSM o teléfono celular:

- **Comandos para información del equipo**
 - ✓ AT+CMGI: Información el fabricante.
 - ✓ AT+CGNS: Obtener # de serie.
 - ✓ AT+CPAS: Leer estado del modem.
- **Comandos de servicio de red**
 - ✓ AT+CSQ: Obtener calidad de la señal
 - ✓ AT+COPS: Selección de un operador.
 - ✓ AT+WOPN: Lee nombre de operadora.
- **Comandos para SMS**
 - ✓ AT+CMGF: Selecciona el formato de los mensajes de texto.
 - ✓ AT+CMGR: Lee un mensaje de texto almacenado.
 - ✓ AT+CMGL: Lista los mensajes almacenados.
 - ✓ AT+CMGS: Envía un SMS.
 - ✓ AT+CMGW: Almacena mensaje en la memoria interna.
 - ✓ AT+CMSS: Envía un mensaje almacenado.

2.1.5 Protocolo SIRC SONY⁶

Las señales transmitidas bajo el protocolo SONY IR más conocido como SIRC se caracterizan por ser de nivel TTL es decir 5 V y se puede emitir directamente del pin de un microcontrolador, su frecuencia de portadora es 40KHz.

⁶ <http://libertadelectronica.wordpress.com/2009/12/03/decodificando-el-infrarrojo-sirc/>

Existen tres versiones de este protocolo: 12 bits, 15 bits y 20 bits; las tres mismas que tienen un campo inicial que es el bit de inicio y dos palabras, una para el comando a enviar y la otra para la dirección o dispositivo.

Dentro de la trama de datos cada palabra es leída desde el bit menos significativo (LBS) hasta el bit más significativo (MBS). En la figura 2.7 se puede apreciar las tres versiones de dicho protocolo.

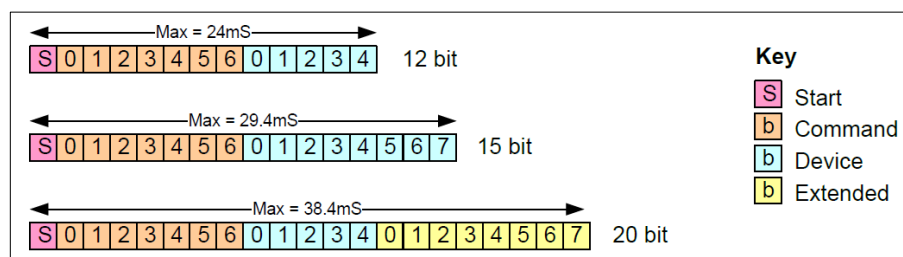


Figura 2.7 Versiones del protocolo SIRC⁷

Los bits representados en la figura anterior se dividen en:

- Señal de inicio (S) + 7 bits para enviar el comando + 5 u 8 bits para el código del equipo + 8 bits añadidos adicionalmente (versión de 20 bits).

El protocolo SIRC usa codificación de bits basada en ancho de pulso para poder enviar paquetes de datos. Los receptores IR de equipos domésticos como televisores, radios, entre otros trabajan con lógica inversa, es decir, cuando reciben una señal infrarroja (IR) cambian a estado 0 lógico y cuando no reciben una señal están en estado 1 lógico.

2.1.5.1 Codificación

Bit de inicio, (S) le indica al dispositivo a controlar que se aliste para recibir datos, la longitud de este pulso es $4T$ (2.4 ms).

Bit 1, su codificación se la realiza por medio de un espacio (vacío) de longitud T , y un pulso igual a T .

Bit 0, su codificación se la realiza por medio de un espacio (vacío) de longitud T , y un pulso igual a $2T$ que equivale a 1,2ms.

⁷ <http://picprojects.org.uk/projects/sirc/sonysirc.pdf>

En la figura 2.8 se muestra la codificación del protocolo SIRC.

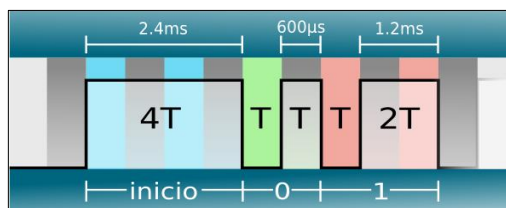


Figura 2.8 Codificación de bits en el protocolo SIRC

Como se puede observar es fácil diferenciar los ceros (0) de los unos (1) y los bits de inicio; la codificación en el protocolo SIRC es muy simple, se restringe a enviar pulsos de duración T, 2T y 4T, por lo no pueden existir equivocaciones al momento de enviar los códigos de control, y con esto pueden ser fácilmente detectados por el receptor.

2.1.6 Software libre⁸

Software libre significa que el software puede ser modificado por los usuarios, los mismos que tienen la libertad de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software, además esto quiere decir que el usuario tiene cuatro libertades sobre el software:

- Libertad de usar el software, con cualquier propósito.
- Libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades, esto quiere decir que se debe tener acceso al código fuente.
- Libertad para distribuir copias, con lo que puedes ayudar otros usuarios.
- Libertad de mejorar el software y publicar las mejoras a los demás usuarios, de modo que todos se beneficien, esto quiere decir que se debe tener acceso al código fuente.

2.1.6.1 Hardware libre

Hardware libre son todos aquellos dispositivos cuyas especificaciones y diagramas circuitales son de libre acceso, ya sea por cierta cantidad de dinero o de forma gratuita.

⁸

<http://186.42.96.211:8080/jspui/bitstream/123456789/156/1/SISTEMA%20CONTROL%20DE%20RUTA%20VEHICULAR%20MEDIANTE%20COMPARACION%20CON%20EL%20USO%20DE%20GPS.pdf>

2.2 Marco conceptual

En esta etapa se describen las características de los dispositivos electrónicos principales utilizados para la construcción del prototipo.

2.2.1 Especificaciones del módulo GSM

El modem GSM utilizado pertenece al fabricante ZTE, modelo MG3006 debido a que se acopla perfectamente a la placa electrónica Arduino Uno, este modelo consta de antena integrada fácil de manipular, es pequeño y potente. Al modem se lo puede apreciar en la figura 2.12:



Figura 2.9 Modem ZTE MG3006

Las características principales de este equipo son:

- Quad-band 850/900/1800/1900 MHz
- Tecnología GSM
- Acceso a través de comandos AT
- Socket para tarjeta SIM
- Rango de alimentación de voltaje: 7V – 25V
- Puerto serial RS232 (BB9)
- Dimensiones: 75mm*50mm*16mm
- Peso < 200G
- Rango de temperatura de operación: -30°C – 75°C

2.2.2 Comandos AT utilizados para la configuración del modem GSM⁹

La configuración del modem GSM se la realizará por medio de los siguientes comandos AT, detallados a continuación:

- **AT**, este es un comando de atención cuya función es monitorear si existe una buena conexión, si la conexión es buena el modem responde OK.
- **AT+CNMI=1,2,0,0,0**, este comando establece el formato del SMS que recibirá el modem. Cuando el modem recibe un nuevo SMS, en el HyperTerminal mostrará el siguiente encabezado en todos los SMS recibidos en el modem:

+ CMT: "61 xxxxxxxx",, "08/04/30, 23:20:00 40"

- **AT+CMGF=1**, este comando permite elegir el modo de interpretación de los datos por parte de modem, como se puede observar es igual a “1”, esto quiere decir que todos los datos serán interpretados en modo de texto, es decir, la secuencia de caracteres que se envía al modem son ASCII normales. El modem al momento de recibir este comando responde OK, indicando que la petición ha sido aceptada.
- **AT+CMGS=**, este comando permite enviar un mensaje SMS. A continuación se muestra un ejemplo de la sintaxis de este comando:

AT+CMGS="99XXXXXXXX" alt 13 <CR>: Return Carrier ... (Enter)

> Modem Listo.... Alt 26 ... (ctrl+z)

Respuesta: +CMGS: xxx

Fin de archivo

⁹ <http://oldlight.wordpress.com/2009/06/16/tutorial-using-at-commands-to-send-and-receive-sms/>

2.2.3 Software Arduino¹⁰

El entorno de Arduino facilita la escritura del código como la también el proceso de cargado a la placa de electrónica de E/S. El software Arduino puede ser utilizado en Windows, Mac, Linux y Androide.

2.2.3.1 Entorno Arduino

El entorno Arduino también es conocido como IDE para Arduino, al mismo que se lo puede apreciar en la figura 2.10.

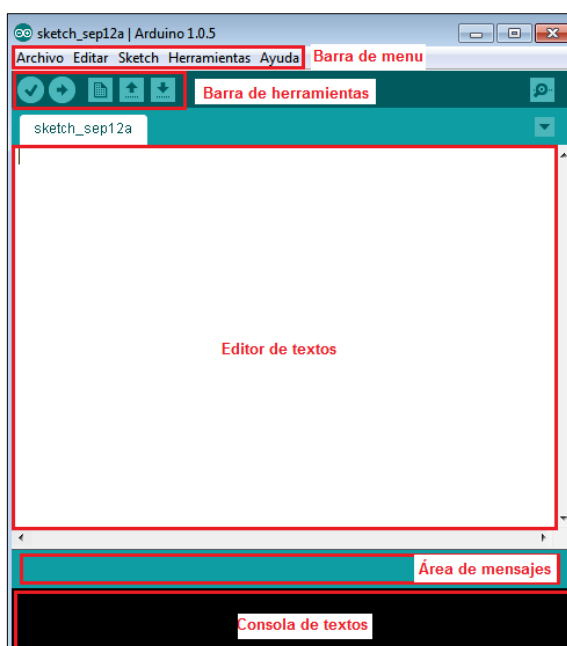


Figura 2.10 Entorno Arduino








El entorno de Arduino está compuesto por un editor de texto para escribir el código a realizar, una área para mensajes, una consola de texto, una barra de herramientas con botones visibles para las funciones comunes, y una barra de menús donde se puede configurar la conexión del hardware.

Los programas son realizados en el editor de texto, en el área de mensajes se muestra información mientras se cargan los programas a la placa electrónica, también muestra los errores que se producen en el transcurso de la compilación del programa. La consola

¹⁰ <http://arduino.cc/es/Main/Software>

de texto muestra el proceso de la compilación y los posibles errores que se producen en el proceso de compilación del programa.

- **Botones visibles en la barra de herramientas**

-  **Verificar**
 Chequea el código en busca de errores.
-  **Stop**
 Finaliza la monitorización serie y oculta otros botones
-  **Nuevo**
 Crea un nuevo *sketch*.
-  **Abrir**
 Presenta un menú de todos los programas *sketch* de su "sketchbook", (*librería de sketch*).
-  **Guardar**
 Salva el programa *sketch*.
-  **Cargar**
 Compila el código y lo envía a la placa E/S de Arduino.
-  **Monitor Serial**
 Inicia la monitorización serie.

2.2.4 Especificaciones de la sirena

La sirena que va ser utilizada cuenta con las características detalladas a continuación y se la puede apreciar en la figura 2.11.

- Voltaje de operación: 6 a 12 Vdc
- Consumo máximo de corriente: 500 mA (max)
- Potencia: 15 W
- Decibeles: 115 DB
- Dimensiones: diámetro 8,5 cm y largo 10,5 cm



Figura 2.11 Sirena

2.2.5 Especificaciones del motor

Las características principales del motor que va ser utilizado se presentan a continuación, y se lo puede apreciar en la figura 2.12.

- Alimentación: 3.3V
- Velocidad reversible: 200rpm
- Torque máximo: 17.7 lb/in
- Peso:180 g



Figura 2.12 Motor

2.2.6 Especificaciones del emisor IR

Las principales características del emisor IR utilizado se presentan a continuación y se lo puede apreciar en la figura 2.13.

- Tipo de filtro: plomo
- Dimensión: (en mm) $\varnothing 5$
- Pico de longitud de onda (λ): $\lambda=950$ nm
- Alta fidelidad
- Angulo de media intensidad: $\phi = \pm 22^\circ$
- Bajo voltaje de operación



Figura 2.13 Emisor IR

2.2.7 Especificaciones de la batería

Las principales características de la batería que va ser utilizada se presentan a continuación y se la puede apreciar en la figura 2.14.

- Voltaje: 12 Vdc
- Corriente máxima: 4 Ah
- DIMENSIONES: alto: 10 cm x largo: 9 cm x ancho: 7 cm



Figura 2.14 Batería de 12 Vdc

CAPÍTULO 3

ESTUDIO, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL PROTOTIPO DE ARMA BARRIAL Y SISTEMA DE GRABACIÓN DE VIDEO ACTIVADO POR SMS

Introducción

Este capítulo está orientado a presentar el diseño, montaje e implementación del prototipo electrónico de alarma comunitaria, en lo que respecta a hardware, software y estructura mecánica.

3.1 Estudio de la tecnología Arduino para ser aplicada en Alarmas Barriales

3.1.1 Arduino

Arduino es una plataforma electrónica libre para la creación de prototipos basados en software y hardware modificables y fáciles de usar.

El microcontrolador de la placa Arduino se programa mediante el lenguaje de programación Arduino (fundamentado en *Wirin*) y el entorno de desarrollo Arduino (IDE) (fundamentado en *Processing*).

Las placas electrónicas Arduino pueden ser realizadas a mano o se las puede adquirir ensambladas de fábrica; al software se lo puede descargar de forma gratuita. Las librerías están disponibles bajo una licencia abierta, así pues son libres de adaptarlas según las necesidades de cada usuario.

3.1.2 Hardware Arduino

Existen diferentes versiones de placas Arduino. La placa básica actual, es, la placa electrónica Arduino Uno, la que incorpora el microcontrolador Atmel ATmega328; en la figura 3.1 se muestra dicha placa electrónica.



Figura 3.1 Placa Arduino Uno

Arduino Uno, es la última revisión de la placa electrónica Arduino USB básica.

La placa electrónica Arduino Uno tiene un multi fusible que protege la conexión USB con el ordenador, de cortocircuitos y sobre tensiones; el fusible proporciona una protección extra; si más de 500 mA son detectados en el puerto USB, el fusible automáticamente elimina la conexión hasta que el cortocircuito o el sobre voltaje desaparezca.

3.1.3 Placa electrónica Arduino Uno

La placa electrónica Arduino Uno cuenta con: 14 pines entradas/salidas digitales, 6 entradas analógicas, un oscilador de cristal de 16 MHz, un conector USB, un conector para la alimentación de voltaje, cabecera ICSP y un pulsador de reset. En la figura 3.2 se muestra gráficamente cada una de las partes mencionadas anteriormente:

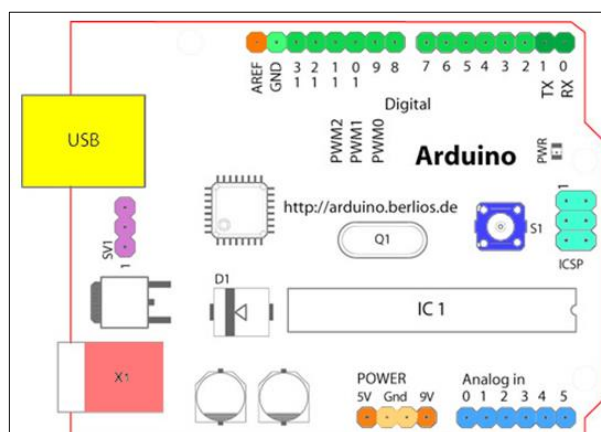


Figura 3.2 Descripción gráfica de distribución de elementos en placa la electrónica Arduino Uno¹¹

Dónde:

- **AREF** - Terminal de referencia analógica (naranja)
- **GND** - Tierra digital (verde claro)
- **DIGITAL** - Terminales digitales 2-13 (verde)
- Terminales digitales 0-1/ E/S serie: TX/RX (verde oscuro)
- **S1** - Botón de reset (azul oscuro)
- **ICSP** - Programador serie en circuito (azul celeste)
- **ANALOG IN** - Terminales de entrada analógica 0-5 (azul claro)

¹¹ <http://arduino.cc/es/Reference/Board>

- **POWER** - Terminales de alimentación y tierra (alimentación: naranja, tierras: naranja claro)
- **XI** - Entrada de alimentación externa (9-12VDC) (rosa)

La distribución de los pines del microcontrolador ATmega 328 se presenta en la figura 3.3, y sus principales características son:

- 32KB de memoria flash para programación
- Tamaño de datos RAM: 2KB
- Interfaz: 2-wire, SPI, USART
- Velocidad: 20MHz
- Puertos de entrada/salida programables: 23
- Temporizadores: 3
- Canales de ADC: 6 canales de 10 bits,
- Empaquetado: PDIP-28

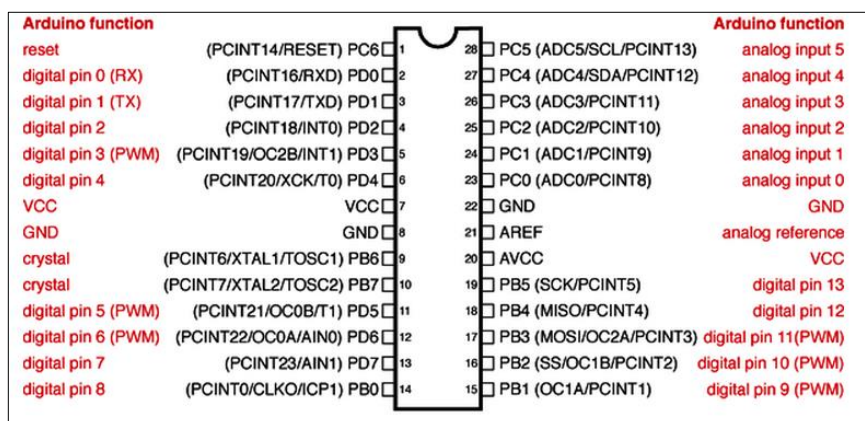


Figura 3.3 Distribución de los pines del microcontrolador ATmega 328¹²

3.2 Diseño del sistema de control prototipo de Alarma Barrial

Para el diseño del sistema prototipo de alarma comunitaria con sistema de grabación de video incorporado activado por SMS, se tomó en cuenta los siguientes requerimientos:

- Una fuente de poder que proporcione el voltaje adecuado para el correcto funcionamiento del sistema (12 Vdc).

¹² <http://arduino.cc/es/Hacking/PinMapping168>

- Un dispositivo que cumpla con la función de controlar el prototipo y grabe en video el hecho delictivo para que posteriormente lo almacene.
- Códigos de identificación para cada zona a las cuales se debe mover la cámara para iniciar el proceso de grabación.
- Una cámara de video que sea controlada mediante el módulo de control del sistema prototipo.
- Un sistema para que trabaje automáticamente con el uso de una batería de 12 Vdc en el caso de un corte de energía eléctrica.

Basándose en las necesidades descritas, el sistema prototipo tiene las siguientes etapas:

- Etapa de hardware
- Etapa de software
- Etapa mecánica

En la figura 3.4 se muestra la distribución general de las tres etapas que conforman el prototipo.

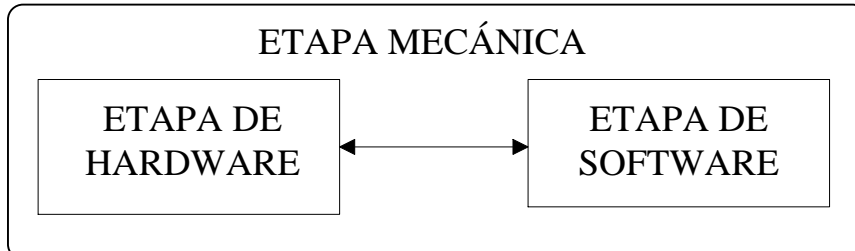


Figura 3.4 Diagrama de bloques general

A continuación se describen cada una de las etapas:

3.2.1 Diseño de Hardware

La etapa de hardware está constituida por los componentes electrónicos que se utilizarán para el diseño del prototipo.

A esta etapa se la ha dividido en bloques que van desde la Etapa de Alimentación, pasa por la Etapa de Inicialización, luego de esta se encuentra la Etapa de Control y culmina en la Etapa de Activación; a continuación se presenta en la figura 3.5 el diagrama de bloques de la Etapa de Hardware.

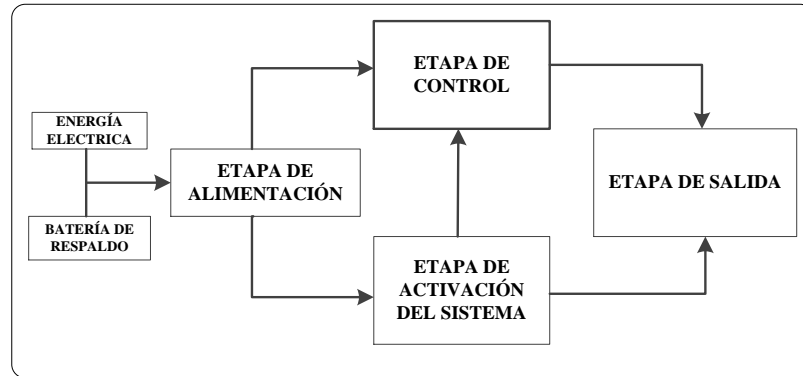


Figura 3.5 Diagrama de Bloques de la Etapa de Hardware

3.2.1.1 Etapa de Control

Esta etapa es la más importante del sistema, ya que en base a esta etapa se trabajará todo el circuito y es el cerebro del prototipo.

La etapa de control es la encargada de tomar las decisiones de activar: la sirena, el sistema de movimiento y el sistema de grabación de video.

Viendo la necesidad de trabajar en software y hardware libre se investigó entre varios tipos de tecnologías existentes en el mercado nacional para ser utilizada como Etapa de Control, y con esto determinar cuál cubre con las especificaciones requeridas por el prototipo, con este fin se tomó en cuenta la opción de Arduino.

Dentro de Arduino existen diversos diseños, la placa electrónica Arduino Uno es la que mejor se acopla a las necesidades del prototipo, su diagrama circuital se observa en la figura 3.6.

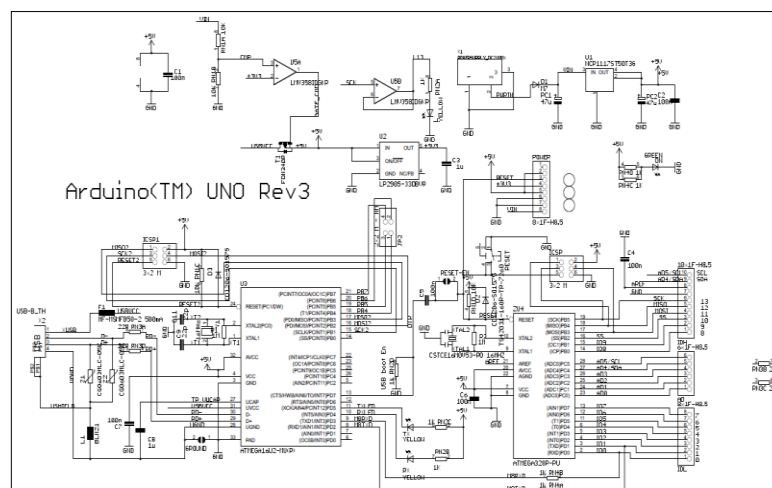


Figura 3.6 Diagrama circuital de la placa electrónica Arduino Uno¹³

¹³ http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

Los pines que se utilizarán para la elaboración del proyecto con su respectiva configuración se presentan a continuación:

- **Pin A0**, a este pin se conectará un fin de carrera (fin1), que se activará cuando el motor que es parte de la etapa de salida se mueva hacia a la derecha.
- **Pin A1**, a este pin se conectará un fin de carrera (fin2), que se activará cuando el motor que es parte de la etapa de salida se mueva hacia a la izquierda.
- **Pin A2**, a este pin se conectará un relé (RLY3) que será el encargado de controlar el encendido y apagado de la sirena que es parte de la Etapa de Salida.
- **Pin A4**, a este pin se conectará un relé (RLY1) que será el encargado de controlar la batería que suministra el voltaje necesario para el movimiento del motor que es parte de la Etapa de Salida.
- **Pin A5**, a este pin se conectará un relé (RLY2) que será el encargado de controlar el motor que es parte de la Etapa de Salida.
- **Pin reset**, en este pin se conectará un pulsador externo para reiniciar el sistema en caso de errores de funcionamiento.
- **Pin 3**, por este pin será utilizado para los códigos de activación de la grabación de video por medio de un emisor IR (D2) los mismos que forman parte de la Etapa de Salida.
- **Pin 8 y 9**, estos pines serán configurados como RX y TX respectivamente para la comunicación de la Etapa de Control con el modem GSM que es parte de la Etapa de Activación y Salida.
- **Pin 11, 12 y 13**, en estos pines se conectarán tres leds (D1, D3 y D4), que indicarán el estado del sistema.

3.2.1.2 Etapa de Alimentación

Dentro de esta etapa se tendrá dos fuentes de alimentación, la una es de corriente alterna que por medio de un circuito se convierte en una fuente de alimentación continua y la otra es una batería que se recarga mediante el mismo circuito al que está conectada.

3.2.1.2.1 Alimentación Alterna

La empresa eléctrica entrega un voltaje de corriente alterna de 110 Vac, por lo que se utilizará un transformador reductor que reducirá los 110 Vac que entran a la bobina primaria y entregará 12 Vac a través de la bobina secundaria y la corriente que entregará será de 1A.

La cámara de video que es parte de la Etapa de Salida se encontrará conectada directamente al suministro de energía eléctrica por medio del adaptador de voltaje ya que permanecerá encendida todo el tiempo de funcionamiento del prototipo.

3.2.1.2.2 Fuente de Alimentación Continua

Esta fuente cumple con la función de polarizar al sistema prototipo con 12 Vdc a: la placa electrónica Arduino Uno, al módulo GSM y a la sirena respectivamente.

Además tiene la característica de funcionar con una batería recargable que en caso de que el servicio eléctrico sea suspendido por cualquier imprevisto la misma entrará en marcha y proporcionará la energía suficiente al sistema para que su funcionamiento sea ininterrumpido durante el corte de suministro eléctrico. La batería se recargará mediante el mismo circuito de fuente.

Para la alimentación de corriente continua se utilizarán los elementos que se muestran en el diagrama circuital de la figura 3.7:

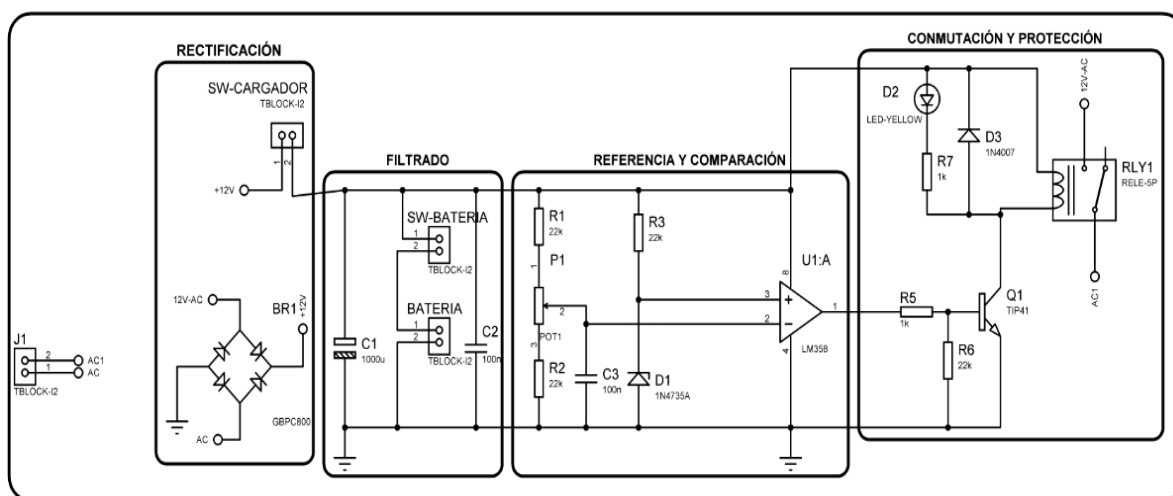


Figura 3.7 Diagrama circuital de la etapa de alimentación

Como se observa en la figura 3.7, la etapa de alimentación cuenta con cuatro fases que son: rectificación, filtrado, referencia/comparación y conmutación/protección.

3.2.1.3 Etapa activación del sistema

El prototipo de alarma comunitaria contará con códigos de identificación específicos asignados a cada una de las tres zonas que monitoreará la cámara de video. A cada usuario se le asignará un código según su ubicación, el mismo que será enviado vía SMS, que activará y desactivará la alarma, dicho código es único e intransferible entre zonas.

Las funciones principales del modem GSM serán: recibir los SMS para la activación, desactivación del prototipo y enviar SMS a los administradores del sistema indicando que se activó la alarma.

Para la comunicación entre la placa electrónica Arduino Uno y el modem GSM se requerirá de tres hilos; RX, TX y GND. Para la conexión se utilizará el cable de datos con conector DB9.

La conexión entre los dos dispositivos se la realizará por medio del puerto serial RS232 del modem hacia los pines 8 y 9 configurados como RX y TX respectivamente; para la comunicación serial se utilizarán los elementos que se muestran en el diagrama circuital de la figura 3.8.

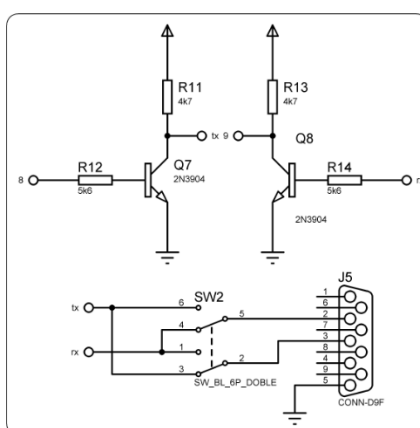


Figura 3.8 Diagrama circuital de la etapa de comunicación

Como se puede observar en el diagrama circuital de la figura 3.5 los transistores Q7 y Q8 (2N39004) son para amplificar las señales transmitidas y recibidas de la placa

electrónica Arduino Uno por medio del cable serial y con esto se asegurará el correcto funcionamiento de las siguientes etapas del prototipo de alarma comunitaria.

3.2.1.3.1 Códigos que la alarma interpretará para su ejecución

Los códigos aquí listados deberán ser ingresados en el modem GSM, con el objeto de que el sistema pueda decodificarlos, para ejecutar la acción correspondiente.

Cabe mencionar que los comandos deberán ser escritos por el usuario tal como se muestra en el listado, esto quiere decir que si el comando contiene letras minúsculas se escribirá con letras minúsculas y si está escrito con letras mayúsculas se deberá escribir con letras mayúsculas.

En la tabla 3.1 se muestra una lista de comandos que se ha seleccionado provisionalmente para las diferentes actividades que el sistema realizará.

Código o Comando	Acción a ejecutar	Descripción
iz	Activar	Activa el sistema y mueve la cámara hacia la izquierda
de	Activar	Activa el sistema y mueve la cámara hacia la derecha
ct	Activar	Activa el sistema y mueve la cámara hacia el centro
of	Apagar	Apaga el sistema

Tabla 3.1 Lista de comandos

3.2.1.4 Etapa de Salida

Para esta etapa se diseñará un circuito electrónico que será acoplado a la placa electrónica Arduino Uno que servirá para acoplar las placas electrónicas diseñadas que conforman la Etapa de Salida, la misma que está compuesta de cuatro etapas que se encuentran distribuidas como se muestra en el diagrama de bloques de la figura 3.9.

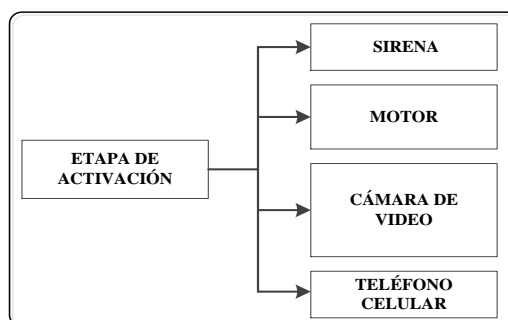


Figura 3.9 Diagrama de bloques de la etapa de salida

La Etapa de Salida activará las cuatro fases que la componen simultáneamente e instantáneamente una vez que el usuario envía un SMS al sistema con el código asignado a cada zona.

3.2.1.4.1 Sirena

Para la activación de esta etapa se utilizará el pin A2 conectado hacia un relé (RLY3) que se encargará de la activación y desactivación de la sirena.

Cuando la Etapa de Control recibe un comando de activación o desactivación descritos en la tabla 3.1, automáticamente envía un pulso en alto (1Logico – 5V) que cambiará de estado al relé y este activará o desactivará la sirena.

Para la activación de la sirena se utilizarán los elementos del diagrama circuital que se presentan en la figura 3.10.

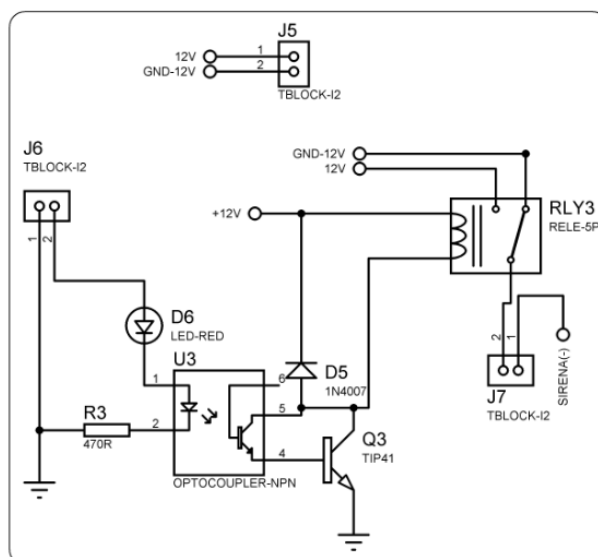


Figura 3.10 Diagrama circuital de la etapa de activación de la sirena

Como se observa en la figura 3.10, en esta etapa se utilizará un optoacoplador NPN que se encargará de aislar el circuito de activación de la sirena, con esto se evitará que retorne ruido a la placa electrónica Arduino Uno por el pin A2.

3.2.1.4.2 Motor

Para esta fase del sistema se utilizará un motor universal debido a su reducido tamaño, poco peso y puede ser alimentado con corriente alterna o con corriente continua.

Para el control del movimiento de la cámara se utilizarán los elementos del diagrama circuital que se presentan a continuación en la figura 3.11.

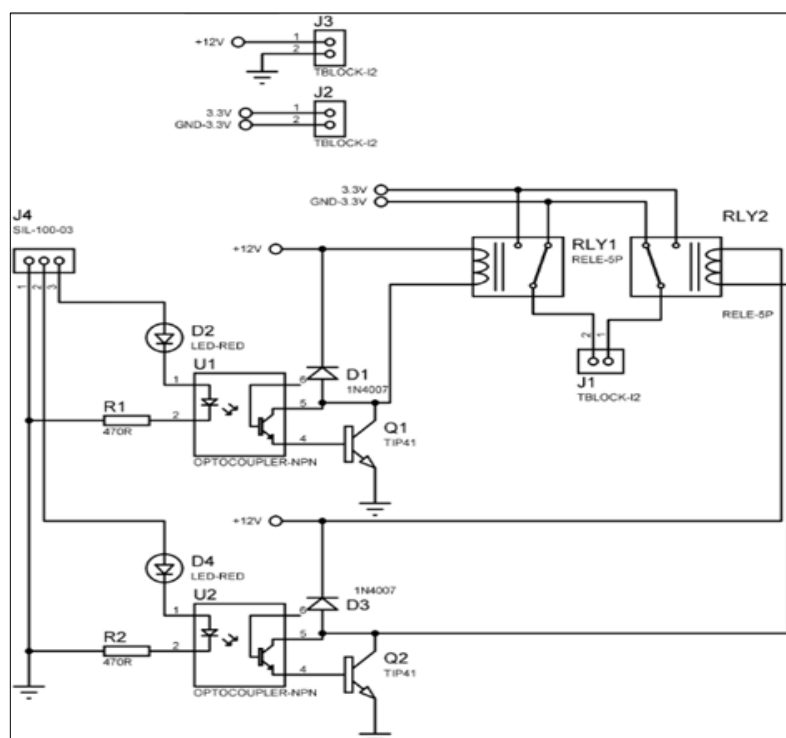


Figura 3.11 Diagrama circuital para el control del movimiento de la cámara de video

Para el control del movimiento del motor en el que se montará la cámara de video se utilizarán los pines: A4 y A5 donde se conectarán dos relés (RLY1 y RLY2) los mismos que servirán para mover el motor hacia la derecha, izquierda y centro, una vez recibido el comando de activación RLY1 se encargará de controlar el motor y RLY2 se encargará de controlar la batería interna del motor; adicionalmente se utilizarán dos fines de carrera normalmente abiertos que se instalarán en cada una de las posiciones a donde apuntará la cámara de video (derecha, izquierda) los que se encargarán limitar el movimiento del motor y se encontrarán conectados a los pines A0 y A1.

El motor se encontrará conectado a los pines “comunes” de los relés RLY1 y RLY2, los mismos que se encuentran protegidos por medio de dos optoacopladores NPN; con esto se aislará el circuito de activación y control del movimiento y se protegerá a la placa electrónica Arduino Uno.

3.2.1.4.3 Cámara de video

Para el diseño de esta etapa se utilizará una cámara de video marca SONY modelo DCR-HC42 debido a que tiene la posibilidad de poder ser controlada por medio de control remoto. El sistema microcontrolado tendrá incorporado un emisor IR conectado al pin 3 por donde se enviará los códigos de control hacia la video cámara. En la figura 3.12 se presenta el diagrama circuital de la etapa de activación del sistema de grabación de video.

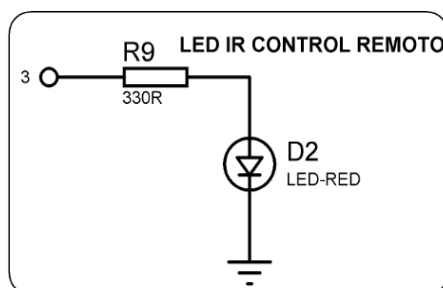


Figura 3.12 Emisor IR para la activación del sistema de grabación de video

Para el diseño de esta fase del sistema se utilizará el protocolo SIRC de SONY a 15 bits ya que la cámara utilizada trabaja bajo este formato.

El código utilizado para el control de inicio de grabación es 4C9D en hexadecimal.

Los códigos hexadecimales correspondientes a los botones del control remoto de la cámara de video se presentan a continuación en la tabla 3.2.

	Botón	Código Hexadecimal
	STAR/STOP	4C9D
	T	2C9B
	W	6C9B
	PHOTO	549C
	PLAY	59C
	REW	39C
	FF	D9C
	STOP	19C
PAUSE	99C	

Tabla 3.2 Botones del control remoto con su respectivo código hexadecimal

3.2.1.4.4 Teléfono celular

El teléfono celular será el encargado de enviar los SMS para la activación y desactivación del sistema los mismos que se guardarán como plantillas para que los usuarios no tengan que escribirlos cada vez que necesiten activar el sistema; además el teléfono celular recibirá los SMS de alerta para advertir a los usuarios que se activó la alarma.

El módulo GSM será configurado adecuadamente para que al momento de recibir el SMS, envíe automáticamente otro SMS de alerta a los usuarios registrados en el sistema, por lo que será necesario contratar un paquete de SMS para el correcto funcionamiento de esta etapa.

3.2.2 Diseño del software

Para el control de la alarma comunitaria en esta sección se diseñará el programa para el microcontrolador ATMEGA 328P cerebro del Arduino Uno y también se elaborarán las configuraciones que controlarán el modem GSM. Además, para entender el funcionamiento del sistema se realizarán diagramas de flujo de cada programa.

3.2.2.1 Diseño del software del Arduino Uno

El programa diseñado para esta etapa será el encargado de controlar el funcionamiento de la alarma. La figura 3.10 muestra el diagrama de flujo de esta fase.

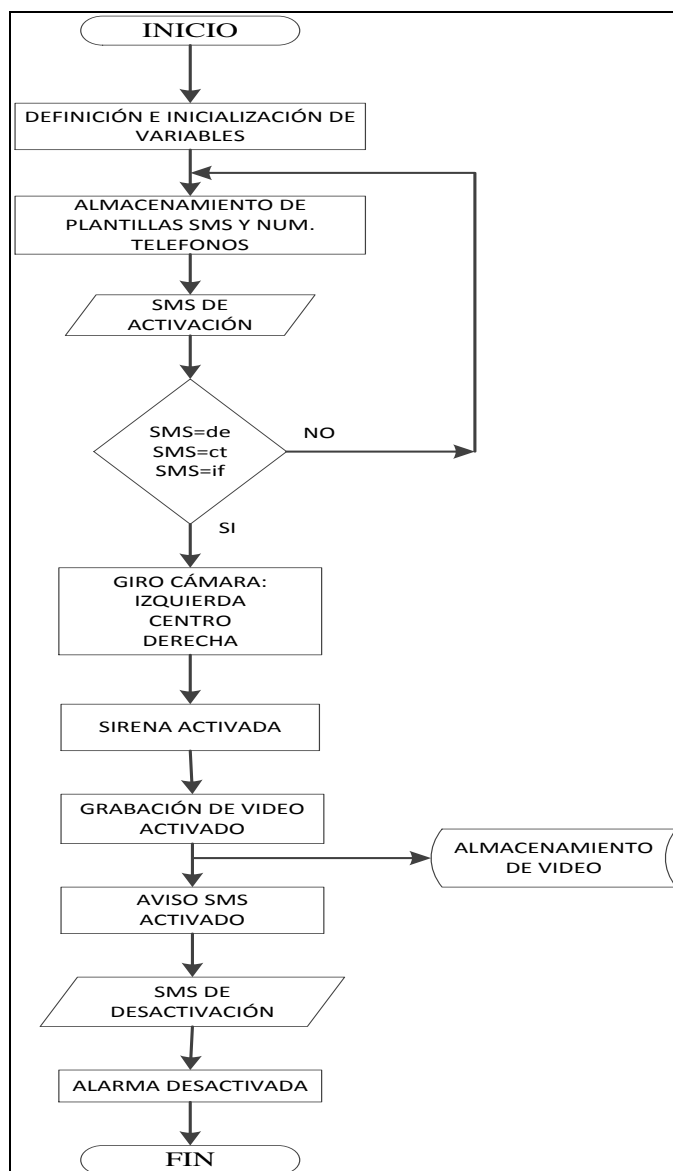


Figura 3.13 Diagrama de flujo para el diseño del software de la placa electrónica Arduino Uno

3.2.2.2 Configuración del modem GSM

La configuración elaborada en esta etapa se encargará de controlar la activación/desactivación del prototipo y del envío de SMS de alerta a los usuarios del sistema sobre la activación de la alarma. La figura 3.14 muestra el diagrama de flujo de esta fase.

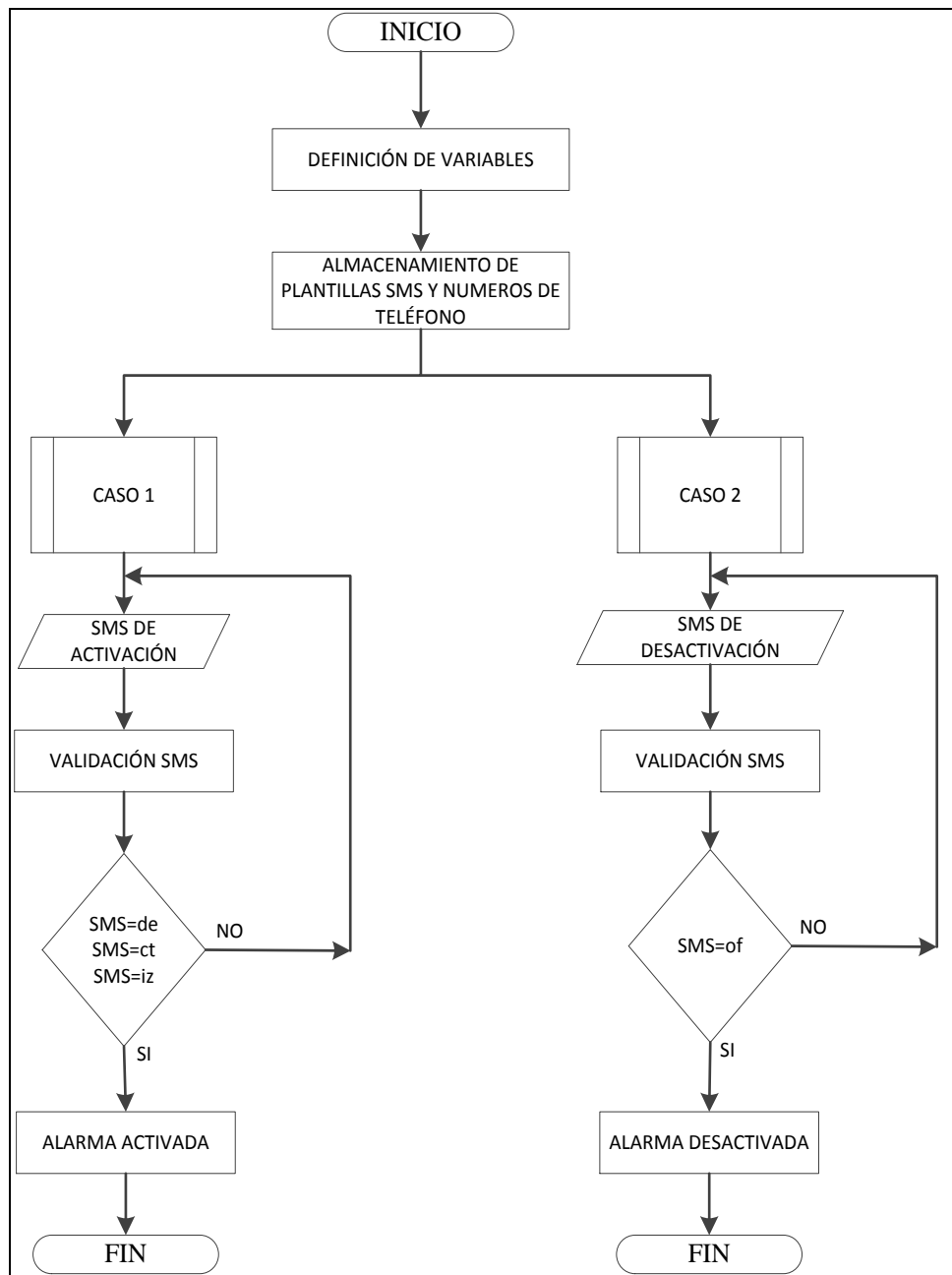


Figura 3.14 Diagrama de flujo de la configuración del modem GSM

3.2.3 Diseño mecánico

En la figura 3.15 se muestra la caja contenedora del sistema prototipo con sus respectivas dimensiones, realizada en QCAD.

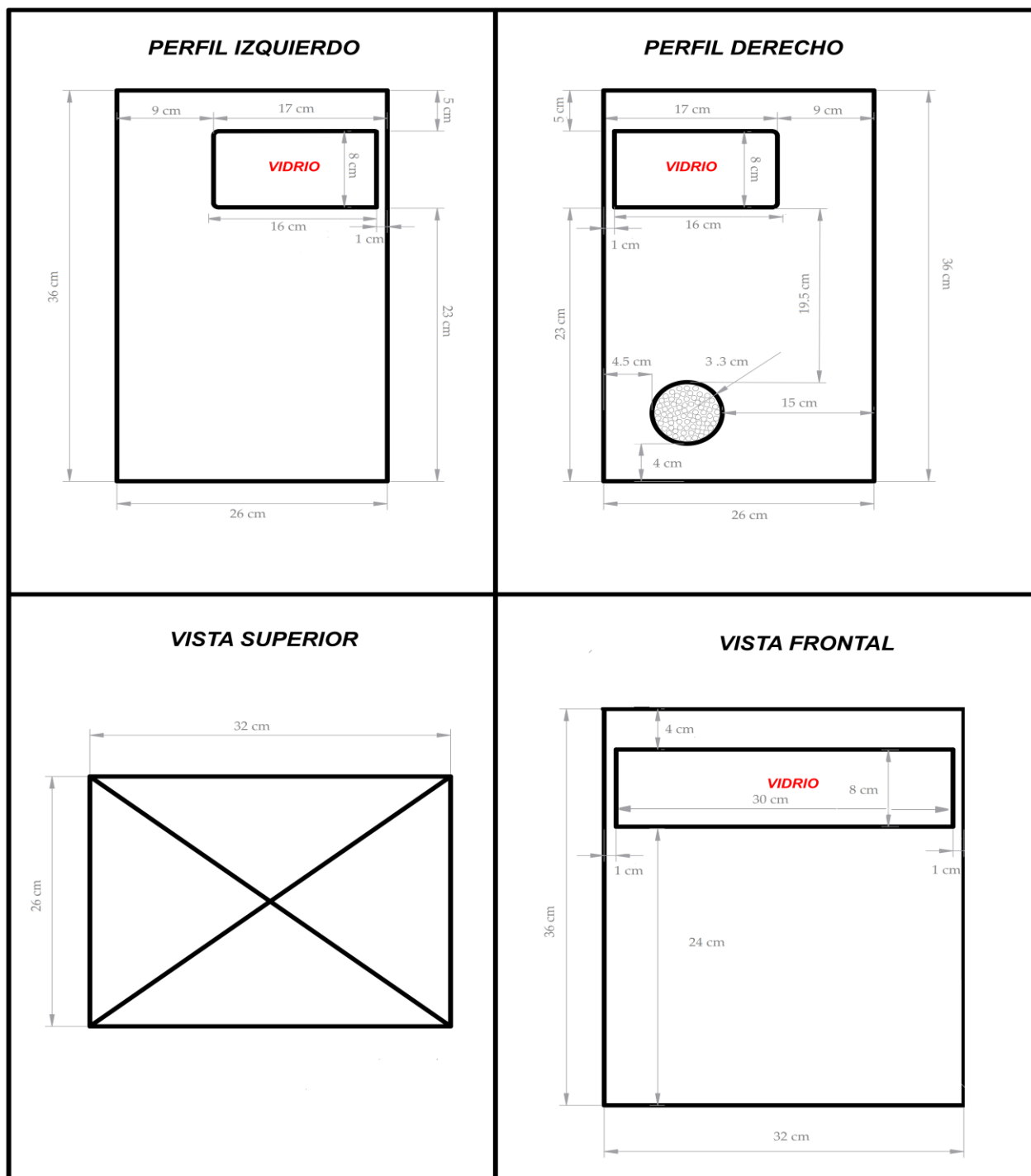


Figura 3.15 Dimensiones de caja contenedora de la alarma

Como se puede apreciar en la figura anterior, se instalarán vidrios a los tres lados de la caja para visibilidad de la cámara de video.

En la figura 3.16 se puede apreciar el interior de la caja, con sus dimensiones, además se puede observar una pequeña caja de madera donde se colocarán los circuitos electrónicos diseñados para el control del prototipo.

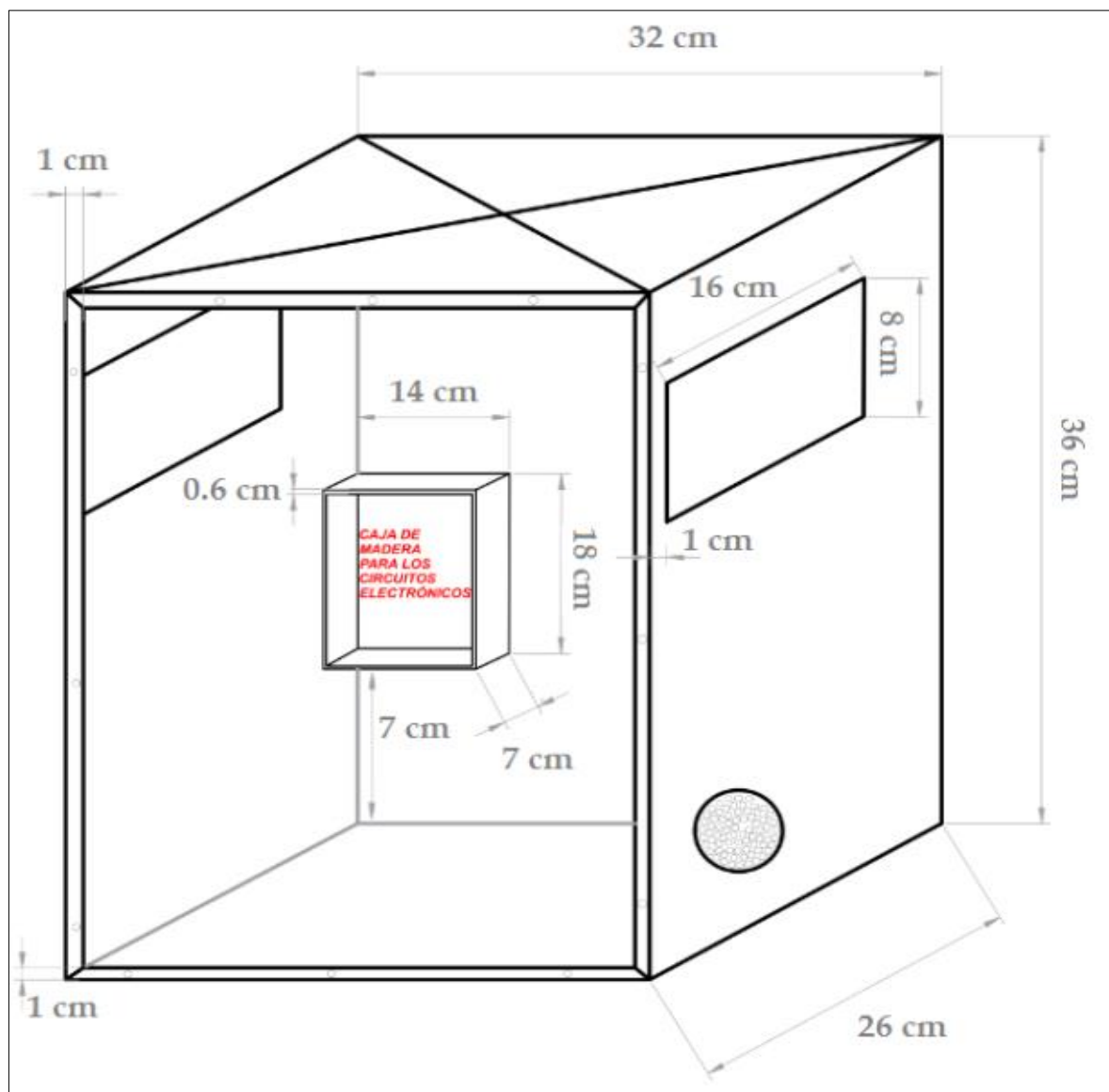


Figura 3.16 Interior de la caja contenedora

La figura anterior muestra las características internas de la caja, la misma que tendrá doblés de 1 cm de ancho alrededor, donde se colocarán empaques de caucho para

aislarla de la humedad y proteger su interior de posibles filtraciones de líquidos, y con esto se asegurará el correcto funcionamiento de la alarma.

3.2.4 Montaje

Diseñados los circuitos en el simulador de circuitos ISIS¹⁴ de PROTEUS, se procederá a armar en el protoboard el diagrama circuital del prototipo para verificar su correcto funcionamiento.

3.2.4.1 Montaje de hardware

En el protoboard se procederá a: Identificar la polarización de cada elemento, es decir cual terminal es tierra y cual terminal es Vcc, con esto se evitarán posibles daños en los elementos o que no funcionen correctamente. Se ajustarán los valores de voltaje para la alimentación de la placa electrónica Arduino Uno, así como también el voltaje para la activación de la sirena y el modem GSM.

En la figura 3.17 se puede apreciar el circuito receptor implementado en el protoboard como para la obtención de los códigos hexadecimales correspondientes a los botones del control remoto de la cámara de video.

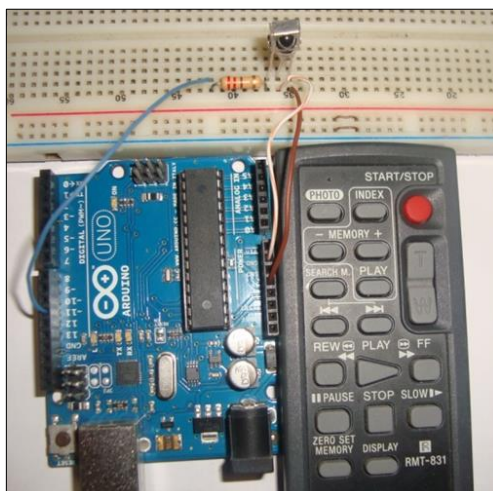


Figura 3.17 Circuito receptor IR¹⁵

¹⁴ ISIS Intelligent Schematic Input System (Sistema de Enrutado de Esquemas Inteligente)

¹⁵ <http://www.davidmiguel.com/arduino/dia-3-sensor-ir/>

La figura muestra un circuito de recepción IR implementado con el receptor PC1838B, el receptor IR tiene una resistencia de 220 Ohm conectada en serie al pin 11 de la placa electrónica Arduino Uno por donde se recibirán los pulsos provenientes del control remoto, los otros dos pines del receptor se encuentran conectados a GND y 5V respectivamente.

En la figura 3.18 se puede apreciar el diagrama circuital general de todo el prototipo implementado en el protoboard.

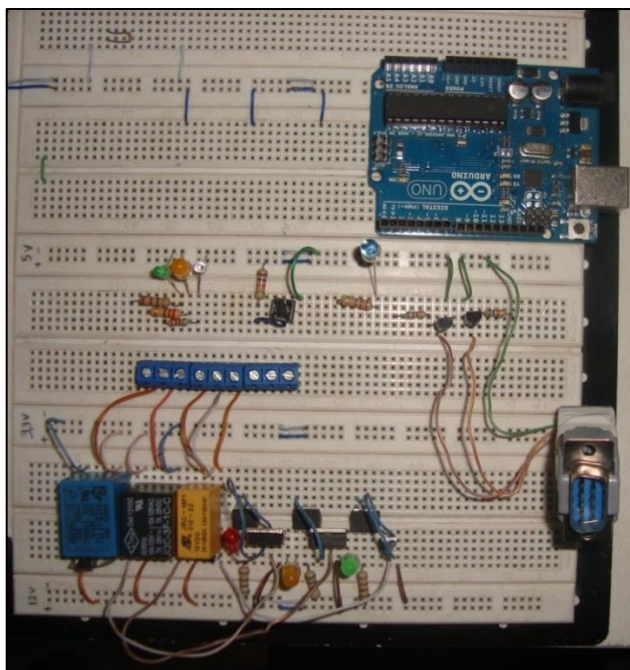


Figura 3.18 Circuito general implementado en el protoboard

Corregidos los errores en el protoboard se procederá a realizar el circuito impreso en la placa de baquelita, para lo cual se utilizará el software PROTEUS, con sus aplicaciones en ISIS para esquemas y simulación, y ARES para la creación de circuitos impresos.

Primero se ingresó en ISIS los circuitos individuales, para ello se debió crear algunos elementos como el Arduino Uno y los relés de 5 pines, para luego, mediante le aplicación ARES crear las pistas de cada una de las placas elaboradas.

El diagrama circuital final del prototipo realizado en ISIS se muestran en la figura 3.19.

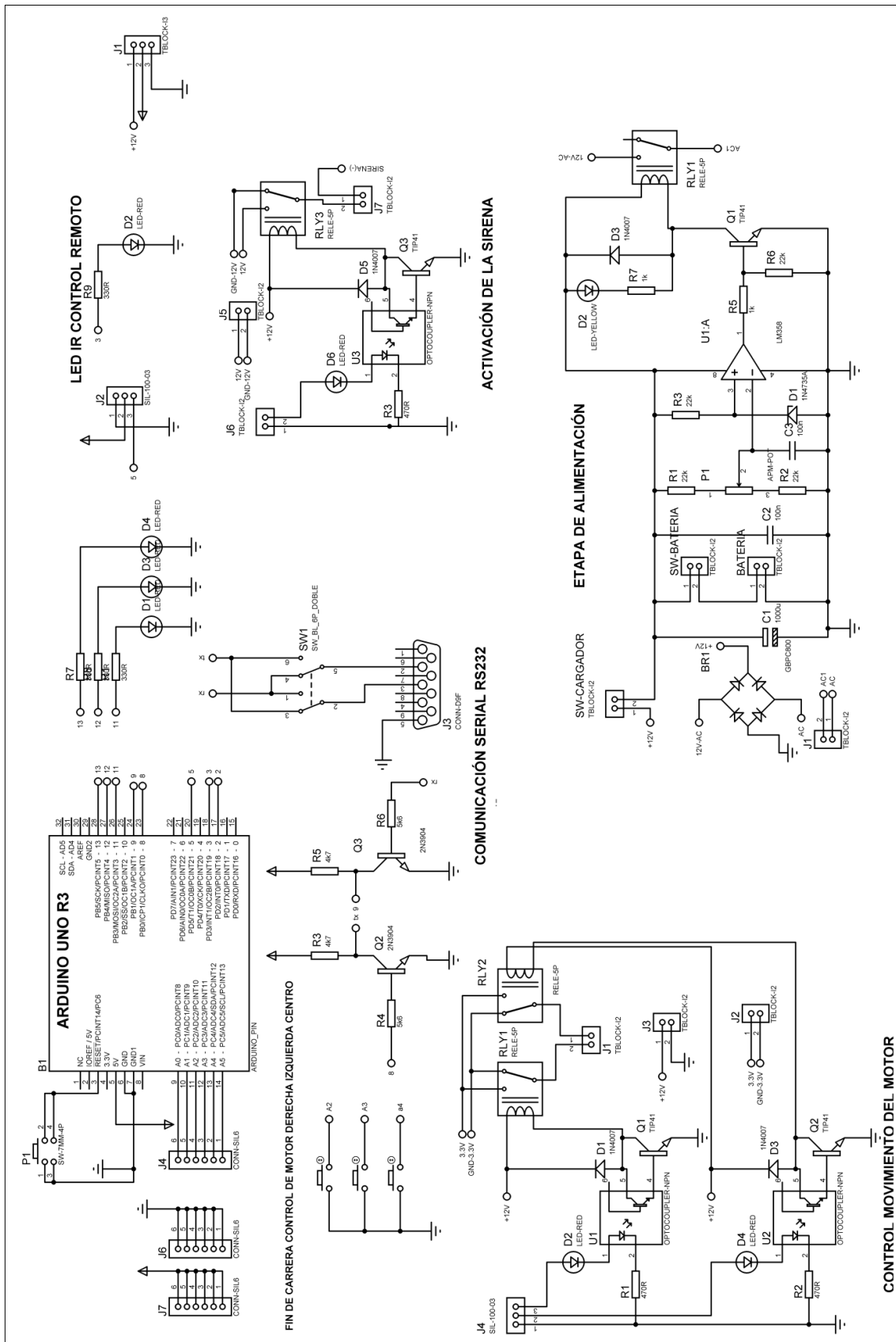


Figura 3.19 Diagrama circuital final implementado en ISIS

3.2.4.2 Montaje del software

El programa iniciará con las configuraciones iniciales, las cuales se encargarán de la configuración de los módulos internos del microcontrolador, tales como, la frecuencia del oscilador interno, configuración de la comunicación serial, y definición de las variables internas que se utilizarán en la implementación del programa, además se definirán las subrutinas que se usarán.

3.2.4.2.1 Programación con software Arduino

Para empezar a utilizar el entorno Arduino es necesario instalar el programa en un computador tomando en cuenta el sistema operativo con el que se va a trabajar, es posible descargar el programa de la página oficial de Arduino www.arduino.cc, donde existen códigos, librerías, hardware existentes, entre otras.

Para la elaboración del presente proyecto se escogió el software Arduino 1.0.5 para Windows, el cual proporciona una interfaz y compatibilidad con Arduino Uno.

Una vez instalado el programa se procede a conectar la placa electrónica Arduino Uno al computador para que sea asignado a un puerto de comunicación (COM), y se seleccionará la placa a ser utilizada y el puerto (COM) asignado como se muestra en las figuras 3.20 y 3.21.

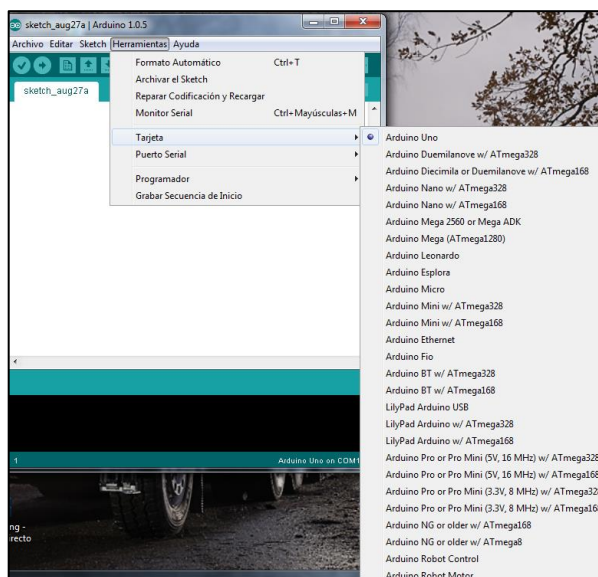


Figura 3.20 Selección de placa Arduino Uno

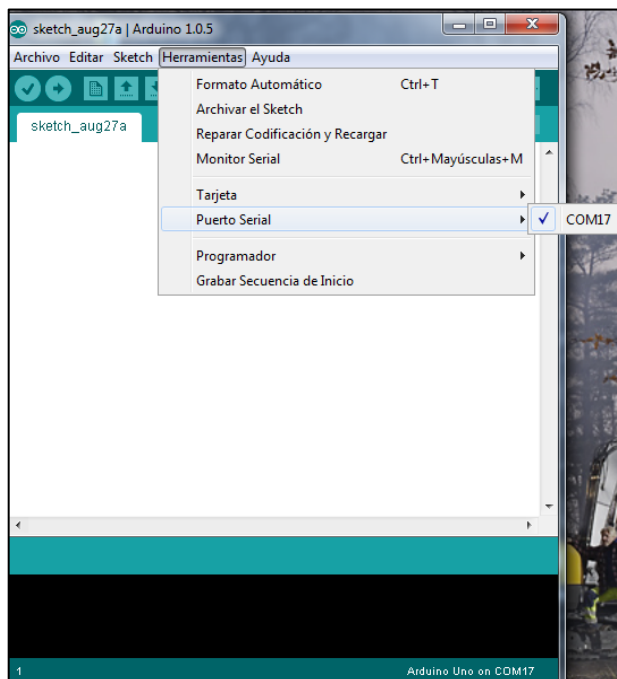


Figura 3.21 Selección del puerto serial asignado

Este proceso es necesario ya que si no se reconoce el puerto serial se tendrán problemas al momento de grabar los programas en la placa electrónica Arduino Uno.

El programa realizado analizará las sentencias donde se incluirán los comandos AT necesarios para configurar el módulo GSM, para que trabaje en modo mensajes de texto y la velocidad de transmisión de información a 9600 baudios.

Para la comunicación del módulo GSM con la placa electrónica Arduino Uno se utilizarán los pines 8 y 9 configurados como TX y RX respectivamente, en esta fase del software se utilizará la librería *SoftwareSerial* incluida en la versión de Arduino utilizada.

Para el control del sensor IR se utilizará el pin 3 de la placa electrónica Arduino Uno, para el correcto funcionamiento de esta etapa del software es necesario descargar y agregar al programa Arduino la librería *IRremote* para el control mediante infrarrojos de la cámara de video; la forma de seleccionar las librerías mencionadas anteriormente se muestra en la figura 3.22.

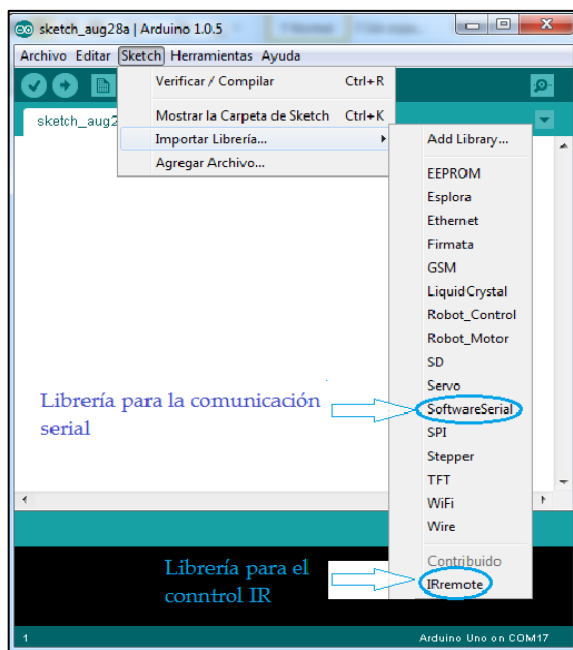


Figura 3.22 Librerías necesarias para ejecutar el programa

3.2.4.2.2 Programa para la obtención de los códigos hexadecimales

A continuación se presenta el programa cargado en la placa electrónica Arduino Uno para la recepción de los códigos hexadecimales del control remoto de la cámara de video se presenta en la figura 3.23.

```
#include <IRremote.h> //Insertamos la libreria IR
const int RECV_PIN = 11; //Asignamos al pin 11 la variable RECV_PIN tipo int
IRrecv irrecv(RECV_PIN); //Creamos la variable receptora
decode_results results; //Recibe el codigo de IR y devuelve true si se a recibido codigo
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); //Inicia el proceso de recepción
  irrecv.blink13(true);
}
void loop() {
  if (irrecv.decode(&results)) { //Cuando recibe un codigo lo almacena el results
    if (results.decode_type == NEC) {
      Serial.print("NEC: "); //Imprime en pantalla results
    } else if (results.decode_type == SONY) {
      Serial.print("SONY: ");
    } else if (results.decode_type == RC5) {
      Serial.print("RC5: ");
    } else if (results.decode_type == RC6) {
      Serial.print("RC6: ");
    } else if (results.decode_type == UNKNOWN) {
      Serial.print("UNKNOWN: ");
    }
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}
```

Figura 3.23 Programa implementado para la recepción IR¹⁶

¹⁶ http://www.pjrc.com/teensy/td_libs_IRremote.html;

Una vez cargado el programa en la placa electrónica Arduino Uno se obtuvieron los siguientes resultados presentados en la figura 3.24.

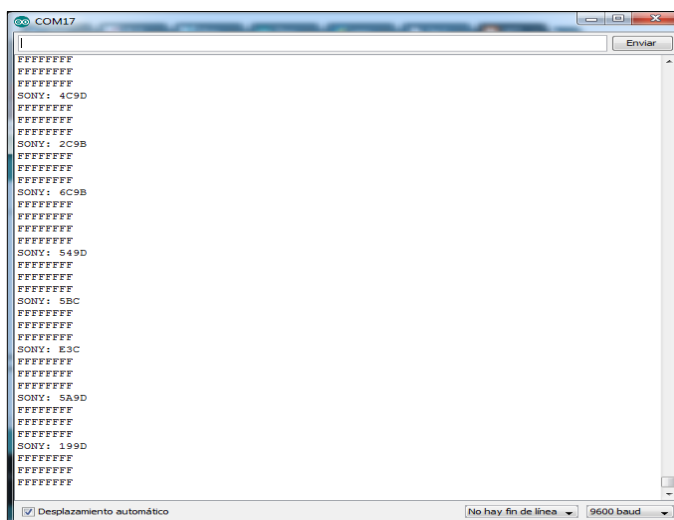


Figura 3.24 Códigos hexadecimales obtenidos del circuito receptor IR

3.2.4.2.3 Diseño de la programación del sistema de control del prototipo

En esta etapa se va a realizar el programa general que controlará el sistema prototipo de alarma comunitaria, el mismo que se muestra a continuación:

```
#include <SoftwareSerial.h>           //Librería para la comunicación serial
#include <IRremote.h>                 //Librería para la © IR
SoftwareSerial mySerial(9, 8); // RX, TX //Configuración de los pines TX/RX

String datos, 44ínea44=""; codigo2;   //Creación variables tipo String
char inChar,inChar2;                  //Creación de variables tipo char
Irsend irsend;                        //Creación de variable tipo Irsend
#define rele1 A4                       //Defino el nombre del pin
#define rele2 A5                       //Defino el nombre del pin
#define fin1 A0                        //Defino el nombre del pin
#define fin2 A1                        //Defino el nombre del pin
#define sirena A2                      //Defino el nombre del pin

boolean ffin1=0;                      //Creo una variable tipo boolean y la inicializo con 0
boolean ffin2=0;                      //Creo una variable tipo boolean y la inicializo con 0
int TMD=10;                           //Creo una variable tipo int y le doy el valor de 10ms
int TMI=10;                           //Creo una variable tipo int y le doy el valor de 10ms
int x=0;                               //Creo una variable tipo int y la inicializo con 0
boolean flf=0;                        //Creo una variable tipo boolean y la inicializo con 0
boolean frt=0;                        //Creo una variable tipo boolean y la inicializo con 0
```

```

boolean fmd=0;           // Creo una variable tipo boolean y la inicializo con 0
int xd;                  // Creo una variable tipo int
boolean fser=0;         //Creo una variable tipo boolean y la inicializo con 0
String otro;            //Creación variables tipo String
int cont;                // Creo una variable tipo int

void setup()            //Lazo para inicializar variables
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);   //Establesco la velocidad de comunicación

  pinMode(11,OUTPUT);   //Inicializo el pin como salida
  pinMode(12,OUTPUT);   // Inicializo el pin como salida
  pinMode(13,OUTPUT);   // Inicializo el pin como salida)

  pinMode(rele1,OUTPUT); // Inicializo el pin como salida
  pinMode(rele2,OUTPUT); // Inicializo el pin como salida
  pinMode(sirena,OUTPUT); // Inicializo el pin como salida

  pinMode(fin1,INPUT);  // Inicializo el pin como salida
  pinMode(fin2,INPUT);  // Inicializo el pin como salida
  digitalWrite(fin1,HIGH); //Activo las pull-up del pin
  digitalWrite(fin2,HIGH); // Activo las pull-up del pin

  digitalWrite(11,HIGH); // Activo las pull-up del pin
  digitalWrite(12,HIGH); // Activo las pull-up del pin
  digitalWrite(13,HIGH); // Activo las pull-up del pin
  delay(800);            //Retardo de 800 ms
  digitalWrite(11,LOW);  // Activo las pull-up del pin
  digitalWrite(12,LOW);  // Activo las pull-up del pin
  digitalWrite(13,LOW);  // Activo las pull-up del pin
  delay(400);           // Retardo de 800 ms

  digitalWrite(sirena,HIGH); //Activo la sirena
  delay(100);             // Retardo de 100 ms
  digitalWrite(sirena,LOW); //Apago la sirena
  delay(100);            // Retardo de 100 ms
  digitalWrite(sirena,HIGH); //Activo la sirena
  delay(100);            // Retardo de 100 ms
  digitalWrite(sirena,LOW); // Apago la sirena

  Serial.println("Iniciando..."); // Imprimo por el Puerto serial
  delay(500);              //Retardo de 500 ms
  // set the data rate for the SoftwareSerial port
  mySerial.begin(2400);    //Establece la velocidad de comunicación del Puerto serial

  mySerial.println("AT"); //Configuro el modem e imprimo en el Puerto serial

```

```

delay(2000);           //Retardo de 2000 ms
mySerial.println("AT+CNMI=1,2,0,0,0");//Configuración del modem
delay(4000);           //Retardo de 4000 ms
mySerial.println("AT+CMGF=1");// Comando del modem
delay(4000);           //Retardo de 4000 ms
mySerial.print("AT+CMGS=");mySerial.print(char(34));
mySerial.print("0959504317");
mySerial.print(char(34));mySerial.print(char(10));mySerial.print(char(13));
delay(2000);           //Retardo de 2000 ms
mySerial.print("Modem listo...");mySerial.print(char(26));mySerial.print(char(10));
mySerial.print(char(13));
delay(2000);           //Retardo de 2000 ms
//mySerial2.begin(9600);

}

void loop()             // Lazo principal de programa
{
  while (mySerial.available()) //Si tengo algún dato en la variable entro al while
  {
    inChar = (char)mySerial.read(); //Leo lo que tengo en la variable inChar
    datos += inChar; //asigno a la variable datos lo que tengo en inChar
  }

  if(datos!="")         //Si datos en diferente de nulo entro al if
  {
    digitalWrite(11,1); //Envío un 1L por el pin 11
    digitalWrite(12,1); // Envío un 1L por el pin 12
    digitalWrite(13,1); // Envío un 1L por el pin 13
    delay(1000);         //Retardo de 1000 ms
    digitalWrite(11,0); // Envío un 0L por el pin 11
    digitalWrite(12,0); // Envío un 0L por el pin 12
    digitalWrite(13,0); // Envío un 0L por el pin 13
    delay(1000);         //Retardo de 1000 ms
    Serial.println(datos); //Imprimo por el puerto serial lo que tengo en
                           // datos
    otro=datos.substring(50,60); //Doy a la variable otro un espacio de 10 bits en
                                  // memoria interna
    Serial.println(otro); //Imprimo la variable otro por el puerto serial
    Serial.println(datos.length()); //Imprimo la longitud de la variable datos
    Serial.println(otro.length()); // Imprimo la longitud de la variable otro
    delay(200);           //Retardo de 200 ms
    otro.trim();          //Elimino los espacios en blanco de la variable otro

    if(otro == "de")     //Si la variable otro es igual a de entro a la rutina
    {
      digitalWrite(sirena,HIGH); //Activo la sirena
    }
  }
}

```

```

if(otro == "de") // Si la variable otro es igual a de entro a la rutina
{
  digitalWrite(12,1); // Envío un 1L por el pin 12
  delay(1000); // Retardo de 1000 ms
  digitalWrite(12,0); // Envío un 0L por el pin 12
  delay(1000); // Retardo de 1000 ms

  ffin2=digitalRead(fin2); //Leo el estado del fin de carrera 2
  Serial.println(ffin2); //Imprimo el estado del fin de carrera 2

  while(ffin2==1) //Si fin de carrera 2 es igual a uno entro a la
  subrutina
  {
    delay(20); //Retardo de 20 ms
    digitalWrite(11,1); // Envío un 1L por el pin 11
    digitalWrite(sirena,HIGH); //Envío un 1L por el pin sirena
    digitalWrite(rele1,1); //Envío un 1l por el pin rele1
    digitalWrite(rele2,0); //Envío un 0L por el pin rele2
    delay(TMD); //Doy un retardo de 10 ms

    digitalWrite(rele1,0); //Apago el relé 1
    digitalWrite(rele2,0); //Apago el relé 2
    digitalWrite(11,0); //Apago el pin 11
    delay(20); //Retardo de 20 ms
    ffin2=digitalRead(fin2); //Asigno a ffin2 el estado de la variable fin2
    Serial.println(ffin2); //Imprimo la varaible ffin2
  }
  datos=""; //Borro la variable datos

  for (int i = 0; i < 5; i++) //Envío 5 tramas de código
  {
    irsend.sendSony(0x4C9D, 15); // Envío el código de on/off de la cámara de video
    delay(30); //Retardo de 30 ms
  }
  Serial.println("recibido ok"); //Imprimo por el puerto serial

  mySerial.print("AT+CMGS=");mySerial.print(char(34));
  mySerial.print("0959504317");
  mySerial.print(char(34));mySerial.print(char(10));mySerial.print(char(13));
  delay(2000);
  mySerial.print("Sensor Derecha se activo alarma encendida");
  mySerial.print(char(26));
  mySerial.print(char(10));mySerial.print(char(13));
  delay(2000);
  Serial.println("Acciono Fin Derecho y paro Motor a derecha");
}

```

```

if(otro == "iz") //Si otro igual a iz entro a la subrutina
{
  digitalWrite(sirena,HIGH); //Activo la sirena
}

if(otro == "iz") //Si otro igual a if entro a la subrutina
{
  digitalWrite(11,1); //Activo pin 11
  delay(1000); //Retardo de 1000 ms
  digitalWrite(11,0); //Apago pin 11
  delay(1000); //Retardo de 1000 ms
  ffin1=digitalRead(fin1); //Asigno a ffin1 el valor de la variable fin1

while(ffin1==1) // Si fin de carrera 1 es igual a uno entro a
la subrutina
{
  delay(20); //Retardo de 20 ms
  digitalWrite(11,1); //Activo el pin 11
  digitalWrite(sirena,HIGH); //Activo el pin sirena
  digitalWrite(rele1,0); //Desactivo pin rele1
  digitalWrite(rele2,1); //Activo pin rele2
  delay(TMI); //Doy un retardo de 10 ms

  digitalWrite(rele1,0); //Desactivo el pin rele1
  digitalWrite(rele2,0); //Desactivo el pin rele2
  digitalWrite(11,0); //desactivo el pin 11
  delay(20); //Retardo de 20 ms
  ffin1=digitalRead(fin1); //Asigno a ffin1 el estado de la variable fin1
  Serial.println(ffin1); //Imprimo la variable ffin1
}
datos=""; //Borro la variable datos

for (int i = 0; i < 5; i++) // Envío 5 tramas de código
{
  irsend.sendSony(0x4C9D, 15); // Envío el código de on/off de la cámara de video
  delay(30); //Retardo de 30 ms
}
Serial.println("recibido ok"); //Imprimo por el Puerto serial
mySerial.print("AT+CMGS=");mySerial.print(char(34));
mySerial.print("0959504317");
mySerial.print(char(34));mySerial.print(char(10));mySerial.print(char(13));
delay(2000);
mySerial.print("Sensor Izquierda se activo alarma encendida");
mySerial.print(char(26));mySerial.print(char(10));mySerial.print(char(13));
delay(2000);
Serial.println("Acciono Fin Izquierdo y paro Motor a Izquierda");
}

```

```

if(otro == "ct") //Si otro es igual a ct entro a la subrutina
{
digitalWrite(sirena,HIGH); //Activo el pin sirena
}

if(otro == "ct") //Si otro es igual a ct entro a la subrutina
{
digitalWrite(13,1); //Activo el pin 13
delay(1000); //Retardo de 1000 ms
digitalWrite(13,0); //Desactivo el pin 13
delay(1000); //Retardo de 1000 ms
ffin2=digitalRead(fin2); //Asigno a ffin2 el estado de la variable fin2
ffin1=digitalRead(fin1); // Asigno a ffin1 el estado de la variable fin1

if(otro == "ct" && ffin2==0) //Si otro es igual a ct y ffin2 es igual
a 0 entro a la subrutina
{
for(xd=0;xd<57;xd++) //Envío 57 pulsos por el pin rele2
{
Serial.println(xd); //Imprimo la variable xd
delay(20); //Retardo de 20 ms
digitalWrite(11,1); //Activo el pin 11
digitalWrite(sirena,HIGH); //Activo el pin sirena
digitalWrite(rele1,0); //Apago el pin rele1
digitalWrite(rele2,1); //Activo el pin relé2
delay(TMD); //Retardo de 10 ms

digitalWrite(rele1,0); //Apago el pin rele1
digitalWrite(rele2,0); //Apago el pin rele2
digitalWrite(11,0); //Apago el pin 11
delay(20); //Retardo de 20 ms
}
datos=""; //Limpio la variable datos
ffin1=digitalRead(fin1); //Asigno a ffin1el valor de la variable fin1
ffin2=digitalRead(fin2); // Asigno a ffin1el valor de la variable fin1
Serial.println("Motor al centro desde Derecha"); //imprimo por el puerto serial
}

if(otro == "ct" && ffin1==0) //Si otro es igual a ct y ffin2 es igual
a 0 entro a la subrutina
{
for(xd=0;xd<52;xd++) //Envío 57 pulsos por el pin rele2
{
Serial.println(xd); //Imprimo la variable xd
delay(20); //Retardo de 20 ms
digitalWrite(11,1); //Activo el pin 11
digitalWrite(sirena,HIGH); //Activo el pin sirena
digitalWrite(rele1,1); //Activo el pin rele1
}
}

```



```

    digitalWrite(rele2,0);           //Apago el pin rele2
    delay(TMD);                     //Retardo de 10 ms
    digitalWrite(rele1,0);         //Apago el pin rele1
    digitalWrite(rele2,0);         //Apago el pin rele2
    digitalWrite(11,0);            //Apago el pin 11
    delay(20);                      //Retardo de 20 ms
}
datos="";                          //Limpio la variable datos
ffin1=digitalRead(fin1);          //Asigno a ffin1 el valor de la variable fin1
ffin2=digitalRead(fin2);          //Asigno a ffin2 el valor de la variable fin2
Serial.println("Motor al centro desde Izquierda");
}

for (int i = 0; i < 5; i++)        //Envío 5 tramas de datos
{

    irsend.sendSony(0x4C9D, 15);    // Envío el código de on/off de la
                                   cámara de video
    delay(30);
}
Serial.println("recibido ok");      //Imprimo pr el Puerto serial
mySerial.print("AT+CMGS=");mySerial.print(char(34));
mySerial.print("0959504317");mySerial.print(char(34));
mySerial.print(char(10));mySerial.print(char(13));
delay(2000);
mySerial.print("Sensor Central se activo alarma encendida");
mySerial.print(char(26));mySerial.print(char(10));mySerial.print(char(13));
delay(2000);
Serial.println("Motor al Centro Alarma Encendida"); //Imprimo por el puerto serial
}

if(otro == "of")                   //Si otro es igual a of entro a la subrutina
{
    digitalWrite(13,1);             //Activo el pin 13
    delay(1000);                    //Retardo de 1000 ms
    digitalWrite(13,0);             //Apago el pin 13
    delay(1000);                    //Retardo de 1000 ms
    digitalWrite(sirena,0);         //Apago el pin sirena

    for (int i = 0; i < 5; i++)      //Envío 5 tramas de datos
    {
        irsend.sendSony(0x4C9D, 15); // Envío el código de on/off de la
        cámara de video

        delay(30);                  //Retardo de 30 ms
    }
    Serial.println("recibido ok");   //Imprimo por el Puerto serial
    mySerial.print("AT+CMGS=");mySerial.print(char(34));
}

```

```

mySerial.print("0959504317");mySerial.print(char(34));
mySerial.print(char(10));mySerial.print(char(13));
delay(2000);
mySerial.print("Mensaje of recibido alarma apagada");
mySerial.print(char(26));mySerial.print(char(10));
mySerial.print(char(13));
delay(2000);
Serial.println("Alarma Apagada");           //Imprimo por el puerto serial
}
datos="";                                   //Limpio la variable datos
}
cont++;                                     //Incremento la variable cont
if(cont>20)                                 //La variable cont va aumentanto hasta 20
{
  Serial.println("Limpio buffer");          //Imprimo por el Puerto serial
  datos = "";                               //Limpio la variable datos
  Serial.println();                          //Imprimo un salto de línea
  cont=0;                                    //Limpio la variable cont
}
delay(1000);                                //Retardo de 1000 ms
}

```

3.2.5 Montaje de la parte mecánica

Para esta fase del sistema se propuso la utilización de materiales como madera, acrílico y metal.

Luego de pruebas realizadas a estos tres materiales se llegó a demostrar que:

- La madera no resiste a los cambios de clima y su tiempo de vida se ve reducido cuando se encuentra instalada a la intemperie ya que debido a la humedad y al calor envejece con rapidez y no proporciona la seguridad que requiere la alarma.
- El acrílico puede ser instalado a la intemperie ya que el material con el que se encuentra construido es muy resistente a los cambios de clima, pero es muy propenso a roturas debido a golpes y su precio es muy elevado por lo que no se tomó en cuenta a este tipo de material para la elaboración de la caja.
- Por último se realizaron pruebas en metal llegando a la conclusión que es el mejor material para la construcción de la caja ya que soporta los cambios bruscos del clima, es resistente a golpes, su tiempo de vida útil es prolongado, puede ser instalado a la intemperie y su precio de adquisición es bajo; por lo que

la caja será elaborada en metal tipo tol de 1 mm de espesor ya que su peso es reducido y brinda la protección y seguridad requeridas para la protección del prototipo de alarma comunitaria.

Luego haber seleccionado el material adecuado para la elaboración de la caja y con los diagramas mecánicos realizados en QCAD se procederá a realizar la implementación de la caja metálica con las dimensiones requeridas. En las figuras 3.25 y 3.26 se pueden apreciar el aspecto final de la caja contenedora de la alarma.



Figura 3.25 Parte interior de la caja



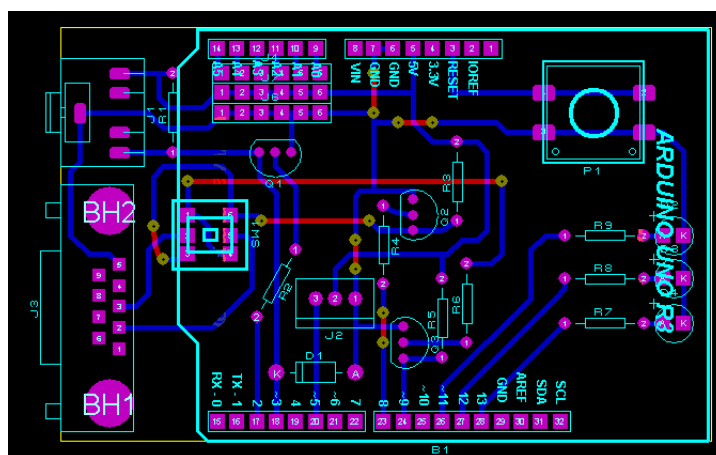
Figura 3.26 Vista frontal de la caja

3.3 Implementación del sistema prototipo de Alarma Barrial

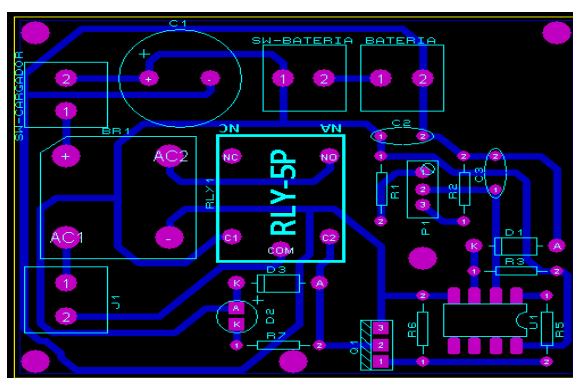
Esta etapa comprende la elaboración de las placas electrónicas y la implementación definitiva del prototipo en el sitio elegido.

3.3.1 Implementación del hardware

Una vez realizados los diagramas circuitales de cada etapa del prototipo se procedió a realizar los correspondientes circuitos impresos o PCB en ARES para ser trasladados a la baquelita, los mismos que se muestran a continuación en la figura 3.27 (a) PCB Etapa de Control, (b) PCB de la fuente de alimentación, (c) PCB de la Etapa de Control del motor y (d) PCB Etapa de Control de activación de la sirena.



(a)



(b)

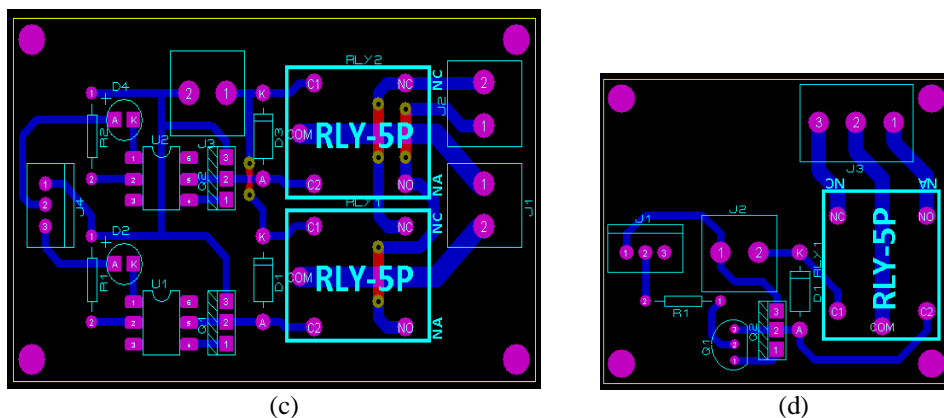
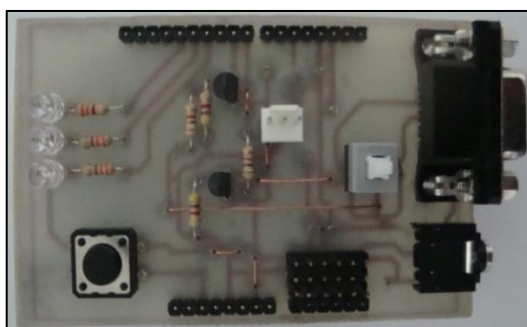


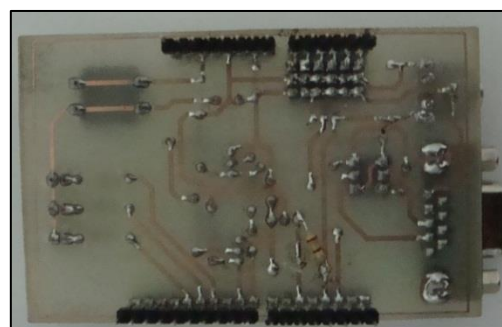
Figura 3.27 PCB correspondientes a cada etapa del prototipo

Luego de realizar los PCB de cada una de las etapas que comprenden el prototipo se procedió a pasarlos a la baquelita; en la figura 3.28 (a) placa electrónica de la etapa de control, (b) placa electrónica de la fuente de alimentación, (c) placa electrónica del control del movimiento del motor y (d) placa electrónica de la activación de la sirena se pueden apreciar las placas electrónicas finales del prototipo.

(a)



(a.1) Vista superior

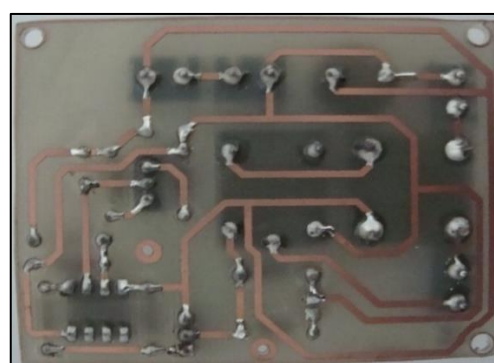


(a.2) Vista inferior

(b)



(b.1) Vista superior

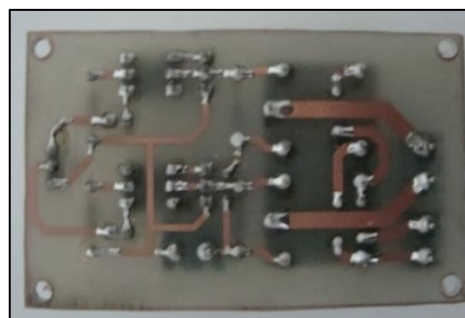


(b.2) Vista inferior

(c)



(c.1) Vista superior



(c.2) Vista inferior

(d)



(d.1) Vista superior



(d.2) Vista inferior

Figura 3.28 Circuitos finales correspondientes a cada etapa del prototipo

Una vez implementados los circuitos electrónicos por separado se procedió a instalarlos en la caja contenedora la misma que fue hecha de madera para aislar a las placas electrónicas de posibles cortocircuitos. En la figura 3.29 se puede observar el montaje final de los circuitos electrónicos.



Figura 3.29 Montaje final de los circuitos electrónicos

Luego de esto se procedió a instalar los elementos que componen el prototipo en la caja metálica, como se muestra en la figura 3.30.



Figura 3.30 Montaje del sistema en la caja metálica

3.3.2 Implementación del sistema

Luego de implementar los diferentes componentes del sistema prototipo en la caja metálica se procedió a instalarla en un poste público ubicado en las calles Vicente Ferrer y Pucará del sector elegido ya que este se encuentra en un sitio estratégico y tiene una amplia visibilidad hacia las tres zonas que serán monitoreadas por la cámara de video. En la en la figura 3.31 se muestra el lugar elegido para la implementación del prototipo.



Figura 3.31 Sitio elegido para la implementación del prototipo

La figura 3.32 muestra las tres zonas que monitoreará la cámara de video, (a) Zona 1, parte derecha de calle Vicente Ferrer; (b) Zona 2, parte izquierda de la calle Vicente Ferrer y (c) Zona 3, parte central calle Pucará.



Figura 3.32 Zonas monitoreadas

Para sujetar la caja contenedora del sistema prototipo al poste se utilizó una amarra metálica, como se observa en la figura 3.33.



Figura 3.33 Amarra metálica

Finalmente la figura 3.34 muestra la implementación final del prototipo.



Figura 3.34 Instalación final del prototipo

CAPÍTULO 4

RESULTADOS Y COSTOS

En el presente capítulo se presentan las pruebas realizadas para verificar el correcto funcionamiento del sistema prototipo, además se muestran los resultados obtenidos, también se presenta una matriz FODA en la que se analizó los pros y contras del proyecto y además se muestran los costos totales del mismo

4.1 Pruebas de funcionamiento

Se realizarán dos tipos de pruebas. Las primeras pruebas fueron realizadas para comprobar la calidad de conexión entre el sistema y el celular de prueba cuando se utilizan SIM de diferentes operadoras; y las segundas pruebas se realizaron con el fin de verificar el funcionamiento general del sistema.

4.1.1 Pruebas de funcionamiento con SIM de diferentes operadoras

Se experimentó con SIM de dos operadoras; para dichas pruebas se tomaron en cuenta las operadoras CLARO y MOVISTAR, con las siguientes configuraciones:

CONFIGURACIÓN	MODEM	CEL. DE PRUEBA
1	CLARO	CLARO
2	CLARO	MOVISTAR
3	MOVISTAR	CLARO
4	MOVISTAR	MOVISTAR

Tabla 4.1 Configuración del SIM

Para las presentes pruebas se tomaron como base 20 repeticiones de envíos de comandos al sistema por cada configuración, los resultados obtenidos se presentan en la Tabla 4.2.

Configuración	N° de intentos	N° de aciertos	% de Error $((V_{real}-V_{medido})/V_{real})*100$
1	20	20	0,0%
2	20	16	20,0%
3	20	19	5,0%
4	20	17	15,0%

Tabla 4.2 Resultado de prueba realizada

Para los cálculos de error se tomó como V_{real} a N° de intentos y V_{medido} a N° de aciertos.

4.1.2 Pruebas de funcionamiento del sistema

Para realizar las pruebas del funcionamiento del sistema prototipo se conectó el sistema al computador por medio de un cable USB con lo que se verificó el progreso de la programación. Para las pruebas se utilizaron dos SIM de la operadora CLARO uno en el modem GSM y el otro en el celular de prueba, el celular utilizado en la pruebas es de marca Nokia, modelo 1001.

Los resultados obtenidos se muestran a continuación:

Al momento de conectar el sistema al suministro de energía la placa electrónica Arduino Uno configura al modem en modo texto; en el momento que el modem se encuentra listo envía un SMS al celular de prueba con el mensaje “Modem listo...” como se observa en la figura 4.1.

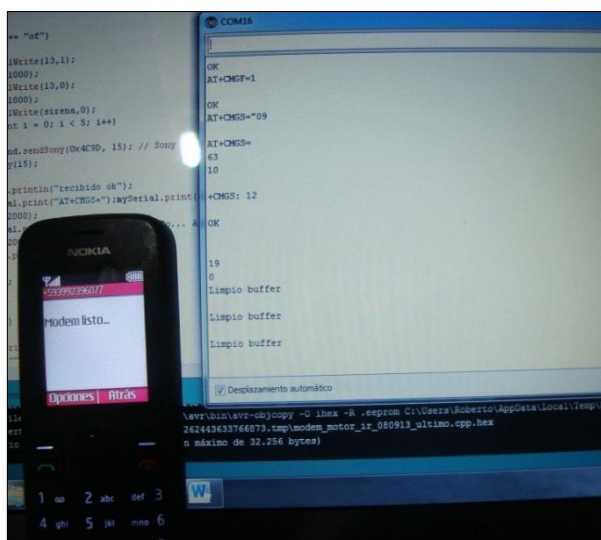


Figura 4.1 Configuración del Modem

En la figura se puede apreciar el monitor serial con el proceso de programación del modem y el SMS recibido luego de dicho proceso.

Una vez configurado el modem el sistema entra en funcionamiento y ya se lo puede utilizar, para lo cual se procedió a enviar los códigos descritos en la tabla 3.1; cabe recalcar que el motor que mueve a la cámara de video queda en la última posición que fue desplazado.

Para mover el motor hacia la izquierda envió el código “iz”. En la figura 4.2 se pueden apreciar los resultados, (a) proceso de programación mostrada en el monitor serial, (b) SMS recibido en el celular de pruebas y (c) sistema en funcionamiento.

```

COM16
1
1
1
1
0
recibido ok
Acciono Fin Izquierdo y paro Motor a Izquierda
AT+CMGS="0959504317"
> Sensor Izquierdo activado... ALARMA A
0... ALARM
63
10
Limpio buffer

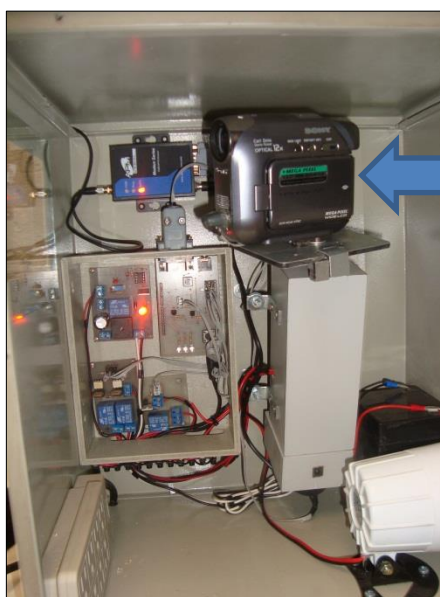
+CMG: "1234", , "2013/09/18 00:01:42-20"
Sin saldo! Envía tu
do! Envía
63
10
Limpio buffer

Desplazamiento automático
No hay fin de línea
9600 baud
  
```

(a)



(b)



(c)

Cámara de video apuntando su lente hacia la zona izquierda.

Figura 4.2 Movimiento de la cámara de video hacia la izquierda

Para mover el motor hacia la derecha se envió el código “de”. En la figura 4.3 se aprecian los resultados, (a) proceso de programación mostrada en el monitor serial, (b) SMS recibido en el celular de pruebas y (c) sistema en funcionamiento.

```

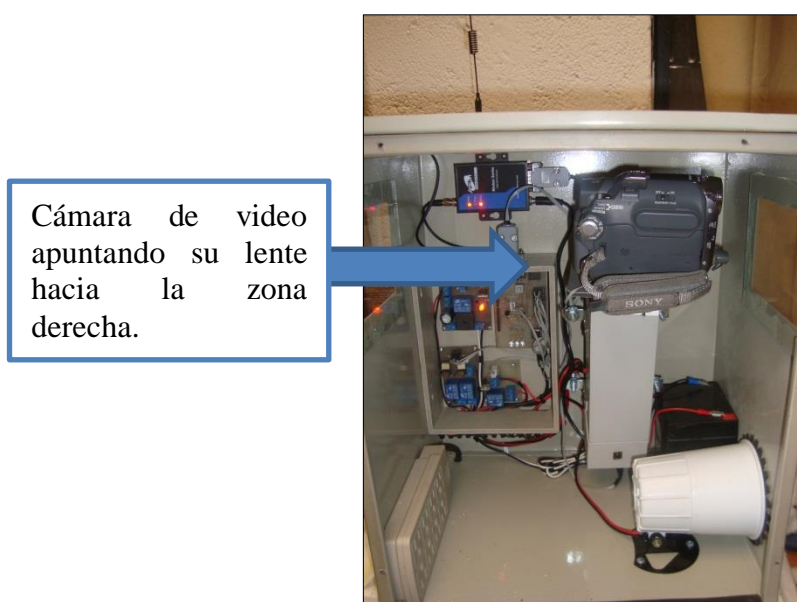
COM16
1
1
1
1
1
1
0
recibido ok
Acciono Fin Derecho y paro Motor a derecha
AT+CMGS="0959504317"
> Sensor Derecho activado... ALARMA ACT
.. ALARMA
63
10
+CMGS: 15
OK
19
0

```

(a)



(b)

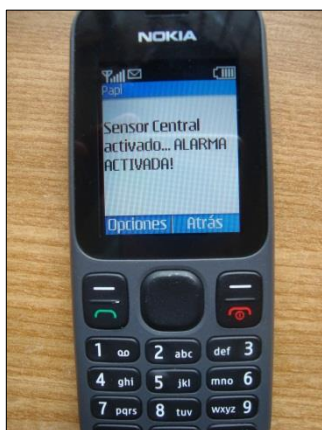
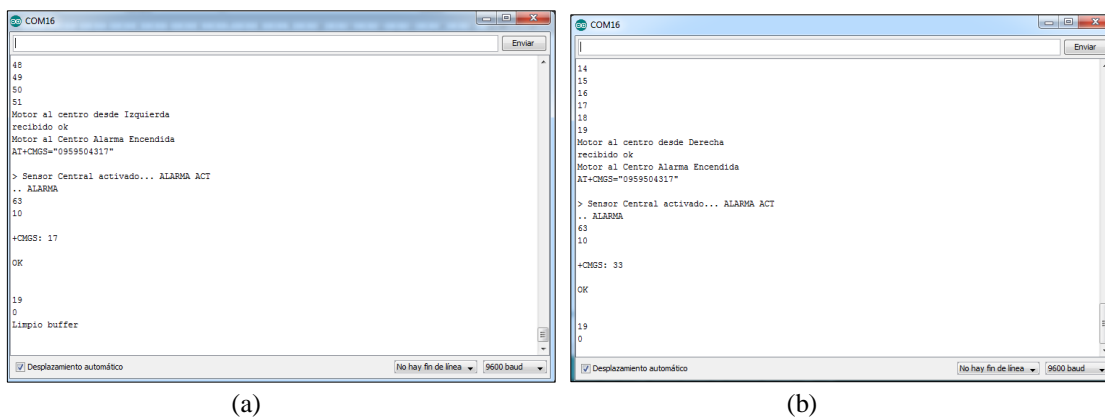


(c)

Figura 4.3 Movimiento de la cámara de video hacia la derecha

Para mover la cámara de video hacia el centro, el sistema censó la posición final del motor mediante los fines de carrera y según sus estados el sistema mostrará dos mensajes diferentes por el monitor serial. En la figura 4.4 se muestran los resultados, (a) motor moviéndose al centro desde la izquierda, (b) motor moviéndose al centro desde la derecha, (c) SMS recibido en el celular de pruebas.

El SMS recibido en el celular de pruebas es el en los dos casos.



(c)



Cámara de video apuntando su lente hacia la zona centro.

(d)

Figura 4.4 Movimiento de la cámara de video hacia el centro

Para desactivar el sistema se envió un SMS con el comando "of". En la figura 4.5 se muestran los resultados, (a) proceso de la programación mostrada en el monitor serial, (b) SMS recibido en el celular de pruebas.

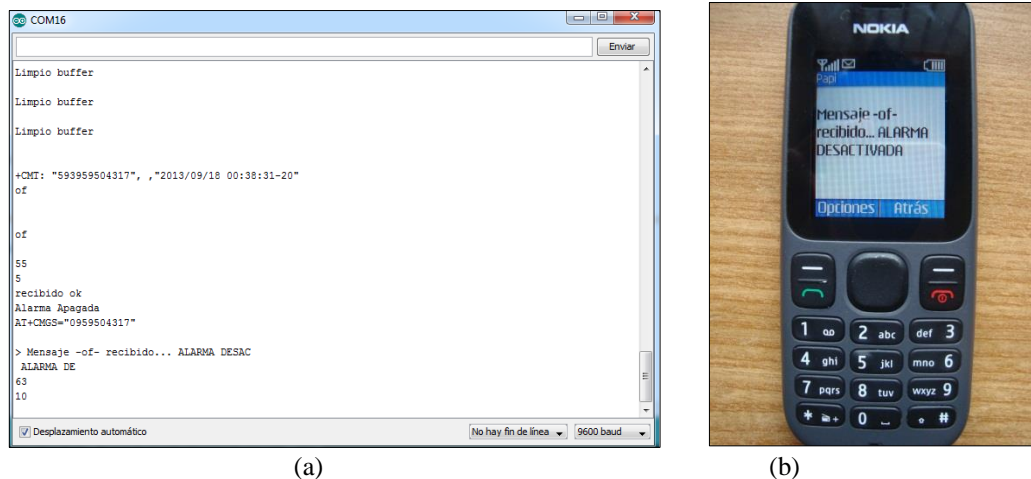


Figura 4.5 SMS generado al desactivar el sistema

Debido a que los comandos de activación/desactivación del sistema ya se encuentran definidos y almacenados en la memoria de la placa electrónica Arduino Uno, cuando el usuario envía un SMS al modem con cualquier otra palabra el sistema no lee dicho mensaje y no realiza acción alguna o sigue ejecutando la tarea programada.

4.2 Análisis de resultados

4.2.1 Análisis de resultados de las pruebas realizadas con SIM diferentes

En la tabla 4.2 se observan los resultados obtenidos de la prueba de intercambio de SIM; el tiempo promedio de recepción y envío de los SMS por parte del modem fue de 5 segundos.

Como se observa, la configuración 1 tiene un 0% de error, lo que quiere decir que el sistema tiene un buen desempeño.

En las configuraciones 2, 3 y 4 se puede observar que no llega el 100% de los intentos realizados y esto se debe a que la red del SIM que se está utilizando en el modem no es la misma que la del celular de prueba, de manera que el SMS tarda demasiado tiempo en llegar o simplemente no llega; por lo que el sistema no realiza ninguna acción. Además, cabe recalcar que la señal de cobertura de la operadora MOVISTAR es débil en el sector donde fué instalado el sistema.

4.2.2 Análisis de las pruebas del funcionamiento del sistema

Una vez realizadas las pruebas de funcionamiento del sistema se dieron como resultado los datos presentados en la tabla 4.3.

Comando	Recibido	Acción	Motor	Sensor	Alarma	Celular
de	ok	fin iz	Hacia la izquierda	Activado	Activada	SMS recibido
iz	ok	fin de	Hacia la derecha	Activado	Activada	SMS recibido
ct	ok	censo fin iz	Al centro de la izquierda	Activado	Activada	SMS recibido
ct	ok	censo fin de	Al centro de la derecha	Activado	Activada	SMS recibido
of	ok	apagar	Desactivado	Desactivado	Desactivada	SMS recibido

Tabla 4.3 Resultados obtenidos de la prueba de funcionamiento del sistema

La tabla muestra los resultados de la prueba del sistema realizados con SIM de la operadora CLARO para una mayor efectividad; y fue monitoreado por medio del puerto serial. Como se puede observar al momento del envío de un comando, el sistema automáticamente realiza las acciones programadas como: activación y desactivación del mismo, movimiento del motor, envío de SMS y accionamiento de los fines de carrera, todo esto realizado de manera simultánea e instantánea, el tiempo que demora el sistema en ejecutar una acción es de aproximadamente 10 s.

En la tabla 4.3 se observa las pruebas que se hicieron al sistema para que realice los movimientos de: derecha, izquierda y centro con lo que el sistema respondió por medio del puerto serial de la manera adecuada; al recibir el SMS respondió OK; se pudo comprobar que la acción realizada fue la que determinaba el proceso con lo cual el motor dió un giro a la posición indicada; el sensor actuó; la alarma se activó y adicionalmente el teléfono celular recibió el SMS de alerta.

Con todo esto se puede concluir que el sistema se encuentra en óptimas condiciones de funcionamiento.

4.3 Análisis FODA

<p style="text-align: center;">FORTALEZAS</p> <ul style="list-style-type: none"> • Prototipo único en el mercado Ecuatoriano basado en software libre. • Capacidad de usuarios ilimitados. • Fácil instalación y uso 	<p style="text-align: center;">OPORTUNIDADES</p> <ul style="list-style-type: none"> • No existe en el país un sistema de alarma comunitaria con las mismas características. • Es innovador y puede ser mejorado • Puede ser instalada en cualquier parte del país.
<p style="text-align: center;">DEBILIDADES</p> <ul style="list-style-type: none"> • Ya existen sistemas similares en el mercado nacional. • La cámara de video debe permanecer encendida permanentemente por lo que su tiempo vida útil se ve reducido • El tiempo de envío de un SMS al sistema es relativamente alto por lo que la respuesta del mismo no es inmediata. 	<p style="text-align: center;">AMENAZAS</p> <ul style="list-style-type: none"> • El diseño puede ser copiado. • La inserción del sistema al mercado podría ser difícil debido a las aplicaciones existentes para teléfonos inteligentes. • La llegada de los SMS al sistema se ve afectada debido a la utilización de SIM de diferentes operadoras y la cobertura GSM.

Tabla 4.4 Matriz FODA

4.4 Análisis Financiero

En el presente ítem se presenta los precios de los elementos electrónicos utilizados para la elaboración del prototipo, los mismos que se encuentran detallados de acuerdo a cada placa electrónica y otros componentes.

Placa Electrónica de Control			
Descripción	Cant.	V.Unitario	V. Total
Led Infrarrojo AZ	1	0,11	0,11
Receptor Infrarrojo	1	0,45	0,45
R 1/4W	9	0,27	2,43
Led VC	3	0,30	0,90
DB9 Placa Hembra	1	0,55	0,55
DB9 Macho Cable	2	0,90	1,80
Case DB9	2	0,60	1,20
Swich GR BL	1	0,65	0,65
Molex 3P	1	0,30	0,30
1N3904	3	0,36	1,08
Pulsador 7MM	1	0,30	0,30
Espadin Simple Macho	3	1,35	4,05
Jack Estéreo Pequeño	1	0,40	0,40
PBC	4	5,00	20,00
Jack Estéreo Medio	1	0,45	0,45
Zener 5.1V	1	0,20	0,20
Conectores Crim de 3P	5	0,70	3,50
TOTAL USD			<u>\$ 38,37</u>

Tabla 4.5 Costo de materiales utilizados en la placa electrónica de Control

Placa de control del motor			
Descripción	Cant.	V.Unitario	V. Total
Optoacopladores 4N25	2	1,30	2,60
R1/4W	2	0,06	0,12
Zócalos 6P	2	0,24	0,48
Tip 41	2	1,10	2,20
1N4148	2	0,24	0,48
Led 5MM	2	0,20	0,40
Bornera 2P	3	1,05	3,15
Espadin Macho	1	0,45	0,45
Relés 12V 5P	2	0,08	0,16
TOTAL USD			<u>\$ 10,04</u>

Tabla 4.6 Costo de materiales utilizados en la placa electrónica del motor

Placa fuente de alimentación			
Descripción	Cant.	V.Unitario	V. Total
R 1/4W	6	0,18	1,08
CE 1000 UF/50V	1	0,60	0,60
CC 100NF	2	0,20	0,40
LM358	1	0,45	0,45
Zócalo 8P	1	0,12	0,12
TIP 41	1	0,55	0,55
Zener 6.2V	1	0,20	0,20
Led	1	0,10	0,10
1N4007	1	0,12	0,12
Borneras 2P	4	1,40	5,60
Puente REC 10K	1	1,00	1,00
Potenciómetro de Precisión 10K	1	0,65	0,65
Relé 12V 5P	1	0,80	0,80
SW Negro On/Of	2	0,90	1,80
Transformador	1	6,00	6,00
TOTAL USD			<u>\$ 19,47</u>

Tabla 4.7 Costo de materiales utilizados en la placa de alimentación

Placa control Sirena			
Descripción	Cant.	V.Unitario	V. Total
R 1/4W 1K	1	0,03	0,03
2N904	2	0,24	0,48
TIP 41	1	0,55	0,55
1N4007	1	0,12	0,12
Molex 3P	1	0,30	0,30
Bornera 2P	1	0,35	0,35
Bornera 3P	1	0,40	0,40
Relé 12V 5P	1	0,80	0,80
TOTAL USD			<u>\$ 3,03</u>

Tabla 4.8 Costo de materiales utilizados en la placa de control de la sirena

Caja Metálica			
Descripción	Cant.	V.Unitario	V. Total
Mano de Obra	1	65	65,00
Silicón Transparente 310ML	1	3,53	3,53
Enchufe 3P 15A 125V	1	0,64	0,64
Amarra cable 12"X4.5MM	1	2,44	2,44
Regleta	1	4,42	4,42
Empaque de caucho	1	0,28	0,28
Tornillos	1	2,30	2,30
Pintura Sintética	1	6,18	6,18
Extensión Eléctrica	1	5,98	5,98
TOTAL USD			<u>\$ 90,77</u>

Tabla 4.9 Costo de materiales utilizados en la elaboración de la caja metálica

Otros Materiales			
Descripción	Cant.	V.Unitario	V. Total
MODEM GPRS- GSM ZTE	1	105,36	105,36
Filmadora SONY Mod. DCR-HC42	1	300,00	300,00
Tarjeta electrónica Arduino Uno	1	32,50	32,50
Batería	1	15,00	15,00
Sirena	1	8,00	8,00
Motor	1	25,00	25,00
Fin de Carrera	2	2,00	4,00
Terminales Eléctricas	2	0,45	0,90
Bornera Eléctrica	1	0,95	0,95
SIM Claro + Recarga (Modem)	1	25,00	25,00
SIM Claro + Recarga (Cell. Pruebas)	1	25,00	25,00
SIM Movistar + Recarga (Cell. Pruebas)	1	25,00	25,00
Internet	1	60,00	60,00
Resma de papel/Impresiones	1	42,00	42,00
TOTAL USD			<u>\$ 668,71</u>

Tabla 4.10 Costo otros materiales utilizados

MANO DE OBRA	
Elaboración y armado del sistema	<u>\$ 320,00</u>

Tabla 4.11 Costo de mano de obra

COSTO TOTAL DEL PROTOTIPO	
Indicador	Valor
Placa electrónica de control	\$ 38,37
Placa de control del motor	\$ 10,04
Placa fuente de alimentación	\$ 19,47
Placa control sirena	\$ 3,03
Otros Materiales	\$ 668,71
Caja Metálica	\$ 90,77
Mano de obra	\$ 320,00
TOTAL USD	<u>\$ 1.150,39</u>

Tabla 4.12 Costo total del prototipo

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La tecnología Arduino con la ayuda de sus múltiples librerías y placas electrónicas brinda la posibilidad de construir el prototipo de alarma comunitaria fácilmente ya que puede controlar sin problemas los periféricos de salida del sistema.
- La programación de la placa electrónica Arduino Uno se puede realizar directamente y sin necesidad de desconectarla del sistema prototipo.
- Debido a que la plataforma de programación Arduino es de acceso libre las librerías utilizadas para la construcción del sistema pueden ser descargadas directamente del internet.
- La gran cantidad de librerías existentes en el internet y que realizan la misma acción provocaron que se deban realizar pruebas de funcionamiento antes de elegir la librería que presente menos fallas y que se acople sin problemas a los requerimientos del prototipo.
- Se diseñó un sistema de control utilizando tecnología Arduino que gracias a sus características permitió implementar todas las especificaciones de la alarma barrial.
- La cámara de video utilizada puede ser remplazada por otra de menor valor que cumpla con los requerimientos del sistema.
- La creación de plantillas SMS con los comandos de activación y desactivación del sistema para ser cargados en los teléfonos celulares de los usuarios facilitaron la utilización del sistema.
- El tiempo que tarda en llegar un SMS hacia el modem desde el celular del usuario y viceversa es de aproximadamente 8 s, que es un retardo aceptable para las condiciones de diseño del sistema.
- Luego de pruebas realizadas al sistema se determinó que su efectividad se ve limitada cuando se utilizan SIM de diferentes operadoras ya que varios SMS

tardan mucho tiempo en llegar o simplemente no llegan al sistema debido a la cobertura de la red GSM.

- Debido a que el sistema trabaja bajo la red GSM, el prototipo puede ser implementado en cualquier región del país donde exista cobertura GSM.

5.2 Recomendaciones

- Capacitar a los usuarios del sistema sobre su utilización y funcionamiento para que no existan falsas activaciones de alarma.
- Los usuarios del sistema necesitan organizarse para responder inmediatamente al SMS de alerta enviado por el prototipo.
- Vigilar constantemente la caja contenedora del prototipo ya que personas mal intencionadas pueden causarle daños irreparables con objetos contundentes.
- Que personal autorizado manipule el prototipo ya que podría ser modificado para otros fines y alterar su funcionamiento.
- Bajar los archivos almacenados en la memoria de la cámara de video a un computador para que ésta no se sature de datos y ocasione un mal funcionamiento del sistema de grabación.

BIBLIOGRAFÍA

Bravo, L (2011). DISEÑO E IMPLEMENTACIÓN DE ALARMAS COMUNITARIAS A TRAVÉS DE UN OPERADOR MÓVIL. Tesis de Ingeniería en Electrónica y Telecomunicaciones, Escuela Politécnica del Ejército, Sangolquí.

De: <http://repositorio.espe.edu.ec/bitstream/21000/3316/1/T-ESPE-031145.pdf>

Sanguano, B (2013). ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PROTOTIPO DE CONTROL VÍA MENSAJE DE TEXTO EN UN TAXÍMETRO. Tesis de Ingeniería en Electrónica y Telecomunicaciones, Universidad Tecnológica Israel, Quito.

De: <http://186.42.96.211:8080/jspui/bitstream/123456789/898/1/bsanguano-ecalvache.pdf>

Pazmiño, D (2012). Diseño e Implementación de un Sistema de Control de Ruta Vehicular Mediante Comparación con el uso de GPS. Tesis de Ingeniería en Electrónica y Telecomunicaciones, Universidad Tecnológica Israel, Quito. De:

<http://186.42.96.211:8080/jspui/bitstream/123456789/156/1/SISTEMA%20CONTROL%20DE%20RUTA%20VEHICULAR%20MEDIANTE%20COMPARACION%20CON%20EL%20USO%20DE%20GPS.pdf>

Burbano, F (2008). CAPITULO 2. INTERFACE ENTRE ORDENADOR Y MÓDEM GSM

Disponible en <http://dspace.ups.edu.ec/bitstream/123456789/166/3/Capitulo%202.pdf>

LUCKYSTAR'S OPINIONS (2009). Using AT commands to Send and Receive SMS

Disponible en <http://oldlight.wordpress.com/2009/06/16/tutorial-using-at-commands-to-send-and-receive-sms/>

ARDUINO (2013). Página oficial

Disponible en <http://arduino.cc/es/Main/Software>

davidmiguel (2013). Receptor/emisor IR

Disponible en <http://www.davidmiguel.com/arduino/dia-3-sensor-ir/>

pjrc (2013). IRremote Library

Disponible en http://www.pjrc.com/teensy/td_libs_IRremote.html

Libertad en la Electrónica (2009). Decodificando el Infrarrojo: SIRC

Disponible en <http://libertadelectronica.wordpress.com/2009/12/03/decodificando-el-infrarrojo-sirc/>

picproject (2010). Sony SIRC infrared protocol

Disponible en <http://picprojects.org.uk/projects/sirc/sonysirc.pdf>

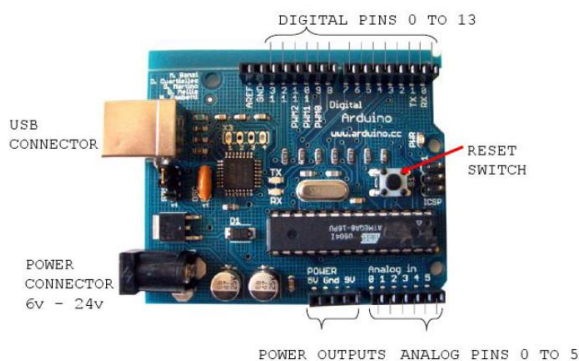
ANEXOS

ANEXO 1

**Datos técnicos de la placa
electrónica Arduino Uno**

TARJETA ARDUINO

La tarjeta Arduino, contiene para interacción con el usuario trece entradas/salidas digitales, seis entradas analógicas y un puerto serial que permite realizar comunicación con periférico, además de un puerto serial una conexión USB, en la figura podemos observar la localización de las entradas analógicas y digitales como los pines de alimentación. También tiene un pulsador para resetear cualquier fallo que exista en los procesos que se vayan a realiza con la tarjeta Arduino.



HARDWARE

El microprocesador ATmega328

- 32 kbytes de memoria Flash
- 1 kbyte de memoria RAM
- 16 MHz • Entradas y salidas
- 13 pines para entradas/salidas digitales (programables)
- 5 pines para entradas analógicas
- 6 pines para salidas analógicas (salidas PWM)
- Completamente autónomo: Una vez programado no necesita estar conectado al PC

ESPECIFICACIONES TECNICAS

- Microcontrolador ATmega328
- Voltaje de operación 5V
- Voltaje de entrada (recomendado) 7-12 V
- Voltaje de entrada (límite) 6-20 V
- Digital I/O Pins 14 (con 6 salidas PWM)
- Entradas analógicas Pines 6
- DC corriente I/O Pin 40 mA
- DC corriente 3.3V Pin 50 mA
- Memoria Flash 32 KB (2 KB para el bootloader)
- SRAM 1 KB
- EEPROM 512 byte
- Velocidad de reloj 16 MHz

Las Entradas analógicas son de 10 bits, por lo que entregan valores entre 0 y 1023. El rango de voltaje está dado entre 0 y 5 volts, pero utilizando el pin AREF, este rango se puede variar a algún otro deseado.

ANEXO 2

**Datos técnicos del
microcontrolador ATmega
328P**

- 1.1.4 Port C (PC5:0)**
Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5.0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.
- 1.1.5 PC6/RESET**
If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.
If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 28-3 on page 318](#). Shorter pulses are not guaranteed to generate a Reset.
The various special features of Port C are elaborated in ["Alternate Functions of Port C" on page 85](#).
- 1.1.6 Port D (PD7:0)**
Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in ["Alternate Functions of Port D" on page 88](#).
- 1.1.7 AV_{CC}**
AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.
- 1.1.8 AREF**
AREF is the analog reference pin for the A/D Converter.
- 1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)**
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

9.8 ATmega328P

Speed (MHz) ⁽⁵⁾	Power Supply (V)	Ordering Code ⁽²⁾	Package ⁽¹⁾	Operational Range	
20	1.8 - 5.5	ATmega328P-AU	32A	Industrial (-40°C to 85°C)	
		ATmega328P-AUR ⁽⁵⁾	32A		
		ATmega328P-MMH ⁽⁴⁾	28M1		
		ATmega328P-MMHR ⁽⁴⁾⁽⁵⁾	28M1		
		ATmega328P-MU	32M1-A		
		ATmega328P-MUR ⁽⁵⁾	32M1-A		
		ATmega328P-PU	28P3		
		ATmega328P-AN	32A		Industrial (-40°C to 105°C)
		ATmega328P-ANR ⁽⁵⁾	32A		
		ATmega328P-MN	32M1-A		
ATmega328P-MNR ⁽⁵⁾	32M1-A				
		ATmega328P-PN	28P3		

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. See [Figure 29-1 on page 308](#).
 4. NiPdAu Lead Finish.
 5. Tape & Reel.

Comparison Between ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P

The ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. [Table 2-1](#) summarizes the different memory and interrupt vector sizes for the three devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

ATmega88PA, ATmega168PA and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48PA, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

ANEXO 3

**Datos técnicos del Modem
GSM ZTE MG3006**

ZTE MG3006 Module Technical Specifications

VERSION: V1.3

ZTE CORPORATION

3 Appearance and framework

Appearance of MG3006 is as following figure 3-1:

Figure 3-1 appearance of MG3006 module



- Dimension (length x width x height) : 44.0 mm x 28.0mm x 7.6mm
- Weight: 8g

4 Functions and interfaces

The basic functions of module are as below:

- Support Quad Band: GSM 850/EGSM 900/DCS 1800/PCS 1900
- Support packet data service
- Support circuit switched data service
- Support SMS service
- Support standard AT commands and extended AT commands
- Support standard UART interface
- Support dual-path audio interface
- Supplementary service functions: incoming call display, call forward, call maintenance, call stand by, triple call service and so on.
- Support TCP/IP protocol

5 Technical specifications

5.1 Communication protocols and technical specifications

The communication protocols and technical specifications of MG3006 modules is as following table 5-1:

Table 5-1 communication protocols and technical specifications

Access mode	GSM
Tech-spec	GSM phase 2/2+
Rx/Tx frequency interval	45MHz for GSM 850 45MHz for EGSM 900 95MHz for DCS 1800 80MHz for PCS 1900
Voice encoding	- Half rate (HR) - Full rate (FR) - Enhanced Full rate (EFR) - Adaptive Multi-Rate (AMR)

- MG3006 frequency band: GSM 850/EGSM 900/DCS 1800/PCS 1900 MHz. There frequency bands are shown in table 5-2. Data transmission rate depends on interval assignment and channel encoding of GPRS.

Table5-2 frequency band

name	Tx frequency band(MHz)	Rx frequency band (MHz)
GSM 850	824~849 MHz	869~894MHz
EGSM 900	880~915 MHz	925~960MHz
DCS 1800	1710~1785MHz	1805~1880MHz
PCS 1900	1850~1910MHz	1930~1990MHz

MG3006 module supports CLASS 10, the interval assignment is as following tableTable5-3:

Table5-3 interval assignment

down	up	Maximum supported interval at the same time
4	2	5

GPRS encoding modes supported by MG3006 module are as following table 5-4:

Table5-4 encoding modes

Encoding modes	Data rate(kbps)
CS-1	9.05
CS-2	13.4
CS-3	15.6
CS-4	21.4

The maximum theoretic data rate supported by MG3006 module is as following table5-5:

Table5-5 data rate

Encoding mode	Download (kbps)	Upload(kbps)
CS-1	36.2	18.1
CS-2	53.6	26.8
CS-3	62.4	31.2
CS-4	85.6	42.8

MG3006 module supports Class B, but when GPRS and GSM service exist at the same time, GPRS breaks off, on the other hand, voice and SMS service of GSM is prior. After that GPRS

5.4 Recommendation of antenna specs

The recommended antenna specs are as following table 5-18:

Table5-18 recommended antenna specs

VSWR	1.5:1 maximum
gain	At least 0 dBi in one direction
Input impedance	50Ω
Polarized form	Vertical polarizing

The requirements for antenna's gain are different in different environment. Commonly, in used frequency range, the larger gain, the better capability; otherwise, out of this range, the smaller gain, the better capability.

The antenna seat's type of MG3006 module is MM9329-2700B.

5.5 Power supply

5.5.1 Input voltage

The input voltage is shown in table 5-19:

Table5-19 input voltage

state	Max. voltage	Typical voltage	Min. voltage
Power supply	4.25 VDC	3.90 VDC	3.30 VDC

5.6 Working conditions

- Working temperature: -20°C ~ +80°C
- Storage temperature: -40°C ~ +85°C
- humidity: 0% ~ 95%

6 Reliability test standard

6.1 Low temperature running experiment

- Required Temperature: -20°C
- Duration Time: 16H
- Reference standard: GB/T 2423.1-2001

6.2 Low temperature storage experiment

- Required Temperature: -40°C
- Duration Time: 24H
- Reference standard: GB/T 2423.1-2001

6.3 High temperature running experiment

- Required Temperature: +80°C
- Duration Time: 16H
- Reference standard: GB/T 2423.2-2001

6.4 High temperature storage experiment

- Required Temperature: +85°C
- Duration Time: 24H
- Reference standard: GB/T 2423.2-2001

6.5 High temperature, high humidity experiment

- Required Temperature: +40°C
- Required Humidity: 85%RH
- Duration Time: 48H
- Reference standard: GB/T 2423.2-2001

6.6 High-low temperature striking experiment

- Cycles: 5
- Temperature Range: -20°C ~ +80°C
- Duration Time: 2h
- recovery time: 2h
- Reference standard: GB/T 2423.3-2001

ANEXO 4

Filmadora SONY modelo

DCR-HC42

SONY

DCR-HC42

NEW

MiniDV Handycam® Camcorder

- ▲ Step-up Feature
- ◆ New Feature

The DCR-HC42 is stylish yet feature packed with a supplied Handycam Station for the convenient ability to share your memories and charging your battery. Along with a 1 Megapixel CCD Imager and 2.7" Wide (16:9) Hybrid Touch Panel LCD your memories will be enhanced with more video captured.

- ▲ **1.0 Megapixel (Gross) Advanced HAD™ CCD Imager**
1/5.5" Advanced HAD™ (Hole Accumulation Diode) CCD with 690K (effective) pixels provides stunning detail and clarity with exceptional video (up to 520 lines of horizontal resolution) and still image performance. The Megapixel technology enhances your digital videos and digital still photography performance.
- **MiniDV Recording**
Offers up to 520 lines of horizontal resolution, 3X the color bandwidth and a significantly higher signal/noise ratio, to provide spectacular video performance.
- **12X Optical/480X Digital Zoom**
The optical zoom helps to bring the action close up from far away. In addition, with the Digital Zoom and interpolation means that extreme digital zooming is clearer, with less distortion than previous types of digital zooms.
- **Professional Quality Carl Zeiss® Vario-Tessar® Lens**
Carl Zeiss lenses have an advanced optical multi-layer coating that offers less glare and flare with increased contrast. This results in vivid image brilliance, true-to-life color saturation and perfect renditions of subtle tones.
- **Progressive Shutter System**
A mechanical shutter system that provides progressive scan performance while utilizing an interlaced scanning system. Digital still images will be sharp and clear with excellent definition.
- **Analog To Digital Conversion with Pass-Through**
Convert and/or record any analog NTSC video source to digital video via the analog inputs. Analog NTSC video can also be passed through the digital Handycam Camcorder directly into a PC via the i.LINK interface in real-time for easy PC editing of your analog footage.
- **Super SteadyShot® Picture Stabilization System**
Picture stabilization system that uses motion sensors to detect and compensate for camera "shake" without compromising picture quality like some other digital stabilization systems.
- ▲ **2.7" wide Touch Panel SwivelScreen™ LCD Display (123K Pixels)**
Provides excellent viewing clarity for both 16:9 and 4:3 aspect with improved resolution. The high resolution 123K pixel LCD screen rotates up to 270 degrees for multiple viewing angles providing sharp, detailed images for monitoring or playback.
- **Handycam Station**
The supplied Handycam Station provides quick and easy connections to a TV for viewing or to a PC for editing and sharing video. Just placing the camcorder on the Handycam Station instantly makes the connections. No need to search for cables or wires when charging or playing back.
- **USB Streaming**
Stay connected to family and friends by broadcasting live video with audio over the internet using the USB interface. You can even playback your favorite recorded scene on the camcorder's tape. It's like having your own video conferencing system.
- **i.LINK® DV Interface***
A digital high-speed bi-directional interface which provides Audio/Video communication between two compatible devices. This connection allows for pure digital transfer between devices equipped with an IEEE1394 interface, such as camcorders, digital VTRs, capture cards and PCs.
- ▲ **InfoLithium® Battery with AccuPower™ Meter System†**
Charge the battery at anytime because unlike NiCad (Nickel Cadmium) batteries, Sony's rechargeable Lithium Ion batteries are not subjected to a life shortening "memory effect". Sony's exclusive AccuPower™ meter displays the battery time remaining in minutes, in either the viewfinder or on the LCD screen.
- **Digital Still Memory Mode with Memory Stick Duo™ Media‡**
Capture still images up to 1152x864 resolution, directly onto Memory Stick Duo™ Media for easy transfer to PCs for emailing, printing or sharing with other compatible Memory Stick™ devices.
- **Tele Macro Function**
Useful for shooting close up shots, but without having the camcorder casting a shadow on the subject. Images shot in the Tele Macro Mode will have the subject properly focused with the background unfocused so that the subject stands out.
- ▲ **Active Interface Shoe**
Provides a location directly on the camcorder on which to add optional accessories while providing power and two-way communication to those accessories.
- **Stamina® Battery Power Management System**
Using the optional NP-FP90 InfoLithium® Battery, the DCR-HC42 can continuously record for up to 8 hours on a single full charge, providing extra long battery life.
- **Super NightShot® Plus Infrared System**
With Sony's Super NightShot Infrared System you can capture natural looking video, even when shooting in low light. Record subjects up to 20 feet away using the built-in infrared system, but without the overall monochrome color common with earlier low-light recording systems.



additional features

- Program AE (Auto Exposure) Modes
- Picture Effects
- Battery Information
- Manual Focus
- Easy Handycam Button

specifications

USB Streaming: Yes
SteadyShot® Image Stabilization: Yes, Super SteadyShot
Accessory Shoe: Yes, Active Interface
White Balance: Auto/Outdoor/Indoor/One-push
Memory Stick PRO™ Media Compatibility:
 Tested to support up to 1GB media capacity*, does not support high speed transfer function; does not support Access Control security function.
Low Light Capability: Super NightShot Plus
Interface: Interface Connector for Handycam Station: Yes
Viewfinder: Color, 123K Pixels
Imaging Device: 1/5.5" 1070K Gross Pixel CCD
Still Actual: 1000K Pixels
LCD: 2.7" Wide (16:9) Color (123K)
USB Port(s): Yes (On Handycam Station)
LANC Terminal: Yes
Digital Audio Input(s): Yes
Digital Video Input(s): Yes
Digital Video Output(s): Yes (Special)
Digital Audio Output(s): Yes (Stereo, Special)
Headphone Output(s): N/A
Input(s): Microphone N/A
Minimum Illumination: 7 Lux (0 Lux with NightShot Plus Infrared System)
Lens Type: Carl Zeiss® Lens:
 Vario-Tessar®: Yes
Zoom: 12X, 480X (Digital)
Focus: Full Range Auto/Manual (Touch Panel)
35mm Conversion: 44-880mm (4:3)-Camera
 46-960mm
 (16:9)-Memory
 44-880mm
 (4:3)-Camera
Focal Distance: 23-46mm; 390mm(Telemacro)
Aperture: 1.8-2.5
Exposure: Yes, Touch Panel (24 steps)
Shutter Speed: 1/2-1/4000 (AE Mode)
Power Consumption: 2.6W/3.0W/3.2W (VFL/CD/VF LCD)
Limited Warranty: 1 Year Parts; 90 Days Labor
Video Actual: 690K Pixels
Filter Diameter: 25mm
Weight: 14 oz (397 g) w/out Tape and Battery
Dimensions (WHD): 2 1/4" x 3 5/8" x 4 1/2"
 (54.7 x 90 x 111.7mm)
Supplied Accessories: Power Adapter/In-Camera Charger (AC-L25)
 InfoLithium® Rechargeable Battery (NP-FP50)
 Handycam® Station (DRCA-C121)
 Remote Commander® Remote Control

(RMT-831)
 Stereo AV Cable
 USB Cable
 Shoulder Strap
 CD-ROM with USB Driver (PicturePackage™ Software Ver. 1.5 for Sony)

UPC Code: 0272426697580

service and warranty

Limited Warranty: 1 Year Parts; 90 Days Labor

optional accessories

Handycam Camcorder Starter Kit for MiniDV (ACC-DV)
 High Capacity InfoLithium™ Batteries (NP-FP50/FP70/FP90)
 Adaptor/Dual Charger (ACV-QP10 AC)
 AC/DC Adaptor (DCC-L50B)
 Carrying Case (LCS-1W4)
 Carrying Case (LCS-1W5)
 Telephoto Lens (VCL-2025S)
 High-Grade Telephoto Lens (VCL-HG2025)
 Wide Angle Lens (VCL-HG0725)
 Wide Angle Lens (VCL-0625S)
 Lightweight Tripod (VCT-R610)
 Pocket Tripod (VCT-TK1)

worryfreedigitalvideo

BATTERY LIFE*

Battery† (Fully charged)	Continuous Record Time		Continuous Playback Time LCD On/Off
	VFL/CD/LCD	Backlight Off/VF+LCD	
NP-FP30	95/80/95/75 min		85/105 min
NP-FP50	105/90/105/85 min		105/135 min
NP-FP70	225/190/225/180 min		225/280 min
NP-FP90	400/345/400/320 min		400/500 min



MEMORY MODE‡

Still Pictures — Fine/Standard (Approx.)

Memory Stick Duo™ Media‡	Fine/Std		Memory Stick PRO Duo™ Media‡	Fine/Std	
	1152 x 864	640 x 480		1152 x 864	640 x 480
8MB	15/37	50/120	256MB	445/1000	1400/3550
16MB	30/74	96/240	512MB	900/2050	2850/7200
32MB	61/150	190/485	1GB	1800/4200	5900/14500
64MB	120/300	390/980			
128MB	245/600	780/1970			



MPEG MOVIE EX MODE‡

h=hours, m=minutes, s=second

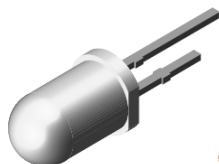
Memory Stick Duo™ Media‡	Email Mode (160x112)		Presentation Mode (320x240)		Memory Stick PRO Duo™ Media‡ (160x112)	Presentation Mode (320x240)	
	Mode	Mode	Mode	Mode		Mode	Mode
8MB	5m 20s	1m 20s	256MB	2h 50m 40s	42m 40s		
16MB	10m 40s	2m 40s	512MB	5h 41m 20s	1h 25m 20s		
32MB	21m 20s	5m 20s	1GB	11h 22m 40s	2h 50m 40s		
64MB	42m 40s	10m 40s					
128MB	1h 25m 40s	21m 20s					

ANEXO 5

Emisor IR TSUS5400


www.vishay.com
TSUS5400, TSUS5401, TSUS5402

Vishay Semiconductors

Infrared Emitting Diode, 950 nm, GaAs


94 8390

FEATURES

- Package type: leaded
- Package form: T-1 $\frac{3}{4}$
- Dimensions (in mm): \varnothing 5
- Leads with stand-off
- Peak wavelength: $\lambda_p = 950$ nm
- High reliability
- Angle of half intensity: $\phi = \pm 22^\circ$
- Low forward voltage
- Suitable for high pulse current operation
- Good spectral matching with Si photodetectors
- Compliant to RoHS Directive 2002/95/EC and in accordance to WEEE 2002/96/EC


RoHS
 COMPLIANT
GREEN
 (E-2008)**

DESCRIPTION

TSUS5400 is an infrared, 950 nm emitting diode in GaAs technology molded in a blue-gray tinted plastic package.

Note

** Please see document "Vishay Material Category Policy": www.vishay.com/doc?99902

APPLICATIONS

- Infrared remote control and free air transmission systems with low forward voltage and small package requirements
- Emitter in transmissive sensors
- Emitter in reflective sensors

PRODUCT SUMMARY

COMPONENT	I_e (mW/sr)	ϕ (deg)	λ_p (nm)	t_r (ns)
TSUS5400	14	± 22	950	800
TSUS5401	17	± 22	950	800
TSUS5402	20	± 22	950	800

Note

- Test conditions see table "Basic Characteristics"

ORDERING INFORMATION

ORDERING CODE	PACKAGING	REMARKS	PACKAGE FORM
TSUS5400	Bulk	MOQ: 4000 pcs, 4000 pcs/bulk	T-1 $\frac{3}{4}$
TSUS5401	Bulk	MOQ: 4000 pcs, 4000 pcs/bulk	T-1 $\frac{3}{4}$
TSUS5402	Bulk	MOQ: 4000 pcs, 4000 pcs/bulk	T-1 $\frac{3}{4}$

Note

- MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS ($T_{amb} = 25^\circ\text{C}$, unless otherwise specified)

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Reverse voltage		V_R	5	V
Forward current		I_F	150	mA
Peak forward current	$t_p/T = 0.5$, $t_p = 100 \mu\text{s}$	I_{FM}	300	mA
Surge forward current	$t_p = 100 \mu\text{s}$	I_{FSM}	2.5	A
Power dissipation		P_V	170	mW
Junction temperature		T_J	100	$^\circ\text{C}$
Operating temperature range		T_{amb}	- 40 to + 85	$^\circ\text{C}$
Storage temperature range		T_{stg}	- 40 to + 100	$^\circ\text{C}$
Soldering temperature	$t \leq 5$ s, 2 mm from case	T_{sd}	260	$^\circ\text{C}$
Thermal resistance junction/ambient	J-STD-051, leads 7 mm, soldered on PCB	R_{thJA}	230	K/W

Rev. 1.8, 24-Aug-11

1

Document Number: 81056

 For technical questions, contact: emittertechsupport@vishay.com

THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc?91000

BASIC CHARACTERISTICS ($T_{amb} = 25^\circ\text{C}$, unless otherwise specified)

PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Forward voltage	$I_F = 100$ mA, $t_p = 20$ ms	V_F		1.3	1.7	V
Temperature coefficient of V_F	$I_F = 100$ mA	TK_{VF}		- 1.3		mV/K
Reverse current	$V_R = 5$ V	I_R			100	μA
Junction capacitance	$V_R = 0$ V, $f = 1$ MHz, $E = 0$	C_j		30		pF
Temperature coefficient of ϕ_e	$I_F = 20$ mA	TK_{ϕ_e}		- 0.8		%/K
Angle of half intensity		ϕ		± 22		deg
Peak wavelength	$I_F = 100$ mA	λ_p		950		nm
Spectral bandwidth	$I_F = 100$ mA	$\Delta\lambda$		50		nm
Temperature coefficient of λ_p	$I_F = 100$ mA	TK_{λ_p}		0.2		nm/K
Rise time	$I_F = 100$ mA	t_r		800		ns
	$I_F = 1.5$ A	t_r		400		ns
Fall time	$I_F = 100$ mA	t_f		800		ns
	$I_F = 1.5$ A	t_f		400		ns
Virtual source diameter		d		2.9		mm

PACKAGE DIMENSIONS in millimeters

