



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE INGENIERÍA ELECTRÓNICA COMUNICACIONES

Tema:

**“CONSTRUCCIÓN DE UN ORDENADOR ELECTRÓNICO VEHICULAR
CON SISTEMA DE SEGURIDAD Y GPS UTILIZANDO RASPBERRY PI Y
HARDWARE LIBRE”**

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones

LÍNEA DE INVESTIGACIÓN: Sistemas Electrónicos

AUTOR: Cristian Roberto Jiménez Tenorio

TUTOR: Ing. Geovanni Danilo Brito Moncayo, Mg.

Ambato - Ecuador

Diciembre 2015

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el tema: **“CONSTRUCCIÓN DE UN ORDENADOR ELECTRÓNICO VEHICULAR CON SISTEMA DE SEGURIDAD Y GPS UTILIZANDO RASPBERRY PI Y HARDWARE LIBRE”**, del señor Cristian Roberto Jiménez Tenorio, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato Diciembre, 2015

EL TUTOR

Ing. Giovanni Danilo Brito Moncayo, Mg.

AUTORÍA

El presente Proyecto de Investigación Titulado: **“CONSTRUCCIÓN DE UN ORDENADOR ELECTRÓNICO VEHICULAR CON SISTEMA DE SEGURIDAD Y GPS UTILIZANDO RASPBERRY PI Y HARDWARE LIBRE”**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato Diciembre, 2015

Cristian Roberto Jiménez Tenorio

CC: 0503075392

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato Diciembre, 2015

Cristian Roberto Jiménez Tenorio

CC: 0503075392

APROBACIÓN DEL TRIBUNAL DE GRADO

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Carlos Serra Jiménez e Ing. Santiago Altamirano Meléndez, revisó y aprobó el Informe Final del Proyecto de Investigación titulado **“CONSTRUCCIÓN DE UN ORDENADOR ELECTRÓNICO VEHICULAR CON SISTEMA DE SEGURIDAD Y GPS UTILIZANDO RASPBERRY PI Y HARDWARE LIBRE”**, presentado por el señor Cristian Roberto Jiménez Tenorio de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ing. José Vicente Morales Lozada, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Carlos Serra Jiménez, Mg.

DOCENTE CALIFICADOR

Ing. Santiago Altamirano Meléndez, Mg.

DOCENTE CALIFICADOR

DEDICATORIA

El presente proyecto está dedicado a mis padres Roberto Jiménez y Enma Tenorio por todo el apoyo que me han brindado en el transcurso de mi vida profesional y guiarme por el camino del bien enseñándome valores, respeto, educación y sobre todo creer en mí, a mi tutor Ing. Geovanni Brito por haberme brindado todo su apoyo y aportado con sus conocimientos, a mis hermanos Franklín, Silvia y Diana por estar siempre apoyándome en todo momento, a mis queridos sobrinos, y una dedicatoria especial a mi abuelita por ser mi madre fiel, amada y querida.

Cristian Roberto Jiménez Tenorio

AGRADECIMIENTOS

A Dios por darme la fuerzas necesarias para salir adelante, y siempre guiarme en el sendero correcto, a mis padres, por su constante apoyo, por su infinito esfuerzo, comprensión, aliento y amor brindado día a día, a mi tutor que supo guiarme para la culminación de mi carrera profesional, a mis hermanos quienes siempre confiaron en mí y supieron apoyarme de una u otra manera para que este sueño se haga realidad, a mi abuelita por su inmenso amor y cariño de madre que me brinda día a día.

Cristian Roberto Jiménez Tenorio

ÍNDICE GENERAL

Tema:	I
APROBACIÓN DEL TUTOR	II
AUTORÍA	III
DERECHOS DE AUTOR	IV
APROBACIÓN DEL TRIBUNAL DE GRADO	V
DEDICATORIA	VI
AGRADECIMIENTOS	VII
ÍNDICE GENERAL	VIII
ÍNDICE DE FIGURAS	XII
ÍNDICE DE TABLAS	XVII
RESUMEN	XVIII
ABSTRACT	XIX
GLOSARIO DE TÉRMINOS	XX
INTRODUCCIÓN	XXIII
CAPÍTULO I	1
EL PROBLEMA	1
1.1 Tema	1
1.2 Planteamiento del problema	1
1.3 Delimitación de contenidos	3
1.3.1 Delimitación espacial	3
1.3.2 Delimitación temporal	3
1.4 Justificación.....	3
1.5 Objetivos	4
1.5.1 General.....	4
1.5.2 Específicos.....	5
CAPÍTULO II	6
MARCO TEÓRICO	6
2.1 Antecedentes Investigativos.....	6

2.2 Fundamentación	7
2.2.1 Sistema eléctrico vehicular	7
2.2.2 Encendido electrónico vehicular	9
2.2.3 Sistema de posicionamiento Global GPS	16
2.2.4 Sistema de Seguridad.....	18
2.2.5 Sistemas de Seguridad Vehiculares	18
2.2.6 Software Libre	30
2.2.7 Sistemas Operativos	31
2.2.8 Hardware libre	34
2.2.9 Pantallas táctiles	34
2.2.10 UDOO	41
2.2.11 Raspberry Pi	43
2.2.12 Plataforma Arduino	48
2.2.13 Comunicaciones inalámbricas	60
2.2.14 Comandos AT	63
2.2.15 Android	63
2.2.16 APP-Inventor	64
2.3 Propuesta de solución	65
CAPÍTULO III	66
METODOLOGÍA	66
3.1 Modalidad de la investigación	66
3.2 Población y muestra	66
3.3 Recolección de información	67
3.4 Procesamiento y análisis de datos.....	67
3.5 Desarrollo del proyecto	67
CAPÍTULO IV.....	69
DESARROLLO DE LA PROPUESTA	69
4.1 Análisis de los Sistemas de seguridad para vehículos actuales.....	69
4.1.1 Sistema de seguridad por contratación pública	69

4.1.2 Sistema de seguridad por contratación privada.	70
4.2 Análisis del Sistema.....	75
4.3 Comparación de las placas de la plataforma de Arduino	79
4.4 Comparación de los ordenadores de placa reducida	79
4.5 Selección del ordenador de placa reducida Raspberry Pi	80
4.6 Comparación de las pantallas	81
4.7 Selección del módulo Shield.....	82
4.8 Selección del Sistema Operativo utilizado en la placa Raspberry Pi	82
4.9 Diseño y construcción del Hardware	84
4.9.1 Control de entradas/salidas del ordenador.....	85
4.9.2 Diseño de circuitos de acoplamiento	86
4.9.3 Conexión entre Arduino Uno y Raspberry Pi.....	91
4.9.4 Conexión de Arduino Uno con Phone Shield Sim908 GSM/GPRS/GPS	91
4.9.5 Conexión de Raspberry Pi CON TONTEC 7" Raspberry Pi LCD Touch Screen Display TFT AT070TN90	92
4.9.6 Diseño de circuito de alimentación.....	93
4.10 Desarrollo del software del sistema. Modos de funcionamiento	94
4.10.1 Instalación del Sistema Operativo Rasbian en la Raspberry Pi.....	95
4.10.2 Instalación de Processing en Raspberry Pi	97
4.10.3 Configuración de Arduino Uno	99
4.10.4 Programación del Sketch en Processing.....	105
4.10.5 Creación de una Aplicación Móvil con App-Inventor	107
4.11 Instalación del sistema.....	118
4.11.1 Bloqueo Central del Vehículo	118
4.11.2 Sensores y actuadores	119
4.11.3 Ubicación del Ordenador Electrónico Vehicular con pantalla touch	120
4.11.4 Instalación de la Aplicación Móvil en un sistema operativo Android.....	121
4.12 Costo del proyecto.....	122
4.13 Implementación del ordenador vehicular en un automóvil Chevrolet Forza.....	123

CAPÍTULO IV.....	126
Conclusiones y Recomendaciones.....	126
Bibliografía y referencias.....	128
ANEXOS.....	131
ANEXO A.- Características técnicas De Arduino Uno.....	132
ANEXO B.- Características técnicas de Raspberry Pi Model B+	136
ANEXO C.- Características técnicas de TONTEC 7”.....	138
ANEXO D.-Características del Phone Shield Sim908 GSM/GPRS/GPS	140
ANEXO E.- Características técnicas de resistencia Pull-Up y Pull-Down	145
ANEXO F.- Código de programación para manejo de datos GPS a través de comandos AT.	146
ANEXO G.- Código de programación para control del sistema de bloqueo a través de comandos AT.....	149
ANEXO H.- Código de programación para establecer la comunicación entre Raspberry pi y Arduino mediante Processing.	153
ANEXO I.- Pruebas del funcionamiento del Ordenador Vehicular	160

ÍNDICE DE FIGURAS

Figura 2.1 Elementos baterías acumuladores.....	8
Figura 2.2 Motor de arranque.....	9
Figura 2.3 Encendido electrónico con generador de Impulsos.....	11
Figura 2.4 Encendido con generador Hall.....	12
Figura 2.5 Encendido electrónico integral.....	13
Figura 2.6 Encendido electrónico con condensador.....	14
Figura 2.7 Dispositivo GPS.....	18
Figura 2.8 Dispositivo de inmovilizador transponder.....	20
Figura 2.9 Llave con un chip insertado en el mango.....	21
Figura 2.10 Unidad de mando de inmovilizador.....	22
Figura 2.11 Diseño de bloques de inmovilizador con teclado.....	24
Figura 2.12 Módulo de alarma de un vehículo.....	26
Figura 2.13 Panel auto chevystar.....	27
Figura 2.14 Dispositivo tracker GPS.....	28
Figura 2.15 Logo Hunter SOS.....	29
Figura 2.16 Funcionamiento Hunter SOS.....	30
Figura 2.17 Logo de software libre.....	34
Figura 2.18 TFT LCD de pantalla táctil Shield Kit.....	35
Figura 2.19 Tontec 7 Raspberry Pi LCD Touch Screen Display TFT AT070TN90.....	37
Figura 2.20 Placa controladora AT070TN90.....	38
Figura 2.21 Digital pantalla del IPS (1280 x 800) kit.....	39
Figura 2.22 Placa controladora VS-TY2662-V1.....	40
Figura 2.23 Placa UDOO.....	42
Figura 2.24 Módulo Raspberry pi.....	43
Figura 2.25 Raspberry Pi 1 Model A+.....	45

Figura 2.26 Raspberry Pi 2 Model B +.....	46
Figura 2.27 Raspberry Pi 2 Modelo B.....	47
Figura 2.28 Aspecto de la placa Arduino.....	48
Figura 2.29 Placa Arduino UNO.....	49
Figura 2.30 Placa Arduino Leonardo.....	50
Figura 2.31 Placa Arduino Mega 2560.....	51
Figura 2.32 Placa Arduino Ethernet.....	52
Figura 2.33 Placa Nabduino 1.2.....	53
Figura 2.34 Entorno de desarrollo Arduino.....	54
Figura 2.35 Selección de caracteres de fin de línea en la ventana "Monitor Serial".....	56
Figura 2.36 Ejemplo de estructura modular con varias shields para Arduino.....	57
Figura 2.37 Módulo GPRS+GPS Quadband para Arduino y Raspberry PI (SIM 908)..	57
Figura 2.38 Módulo 3G/GPRS+GPS para Arduino/Raspberry PI.....	58
Figura 2.39 Shield sim 900.....	59
Figura 2.40 Shield GSM/GPRS/GPS Sim 908.....	60
Figura 2.41 Arquitectura de red del sistema GSM.....	61
Figura 2.42 Arquitectura de red del sistema GPRS.....	62
Figura 2.43 Diagrama de bloques del funcionamiento de Android.....	64
Figura 2.44 Pasos para crear una aplicación Android en App-Inventor.....	65
Figura 4.1 Diagrama de conexión de un sistema de seguridad vehicular.....	70
Figura 4.2 Diagrama de conexión de alarma Chevistar.....	72
Figura 4.3 Descripción de las opciones de Chevystar.....	73
Figura 4.4 Diagrama de comunicación TRACKER GPS.....	73
Figura 4.5 Componentes principales que conforman el Ordenador Vehicular.....	76
Figura 4.6 Diagrama de flujo del comportamiento del Sistema propuesto.....	78
Figura 4.7 Diagrama de ubicación de los componentes del sistema.....	84

Figura 4.8 Distribución de los pines del Arduino Uno.....	85
Figura 4.9 Circuito de acoplamiento de los sensores de puertas.....	88
Figura 4.10 Circuito de acoplamiento con la señal de encendido.....	89
Figura 4.11 Diagrama para control del Bloqueo Central.....	90
Figura 4.12 Diagrama de conexión para abrir y cerrar los seguros de las puertas.....	90
Figura 4.13 Diagrama de control de encendido y apagado de la pantalla touch.....	91
Figura 4.14 Diseño esquemático entre el sistema de alarma y Arduino Uno.....	91
Figura 4.15 Conexión entre Raspberry Pi y Arduino Uno.....	92
Figura 4.16 Plataforma Arduino Uno + Shield Sim908 GSM/GPRS/GPS.....	92
Figura 4.17 Interface entre Raspberry Pi con LCD touch 7”	92
Figura 4.18 Acondicionamiento de alimentación para el Sistema.....	94
Figura 4.19 Sistemas operativos disponibles para la instalación.....	96
Figura 4.20 Menú principal de las opciones de configuración de Raspberry pi.....	96
Figura 4.21 Entorno grafico de Raspberry PI.....	97
Figura 4.22 Entorno de programación de Processing.....	99
Figura 4.23 Configuración de la placa Phone Shield SIM908 para uso del servicio....	100
Figura 4.24 Configuración de la placa Phone Shield Sim908 para uso del servicio GPS.....	100
Figura 4.25 Montaje de la Phone Shield Sim908 con Arduino Uno.....	101
Figura 4.26 Verificación de posibles errores en el ID de Arduino.....	102
Figura 4.27 Datos del GPS mostrados en el Monitor Serial del ID de Arduino.....	102
Figura 4.28 Verificación de posibles errores del sketch en el ID de Arduino.....	104
Figura 4.29 Datos de sensores mostrados en el Monitor Serial del ID de Arduino.....	105
Figura 4.30 Interface Usuario-Máquina, Máquina-Usuario.....	106
Figura 4.31 Sketch en el software de Processing.....	107
Figura 4.32 Entorno Grafico generado por el sketch en Processing.....	108

Figura 4.33 Diagrama de flujo del funcionamiento de la Aplicación Móvil.....	109
Figura 4.34 Versión beta de app inventor.....	110
Figura 4.35 Creación de un nuevo proyecto en App Inventor Beta.....	110
Figura 4.36 Pantalla de diseño de App-Inventor beta.....	111
Figura 4.37 Componentes agregados aun sin configurar.....	111
Figura 4.38 Configuración del componente ActivityStarter1.....	112
Figura 4.39 Pantalla Parcial mostrando los componentes Personalizados.....	113
Figura 4.40 Vista del Android blocks Editor.....	115
Figura 4.41 Comportamiento del Componente Button ACTIVAR.....	115
Figura 4.42 Comportamiento del Componente Button DESACTIVAR.....	116
Figura 4.43 Comportamiento del Componente Button Obtener Ubicación.....	117
Figura 4.44 Comportamiento del Componente Message.Received.....	117
Figura 4.45 Uso de la función Make text para almacenar el mensaje recibido en la variable Text.....	117
Figura 4.46 Lógica del componente Extraer_Ubicacion.....	118
Figura 4.47 Lógica del componente Buscar_en_Google_Maps.Click.....	119
Figura 4.48 Ubicación del bloqueo Central.....	120
Figura 4.49 Montaje de los pulsadores de puertas.....	120
Figura 4.50 Conexión de los Seguros centralizados.....	121
Figura 4.51 Ubicación de la pantalla touch 7"	121
Figura 4.52 Generación del archivo ejecutable .apk de la aplicación Móvil.....	122
Figura 4.53 Archivo Tesis_Ordenadorl_Vehicular.apk copiado en el Smartphone.....	122
Figura 4.54 Aplicación Tesis_Ordenador_Vehicular en el Smartphone.....	123
Figura 4.55 Instalación de los Pulsadores en las puertas y compartimientos.....	125
Figura 4.56 Vista Frontal del Ordenador Vehicular.....	125
Figura 4.57 Vista posterior del Ordenador Electrónico Vehicular.....	126

Figura 4.58 Desactivación del Bloqueo Centralizado en la pantalla Touch.....	161
Figura 4.59 Vehículo con alguna puerta o compartimiento abierto.....	161
Figura 4.60 Vehículo con las puertas y compartimientos cerrados.....	162
Figura 4.61 Activación del sistema de bloqueo de acceso al vehículo.....	162
Figura 4.62 Desactivación del sistema de bloqueo de acceso al vehículo.....	163
Figura 4.63 Reacción del Ordenador cuando alguna puerta lateral es abierta.....	163
Figura 4.64 Reacción del Ordenador cuando alguna puerta lateral es abierta.....	164
Figura 4.65 Localización del vehículo con datos del GPS.....	164

ÍNDICE DE TABLAS

Tabla 2.1 Descripción de los pines de la 3.5 " TFT LCD de pantalla táctil.....	37
Tabla 2.2 Descripción de los pines de la tarjeta AT070TN90.....	38
Tabla 2.3 Descripción de la placa controladora VS-TY2662-V.....	40
Tabla 4.1 Tabla de las Características de las Series Arduino.....	79
Tabla 4.2 Tabla comparativa de los Ordenadores de placas reducidas.....	80
Tabla 4.3 Tabla comparativa de la Placas Raspberry Pi.....	81
Tabla 4.4 Tabla comparativa de las pantallas Tóuch compatibles para Raspberry Pi.....	81
Tabla 4.5 Tabla comparativa de los Shields GSM/GPRS para Arduino.....	82
Tabla 4.6 Tabla descriptiva de los S.O. disponibles para Raspeberry Pi.....	83
Tabla 4.7 Tabla de función de los pines del Arduino Uno.....	86
Tabla 4.8 Tabla de Rango de voltajes de los Modulos del Sistema.....	94
Tabla 4.9 Lista de campos para configurar el componente ActivityStarter1.....	113
Tabla 4.10 Listado de componentes usados en la aplicación Móvil.....	115
Tabla 4.11 Costo de los materiales para el proyecto.....	125

RESUMEN

El presente proyecto de investigación detalla la creación e implementación de un Sistema de monitoreo y control de los dispositivos de alarma que brindan seguridad a un automóvil, con un Sistema de rastreo GPS, denominado Ordenador Electrónico Vehicular basado en Hardware y Software libre. La implementación de este sistema utiliza una lógica similar a los sistemas de alarmas tradicionales, el método de operación está basado en la tecnología que actualmente la sociedad dispone, mediante una clave de acceso a través de un “LDC Touch” incorporada dentro del vehículo, y un control de mando a través de una “Aplicación Móvil” para Sistema Operativo “Android”.

El Ordenador Electrónico Vehicular está constituido por un controlador Arduino Uno y Ordenador de placa reducida “Raspberry Pi” los cuales ejecutan acciones que el usuario seleccione.

El servicio GPS y GSM ofrece el módulo “Phone Shield Sim908” el cual interactúa con “Arduino Uno” y así realizar la comunicación mediante “Comando AT” hacia la “Aplicación Móvil”

Palabras Claves

Comandos AT, Android, Aplicación Móvil, Raspberry Pi, Phone Shield Sim908, Arduino Uno, LCD Touch.

ABSTRACT

This research project details the creation and implementation of a monitoring and control alarm devices that provide security to a car with a GPS tracking system , called electronic computer based on hardware and software Vehicular free . The implementation of this system uses a similar logic to the traditional systems of alarms, the method of operation is based on the technology that currently the company has, through a password through a "LDC Touch" built into the vehicle, and remote control through a "Mobile Application" for operating system "Android".

Vehicle Electronic Computer consists Arduino One computer controller and reduced plate "Raspberry Pi" which perform actions that the user selects.

The GPS and GSM service offers the "Phone Sim908 Shield" module which interacts with "Arduino Uno" and thus make communication through "AT Command" to the "Mobile Application"

Keywords:

AT commands, Android, Mobile Application, Raspberry Pi, Sim908 Phone Shield
Arduino Uno

GLOSARIO DE TÉRMINOS

HLR.- Registro de posición base

VLR.- Registro de posición visitante

EIR.- Registro de identificación del equipo

AUC.- El Centro de autenticación

SGSN.- Nodo de soporte de servicios GPRS

GNU.- Gnu Not Unix, que es un sistema operativo similar a Unix.

KDE.- (K Desktop Enviroment). Es un proyecto de software libre con el objetivo de ser un poderoso entorno de escritorio.

GNOME.- Es un entorno de escritorio y de infraestructura de desarrollo para sistemas operativos GNU/Linux, Unixy derivados Unix como, BSD o Solari; compuesto enteramente por software libre.

XFCE.- Es un entorno de escritorio libre para sistemas tipo Unix como GNU/Linux, BSD

TCP.- Protocolo de Control de Transmisión.

IP.- Protocolo de Internet

PPP.- (Point to point protocol – Protocolo punto a punto). El PPP es un protocolo de nivel de enlace para hacer conexión entre dos puntos.

LCD.- Pantalla de Cristal Líquido.

TFT.- Transmisión de efecto Campo.

SPI.- (Serial Peripheral Interface Bus o bus serial de interfaz de periféricos) es un estándar de datos seriales.

HDMI.- High Definition MultiMedia Interface.

VGA.- Arreglo Grafico De Video.

ARM.- Familia de microprocesadores producidos por la empresa ARM Holding.

CPU.- Unidad Central de Procesos.

GPIO.- General Purpose Input/Output, Entrada/Salida de Propósito General.

SD.- Seguridad Digital.

BSS.- Estación Base.

BSC.- Estación Base de Control.

BTS.- Estación Base de Sincronismo.

Ordenador.- Máquina electrónica capaz de almacenar información y tratarla automáticamente mediante operaciones matemáticas y lógicas controladas por programas informáticos.

Electrónica.- Parte de la física que estudia los cambios y los movimientos de los electrones libres y la acción de las fuerzas electromagnéticas y los utiliza en aparatos que reciben y transmiten información.

Sistema.- Conjunto ordenado de normas y procedimientos que regulan el funcionamiento de un grupo o colectividad.

Seguridad.- Ausencia de peligro o riesgo.

GPS.- Sistema americano de navegación y localización mediante satélites.

Raspberry PI.- Ordenador de placa reducida o (placa única) (SBC) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi.

Hardware.- Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

Microcontroladores.- Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

Sensor.- Dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

Aplicación.- Programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.

INTRODUCCIÓN

El presente proyecto de Investigación analiza los sistemas de seguridad para vehículos que existen actualmente en el país conjuntamente con los sistemas eléctricos y electrónicos internos de los automotores que permiten su manipulación, movilización y bloqueos. El análisis de los sistemas previamente mencionados permiten diseñar y construir un Ordenador Electrónico vehicular que mantiene un control y monitoreo a distancia mediante una aplicación Móvil basado en tecnología de comunicación AT y directa a través de una interface táctil instalada dentro del vehículo, posee un sistema incorporado por hardware y software libre el cual no se encuentra inscrito a ninguna suscripción para su uso.

El proyecto se llevó a cabo con el desarrollo de cinco capítulos, los mismos que se describen a continuación:

Capítulo I: Se analiza y se plantea el problema, posteriormente se realiza la delimitación tanto de contenidos como de espacio y tiempo, se justifica el proyecto para plantear los objetivos.

Capítulo II: Incluye los antecedentes Investigativos y la fundamentación teórica, que servirá para dar soporte científica a la investigación.

Capítulo III: Contiene la metodología empleada para la realización de este proyecto como: la Modalidad, la Recolección además se establece los pasos a seguir para el desarrollo del proyecto.

Capítulo IV: Se desarrolla el proyecto mediante el diseño e implementación del ordenador electrónico vehicular con sistema de seguridad y GPS Utilizando Raspberry Pi y Hardware Libre

Capítulo V: Finalmente se presenta las conclusiones y recomendaciones obtenidas durante el desarrollo del proyecto.

CAPÍTULO I

EL PROBLEMA

1.1 Tema

“CONSTRUCCIÓN DE UN ORDENADOR ELECTRÓNICO VEHICULAR CON SISTEMA DE SEGURIDAD Y GPS UTILIZANDO RASPBERRY PI Y HARDWARE LIBRE”.

1.2 Planteamiento del problema

La industria automotriz se encarga del diseño, desarrollo, fabricación, ensamblaje, comercialización y venta de automóviles livianos y pesados; en diferentes modelos, colores y con prestaciones de acuerdo a las exigencias del mercado. En los últimos años estas empresas se han enfocado en la creación de sistemas de protección y seguridad utilizando el avance tecnológico de la electrónica y las telecomunicaciones, creando computadoras inteligentes que vienen incorporadas en vehículos nuevos y en las versiones recientes, las cuales brindan mejor seguridad y protección de agentes externos, permitiendo la manipulación única por parte del usuario o dueño, debido a la confiabilidad y tecnología que estos sistemas brindan hacen que la implementación se realice en vehículos de lujo elevando su costo.

En el país el factor económico es uno de los mayores limitantes para la adquisición de vehículos equipados de alta tecnología los mismos que han sido fabricados con diferentes prestaciones, en cuanto a sistemas de vigilancia, seguridad y protección se refieren, actualmente los vehículos que circulan en el país no disponen de altos

estándares de seguridad, por el tema comercial y competitividad; dando lugar a que estos sean más susceptibles a robos.

En el Ecuador existen empresas dedicadas a la seguridad vehicular, algunas de estas ofrecen el servicio de localizadores GPS como por ejemplo GPS Tracker, Chevistar, entre otros, este servicio requiere un mantenimiento constante y directo desde la empresa contratista, además su función requiere que el usuario realice la petición hacia la central de monitoreo para solicitar la ubicación geográfica del vehículo, función que no puede realizar el usuario directamente.

Los sistemas de alarmas vehiculares comunes permiten tener un control directo desde el usuario hacia el automotor a través de un control de mando personal. Este servicio posee la función de sensores instalados en puntos estratégicos que al activarse envían una señal a una alarma sonora ubicada en el automotor de forma interna. El inconveniente de estos es que no poseen la función de aviso al dueño cuando se encuentra a una distancia inmune de ser escuchado.

Actualmente en la sociedad, la necesidad de transportarse y las facilidades que tienen las personas con los vehículos han mostrado también requerir de mayores exigencias y esto ha implicado desarrollo también de la economía puesto que los vehículos se utilizan también para el comercio o como servicio de transporte. El crecimiento de automotores en las vías se incrementa a medida que la población aumenta, esto ha ocasionado que además aumenten los robos de autos y la delincuencia busque métodos para evadir los sistemas de seguridad genéricos que se han mantenido en los últimos años.

Un vehículo posee un sistema de encendido electrónico el cual es activado por una llave que permite la circulación de corriente mediante un Switch para obtener el arranque y su funcionamiento con el objetivo de lograr la movilización de vehículo, de ahí surge la idea de realizar la creación sistema electrónico inteligente de control de encendido y seguridad que permita la operación propia por parte del propietario y evitar que personas no autorizadas para su manejo puedan manipular y causar la sustracción del automotor.

1.3 Delimitación de contenidos

Área académica: Física y Electrónica

Línea de Investigación: Sistemas Electrónicos

Sub líneas de Investigación: Sistemas Embebidos

1.3.1 Delimitación espacial

La presente investigación se desarrolló en la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato.

1.3.2 Delimitación temporal

La presente investigación se desarrolló desde Diciembre 2014 hasta julio 2015, a partir de su aprobación por el Honorable Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.4 Justificación

Actualmente la adquisición de un vehículo es sumamente necesaria no solo por la comodidad, exclusividad y el servicio que éste preste sino también que en general el mercado automotriz ha permitido incrementar la economía en el país; conjuntamente con la tecnología se han modificado los sistemas de seguridad, y eso ha dado lugar a que los fabricantes desarrollen también sensores, alarmas, seguros en general mucho más fiables de tal manera que un vehículo sea difícilmente un motivo de robo.

Las personas desean proteger y salvaguardar sus bienes al precio que sea, sin embargo las autopartes tienen incorporado un sistema de seguridad convencional que viene incorporado en los vehículos, la seguridad que brinda es limitada, para ello se propone realizar el presente trabajo de investigación e implementación el cual será proveer de un cerebro computarizado (tipo ordenador) el mismo que será visible en el vehículo; el uso puede ser también llevado a una aplicación a un teléfono móvil, este controlador facilita el acceso, manipulación y encendido del automotor además de poseer el servicio de posicionamiento global GPS propio, el mismo que no debe estar sujeto a suscripciones y es exclusivo para el cliente; esto permitirá elevar el sistema de

seguridad dando lugar a que esta herramienta disminuya el alto índice de delincuencia que en Ecuador se ha generado en los últimos años.

En el país existen varias empresas que brindan sistemas de seguridad para vehículos, en algunos casos se ofrece también el servicio de rastreo satelital con localizadores GPS, el problema de estos equipos es que están sujetos a suscripciones para su funcionamiento con costos elevados y no cuentan con software libre, los sistemas de alarmas audibles actuales permiten controlar y proteger los componentes del vehículo de todo daño o riesgo de robo utilizando un conjunto de elementos y sistemas ubicados en el automotor de forma interna, es por esto que este proyecto tendrá un impacto social y económico debido al costo de su implementación y su mantenimiento además de tener un control propio por parte del propietario.

Los sistemas eléctricos de encendido que poseen los vehículos que circulan actualmente en el país poseen elementos que son de fácil manipulación debido a que sus componentes son mecánicos. De ahí la importancia de implementar un sistema electrónico inteligente en un vehículo ya que se tendrá un mejor control de seguridad y encendido por parte de personas autorizadas por el propietario, de esta manera se beneficiaran a dueños o encargados de mantener la integridad y seguridad del automotor.

La construcción de un ordenador electrónico vehicular con sistema de seguridad y GPS utilizando Raspberry PI y hardware libre del presente proyecto es factible realizarlo ya que se cuenta con las herramientas e información necesaria para su elaboración.

1.5 Objetivos

1.5.1 General

- Construir un ordenador electrónico vehicular con sistema de seguridad y GPS utilizando Raspberry PI y hardware libre.

1.5.2 Específicos

- Analizar los sistemas de seguridad para vehículos actuales.
- Investigar sobre el sistema electrónico de encendido de un vehículo.
- Implementar el Ordenador electrónico vehicular con sistema de seguridad basado en comunicaciones inalámbricas.
- Crear una aplicación móvil para controlar el Ordenador electrónico básico del Vehículo.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes Investigativos

Mediante la indagación realizada en los archivos de la Biblioteca de la Facultad de Ingeniería en Sistemas Electrónica e Industrial, se obtuvo como resultado la existencia del proyecto que se relaciona a una parte de la presente investigación, realizado por Luis Eduardo Cando Tite, este sistema está basado en la creación de un bloqueo electrónico en el encendido del vehículo, para tener mayor nivel de seguridad al instante de encender el automóvil. La instalación de este sistema no es lo mismo que la instalación de una alarma, porque el encendido del vehículo se lo realiza mediante el ingreso de la huellas dactilares que tiene el modulo biométrico. [1]

En el repositorio virtual de la Escuela Politécnica Nacional Facultad de Electrónica y Telecomunicaciones se obtuvo como resultado un proyecto de Tesis de Grado de Ingeniería cuyo objetivo es implementar un dispositivo de bajo costo para proteger un vehículo, realiza por Malla Mora, Freddy Oswaldo; Chilig Quillupangui, Richard Javier, utilizando la telefonía celular como medio de comunicación entre el dispositivo y el usuario a través de un modem instalado en el interior del vehículo, el cual, en conjunto con el microcontrolador, se encarga de recibir comandos que permiten bloquear el automóvil y reportar al usuario, mediante mensajería instantánea, tanto el estado del dispositivo como también diversas condiciones de alarmas en el vehículo. [2]

En la Universidad Distrital Francisco José de Caldas de Bogotá, Colombia se obtuvo como resultado la existencia de un proyecto de Tesis de Grado de Ingeniería en el cual muestra el proceso de un sistema de localización, rastreo y monitoreo satelital de camiones mediante el uso de GPS y un dispositivo móvil Realizado por Gina Catalina Quevedo Quevedo, en el cual las coordenadas son enviadas desde dicho dispositivo a la central de monitores y son procesadas en varias etapas: captura, procesamiento, visualización y por último a la base de datos. [3]

2.2 Fundamentación

2.2.1 Sistema eléctrico vehicular

Es el encargado de repartir alimentación hacia todo el coche, sin él no se podría arrancar el auto o encender las luces. Está formado por:

- Sistema de generación y almacenamiento.
- Sistema de encendido.
- Sistema de arranque.
- Sistema de inyección de gasolina.
- Sistema de iluminación.
- Instrumentos de control. [4]

Componentes que conforman el sistema Eléctrico Vehicular

Existen varios componentes que conforman el sistema electrónico vehicular que son:

Batería de acumuladores.

Una batería está formada por varios acumuladores, y puede ser ácida o alcalina en función de la naturaleza del electrolito.

Por ejemplo, las baterías de los coches son ácidas, porque contienen un electrolito de ácido sulfúrico en el que se sumergen una placa de plomo metálico y otra de dióxido de plomo.

Cuando se agota el plomo o el dióxido de plomo la batería está gastada y para recargarla se hace pasar una corriente eléctrica de la placa positiva a la negativa mediante un

alternador o dinamo (o a veces conectándola al enchufe de casa), de manera que el sulfato de plomo se vuelve a descomponer en plomo en la placa negativa, y en la positiva en dióxido de plomo. [5]

En la figura 2.1 se muestra un esquema de una batería con todas sus partes.

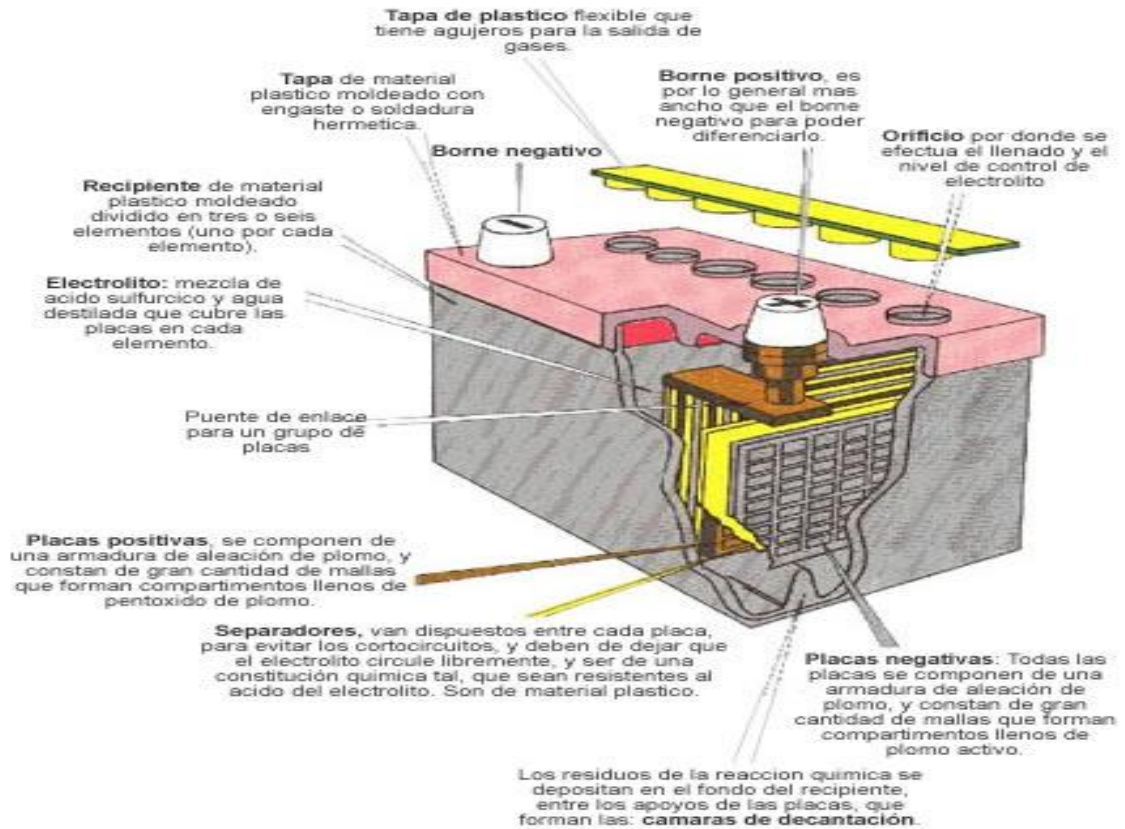


Figura 2.1 Elementos baterías acumuladores.
Fuente: Baterías y Acumuladores, Tecnología. [5]

Circuito de arranque

El motor de combustión interna no es capaz de arrancar por sí solo. Los motores de automóviles (tanto de encendido a chispa como diésel) son arrancadores por un motor eléctrico pequeño pero potente. Este motor es llamado un motor de arranque.

La batería envía corriente al motor de arranque cuando el operador gira el interruptor de encendido (llave de encendido) para arrancar. Esto hace que un engranaje de piñón en el motor de arranque engrane con los dientes del engranaje de anillo, girando con ello el cigüeñal del motor para el arranque.

El circuito de arranque típico consiste en la batería, el motor de arranque y el mecanismo de accionamiento, el interruptor de encendido, el relé de arranque o solenoide, un interruptor de seguridad neutro (transmisiones automáticas), el cableado para conectar estos componentes. [6]

Motor de arranque

Máquina eléctrica auxiliar que, aprovechando la energía eléctrica de la batería, produce energía mecánica y la transmite después al motor determinando la puesta en marcha.

Los motores térmicos no pueden ponerse en movimiento por sí solos, sino que deben ser ayudados hasta que alcanzan un régimen de rotación superior al crítico, en el cual el funcionamiento del motor es inestable y existe el peligro de que se «cale», es decir que se pare el motor. Este régimen mínimo, que permite el funcionamiento regular del motor, es suficiente para que el par desarrollado por el propio motor pueda vencer la resistencia de los rozamientos internos; depende de las características del motor, de la temperatura exterior, de la carburación y de la viscosidad del lubricante. [7]



Figura 2.2 Motor de arranque

Fuente: http://commons.wikimedia.org/wiki/File:Automobile_starter.JPG

2.2.2 Encendido electrónico vehicular

El encendido electrónico o CDI (Capacitor Discharge Ignition), es un sistema de encendido para motores de ciclo Otto tanto de dos tiempos (2T) como cuatro tiempos (4T) en el cual la función de interrumpir la corriente del primario de la bobina para generar por autoinducción la alta tensión necesaria en la bujía no se hace por medios mecánicos como en el sistema de ruptor o platinos, sino mediante uno o varios transistores. [8]

Ventajas:

- Ausencia de desgastes debido a la ausencia de leva para abrir y cerrar los platinos u otras piezas mecánicas.
- Se posibilita el aumento de la corriente de primario lo cual beneficia el secundario y por tanto la energía disponible para la chispa en la bujía.
- No se desajusta, por lo que no requiere puesta a punto.

Al ser electrónico, la eficiencia del chispazo en las bujías es más precisa que el sistema convencional. [8]

Circuito electrónico.

Las cargas eléctricas que constituyen una corriente eléctrica pasan de un punto que tiene mayor potencial eléctrico a otro que tiene un potencial inferior. Para mantener permanentemente esa diferencia de potencial, llamada también voltaje o tensión entre los extremos de un conductor, se necesita un dispositivo llamado generador (pilas, baterías, dinamos, alternadores...) que tome las cargas que llegan a un extremo y las impulse hasta el otro. El flujo de cargas eléctricas por un conductor constituye una corriente eléctrica. [9]

Encendido electrónico con generador de impulsos.

Uno de los sistemas totalmente electrónicos es el de generador de impulsos, donde el ruptor ha sido sustituido por un sistema capaz de engendrar golpes de corriente y entregarles al formador de impulsos, donde son amplificados y transmitidos al circuito de mando de un transistor que realiza la conmutación. Los impulsos suelen ser engendrados mediante «generadores de inducción» o «generadores fotoeléctricos».

El generador de inducción dispone de una rueda de aspas (Fig. 2.3.) de acero magnético, que produce durante su rotación una variación del flujo magnético del imán permanente C. En la bobina B se induce de esta forma una tensión, que se hace llegar al formador de impulsos.

La rueda A tiene tantas aspas como cilindros el motor y a medida que se acerca cada una de ellas a la bobina de inducción, la tensión va subiendo cada vez con más rapidez, hasta alcanzar su valor máximo cuando el aspa y la bobina están frente a frente. Al

alejarse el aspa siguiendo el giro, la tensión cambia muy rápidamente de sentido y asciende a su valor negativo máximo. Este cambio tiene lugar en el punto de encendido y el impulso así originado se hace llegar al formador de impulsos. [10]

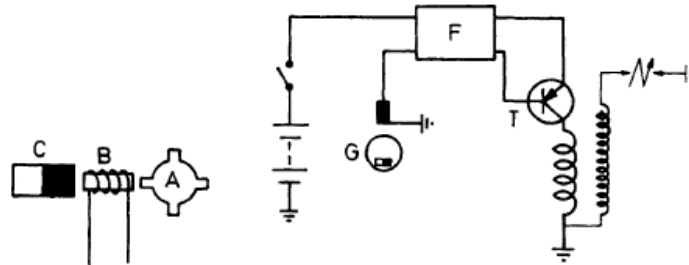


Figura 2.3 Encendido electrónico con generador de Impulsos

Fuente: Técnicas del automóvil – equipo eléctrico. [4]

Sistema electrónico de encendido por impulsos de inducción.

Este sistema de encendido dispone de los mismos elementos (batería, bobina, distribuidor, bujías) que los empleados en un encendido convencional, e igualmente, para la variación del punto de encendido utiliza los mismos dispositivos de avance centrífugo y por depresión, es decir, contrapesos y cápsula de vacío.

En la cabeza del distribuidor, es sustituido el clásico ruptor por un generador cuyos impulsos se hacen llegar a un componente adicional (módulo electrónico), que después de tratarlos convenientemente, determina el instante de corte de la corriente primaria en la bobina y, con ello, el salto de chipa en la bujía.

En el interior de esta unidad gira la rueda 4, llamada rotor, que forma parte del eje de mando del distribuidor, ocupando en él la posición de la leva en los distribuidores convencionales. [10]

Sistema electrónico de encendido con generador Hall.

En este sistema de encendido, el generador de impulsos basa su funcionamiento en el llamado «Efecto Hall» como se muestra en la siguiente figura 2.4

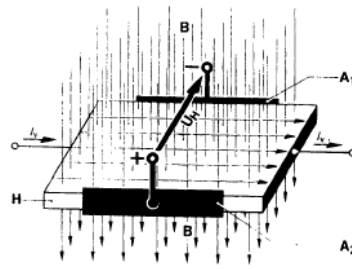


Figura 2.4 Encendido con generador Hall.

Fuente: Técnicas del automóvil – equipo eléctrico. [4]

Entre dos superficies de contacto A1 y A2, se dispone una capa de semiconductor H (capa Hall), que es atravesada por una corriente I_v . Al exponer la capa a la acción de un campo magnético B, perpendicular a la línea de unión de las superficies de contacto A1 y A2, se origina una tensión U_H entre las superficies de contacto, llamada tensión de Hall. Manteniendo constante la intensidad de la corriente I_v , la tensión U_H depende solamente del campo magnético B, cuyas variaciones periódicas en el ritmo de encendido pueden lograrse con facilidad, consiguiendo con ello una variación en la tensión de Hall en el ritmo de encendido, que será empleada en el gobierno del transistor de conmutación, con el que se logran los cortes de la corriente primaria en la bobina de encendido.

Esquemáticamente, el generador Hall empleado en los sistemas de encendido electrónico, se compone básicamente de una barrera magnética (parte fija) y un tambor obturador (parte rotatoria). [10]

Encendido Electrónico Integral.

Se entiende por esta designación, un sistema electrónico de encendido sin ruptor, que además suprime totalmente los dispositivos mecánicos de corrección de avance al encendido, a los que sustituye por componentes electrónicos.

Comparativamente con los sistemas electrónicos ya tratados, el encendido electrónico integral ofrece las mismas ventajas que ellos, e incluso ciertas mejoras. Un calculador electrónico recoge informaciones de régimen y carga del motor de combustión y genera el correspondiente avance al encendido, que, en cualquier caso, será el más adecuado. Este mismo calculador trata igualmente las señales de mando para cortar o dar paso a la

corriente primaria en la bobina de encendido, determinando el instante en que debe saltar la chispa en la bujía que corresponda, a la que se hace llegar por medio de un distribuidor convencional.

Se ha representado esquemáticamente en la Figura 2.5. un sistema de encendido electrónico integral, cuyo principal componente es el módulo o calculador electrónico 7, que define la ley de avance al encendido esencialmente, en función de la velocidad de rotación del motor y de la depresión reinante en el colector de admisión. Las señales correspondientes son proporcionadas al calculador por el captador de posición 2, situado frente al volante de inercia 1 del motor de combustión, y por una cápsula manométrica 4 unida neumáticamente al colector de admisión. [10]

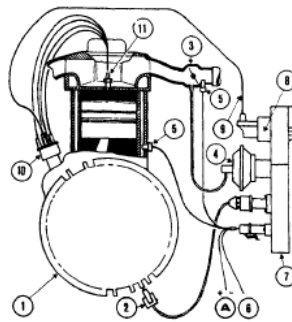


Figura 2.5 Encendido electrónico integral.
Fuente: Técnicas del automóvil – equipo eléctrico. [4]

Encendido electrónico por descarga de condensador.

Este sistema de encendido llamado también «encendido de tiristor», funciona basado en un principio totalmente distinto. Su característica esencial la constituye el hecho de que la energía de encendido es acumulada en el campo eléctrico de un condensador, donde la capacidad y tensión de su carga, determinan la magnitud de la energía acumulada. Para conseguir la chispa de encendido, se provoca la descarga del condensador.

El encendido de alta tensión por descarga de condensador, se aplica en la actualidad en todos aquellos casos que se plantean exigencias especiales en cuanto al régimen de giro del motor. Aquí la energía de encendido no se ve afectada de manera sensible con los regímenes altos de rotación del motor como se muestra en la figura. 2.6 [10]

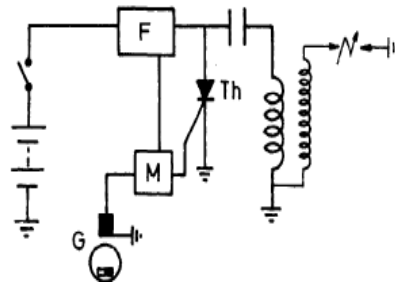


Figura 2.6 Encendido electrónico con condensador
Fuente: Técnicas del automóvil – equipo eléctrico. [4]

Comparación de los sistemas de encendido.

Se tomarán en cuenta las siguientes características:

- Grado en que la alta tensión y la energía de encendido dependen del número de chispas o de la velocidad de rotación del motor.
- Descenso de la alta tensión debido a derivaciones.
- Duración de la chispa de encendido.
- Vida útil de las piezas y facilidad de mantenimiento.

Las principales diferencias entre cada tipo de encendido son:

a. Encendido convencional.

Ofrece un buen comportamiento para exigencias normales (hasta 20.000 chispas). La ejecución técnica del ruptor, sometido a grandes cargas por la corriente primaria, constituye un compromiso entre el comportamiento de conmutación a baja velocidad de rotación y el rebote a alta velocidad. Derivaciones debidas a condensación de agua, suciedad, residuos de la combustión, etc., disminuyen la tensión disponible en medida muy considerable. [10]

b. Encendido con ayuda electrónica.

Existe una mayor tensión disponible, especialmente en los altos regímenes del motor. Utilizando juegos de contacto de reducido rebote, puede conseguirse que este sistema de encendido trabaje sin perturbaciones hasta aproximadamente 24.000 chispas por minuto. [10]

c. Encendido electrónico por generador de impulsos.

Los sistemas sin contactos satisfacen exigencias aún mayores. Los generadores de impulsos están exentos totalmente de mantenimiento. El número límite de chispas es de 30.000. Como consecuencia de la menor impedancia de estas bobinas, la subida de tensión secundaria es más rápida y, en consecuencia, la tensión de encendido es menos sensible a derivaciones eléctricas. [10]

d. Encendido totalmente electrónico.

Al quedar suprimidos los dispositivos mecánicos de los sistemas de avance al encendido, por la aplicación de componentes electrónicos, se obtiene mayor precisión en las curvas de avance, que pueden adaptarse fácilmente, cualquiera que sea su ley, cumpliendo perfectamente las actuales y severas normas en cuanto a polución. El mantenimiento de estos sistemas de encendido, es prácticamente nulo. [10]

e. Encendido por descarga de condensador.

Este sistema ofrece en el campo de altos regímenes del motor, una tensión todavía mayor y por tanto, aún más energía de encendido. La subida rápida en extremo de la tensión de encendido hace a la instalación insensible a derivaciones eléctricas. Sin embargo, la chispa de encendido es de muy breve duración. [10]

Verificación y control de los encendidos electrónicos.

La verificación en general de un circuito de encendido electrónico, sea cual fuere su tipo, comenzará por las bujías, que deben ser limpiadas y regladas en caso necesario. La limpieza de los cables de alta tensión, tapa de distribuidor y de bobina, es otra de las operaciones a realizar antes de iniciar cualquier verificación, dado que la alta tensión suministrada por estos sistemas de encendido es superior a la de los convencionales y se ve aumentado el riesgo de que el operario sufra descargas de alta tensión debidas a fugas o malas conexiones de los cables. Inmediatamente después, se procederá a verificar la puesta a punto del encendido en aquellos sistemas que lo requieran. [10]

2.2.3 Sistema de posicionamiento Global GPS

El sistema de Posicionamiento Global (GPS / Global Positioning System) es un proyecto espacial, desarrollado por el departamento de defensa de Estados Unidos, que está compuesto de una red de satélites que continuamente está transmitiendo información codificada mediante señales de radio de muy baja potencia. Lo que hace posible la identificación precisa de cualquier ubicación o lugar sobre la tierra midiendo las distancias desde los satélites. En origen, su objetivo era meramente militar y perseguía dotar a las tropas y dispositivos militares de una referencia espacial y temporal precisa. [11]

Características del GPS

El sistema GPS es un sistema de navegación satelital que asiste (enviándole una señal) a los navegadores y que consiste en una red de 32 satélites orbitando a 20.000 kilómetros de la tierra, en el espacio y en seis caminos orbitales diferentes. De ellos, solo 24 omiten señal, mientras que los otros 8, llamados de repuesto, son puestos en funcionamiento en caso de que falle alguno de los emisores. [11]

Características técnicas:

Sistema de satélites: Está formado por 24 unidades con trayectorias sincronizadas para cubrir toda la superficie del globo terráqueo. Más concretamente, repartidos en 6 planos orbitales de 4 satélites cada uno. La energía eléctrica que requieren para su funcionamiento la adquieren a partir de dos paneles compuestos de celdas solares adosados a sus costados. [11]

Estaciones terrestres: Envían información de control a los satélites para controlar las órbitas y realizar el mantenimiento de toda la constelación. [11]

Terminales receptores: Indican la posición en la que están; conocidas también como unidades GPS, son las que podemos adquirir en las tiendas especializadas. [11]

Segmento espacial

- Satélites en la constelación: 24 (4×6 órbitas).
- Altitud: 23.200 km.

- Período: 11 h 56 min (12 horas sidéreas).
- Inclinación: 55 grados (respecto al ecuador terrestre).
- Vida útil: 7,5 años.

Segmento de control (estaciones terrestres)

- Estación principal: 1
- Antena de tierra: 4
- Estación monitora (de seguimiento): 5
- Señal RF
- Frecuencia portadora:
- Civil - 1575,42 MHz (L1). Utiliza el Código de Adquisición Aproximativa (C/A).
- Militar – 1227,60 MHz (L2). Utiliza el Código de Precisión (P), cifrado.
- Nivel de potencia de la señal: –160 dBW (en superficie tierra).
- Polarización: circular dextrógira.

Exactitud

Posición: oficialmente indican aproximadamente 15 m (en el 95% del tiempo). En la realidad un GPS portátil mono frecuencia de 12 canales paralelos ofrece una precisión de 2,5 a 3 metros en más del 95% del tiempo. Con el WAAS/ EGNOS / MSAS activado, la precisión asciende de 1 a 2 metros. [11]

- Hora: 1 ns
- Cobertura: mundial
- Capacidad de usuarios: ilimitada
- Sistema de coordenadas:
- Sistema Geodésico Mundial 1984 (WGS84).
- Centrado en la Tierra, fijo.
- Integridad: tiempo de notificación de 15 minutos o mayor. No es suficiente para la aviación civil.
- Disponibilidad: 24 satélites (70%) y 21 satélites (98%). No es suficiente como medio primario de navegación. [11]



Figura 2.7 Dispositivo GPS

Fuente: <http://samubebe88.blogspot.com/2010/08/gps.html>

2.2.4 Sistema de Seguridad

Un sistema de seguridad se lo conoce como al conjunto de dispositivos colocados en distintos lugares estratégicos para detectar la presencia, irrupción, o invasión de un desconocido, individuo o agente externo lo que impide el acceso al sitio u objeto manipulado.

2.2.5 Sistemas de Seguridad Vehiculares

El sistema de seguridad de un vehículo, no solo se limita a la seguridad que presta el mismo en su estructura física, sino que va más allá ya que los factores externos que se presenten son de diferente índole y seguridad que quieren todas las marcas es conseguir la combinación perfecta de una seguridad activa y pasiva en sus modelos.

La seguridad activa se le atribuye todos los elementos técnicos que forman parte del automóvil, mientras que la seguridad pasiva son los elementos que comprenden formar parte de un sinnúmero de características que disminuyen el riesgo para que los usuarios sufran accidentes.

Más allá de que un automóvil cuente con todas las prestaciones técnicas, el correcto funcionamiento y mantenimiento garantizará la seguridad. Se han considerado tres tipos de seguridad que existen para vehículos que son:

a. Sistemas de seguridad antirrobo instalados de fábrica

La mayoría de los vehículos actuales, vienen equipados de fábrica con un sistema de seguridad antirrobo, que bloquea generalmente el arranque del motor. Existen varios tipos: [12].

- Llave Transponder
- Comando remoto Infrarrojo
- Teclado numérico

Inmovilizador Transponder

El mercado exige automóviles cada vez más seguros, tanto desde el punto de vista de la seguridad activa como de la pasiva, pero también preocupa la seguridad de los bienes depositados en el interior y la dificultad que opone para ser robado.

Por estas razones y otras de carácter legal en determinados países, los fabricantes de automóviles incorporan a los mismos sistemas que garanticen que solo el usuario autorizado pueda hacer uso del vehículo. Uno de los sistemas más utilizados es el inmovilizador de tipo transponder, debido a su alto grado de inviolabilidad y el que no requiere por parte del usuario, otra manipulación en su funcionamiento que no sea la del simple hecho de accionar la llave de arranque. [12]

¿Qué es un inmovilizador con transponder?

El inmovilizador con transponder es un sistema que solo permite el arranque del vehículo con las llaves autorizadas. Intentarlo con cualquier otra llave implica que el motor arranca, pero solo funciona durante algunos segundos (en la mayoría de los casos). La causa es que su funcionamiento se basa en el bloqueo de la unidad de mando del motor, que si no se dan las circunstancias adecuadas, no excita el relé de la bomba de combustible y no activa ni a los inyectores ni a la etapa de potencia del encendido (Algunos Ford son una excepción a lo último). Lógicamente, así es imposible el arranque del motor. [12]

Funcionamiento del inmovilizador con transponder

Existen diversas versiones de inmovilizador que presentan diferencias en algunos componentes, pero básicamente los principios de funcionamiento son muy parecidos.

En el sistema de inmovilizador con transponder, la llave incorpora un pequeño chip insertado en el mango de la misma y que emite un código por radiofrecuencia en el momento en que se acciona el contacto. Este código es captado por una antena o unidad lectora, normalmente ubicada en el conmutador de arranque (clausor).

El código captado por la unidad lectora es enviado a la unidad de mando del inmovilizador, que procede a compararlo con el que tiene memorizado. Simultáneamente la unidad de mando del motor envía a su vez a dicha unidad el código que le corresponde, y que también es comparado con el memorizado. La unidad de mando del inmovilizador autoriza el arranque a la unidad de mando del motor cuando los códigos emitidos por la llave y por la unidad de mando del motor coinciden con los almacenados en su memoria durante el proceso de grabación. En el caso de que uno de los dos códigos no coincida, la unidad de mando del motor pierde la autorización de arranque y el motor se para aproximadamente a los dos segundos de haberse iniciado el arranque. Estos dos segundos en los que el motor arranca son necesarios porque la autorización y el código circulan por un único cable entre las unidades de mando del inmovilizador y del motor. [12]



Figura 2.8 Dispositivo de inmovilizador transponder
Fuente: Manual Técnico, Inmovilizadores. [12]

Como es lógico, estos sistemas permiten la adaptación de varias llaves y la sustitución de algún componente averiado aplicando los procedimientos correspondientes. En

algunos casos será necesaria la utilización de equipos especiales y en otros bastará con el empleo de una llave maestra.

Como queda establecido, el inmovilizador no requiere para su funcionamiento ninguna intervención por parte del usuario del automóvil que no sea la de accionar la llave de contacto de la forma habitual, tanto es así, que en muchos casos el cliente no relaciona el hecho de que el motor no arranca con la presencia de este sistema. El mecánico lo debe tener claro: si el motor arranca bien, funciona bien durante dos segundos y se para enseguida, es muy probable que la causa radique en un inmovilizador con problemas. [12]

Elementos que intervienen en el inmovilizador transponder

El sistema inmovilizador con transponder está compuesto de los siguientes componentes:

Llaves con un chip insertado en el mango de las mismas, y que no presentan ninguna diferencia con otros tipos de llaves. En algunas marcas se emplea además una llave maestra o llave de programación que por lo general es de distinto color. Ninguna de estas llaves necesita pilas para su funcionamiento. [12].



Figura 2.9 Llave con un chip insertado en el mango
Fuente: Manual Técnico, Inmovilizadores. [12]

Unidad lectora, que actúa como fuente de alimentación y como antena. Tiene forma de anillo y está colocada en la parte superior del contactor de la llave de contacto.

Unidad de mando del inmovilizador, situada usualmente cerca de la columna de dirección, bajo el tablero. Existe la tendencia a incluirla dentro de otros componentes, como por ejemplo el cuadro de instrumentos o junto con la misma antena. [12]



Figura 2.10 Unidad de mando de inmovilizador

Fuente: Manual Técnico, Inmovilizadores. [12]

Unidad de mando del motor, que no presentan diferencias exteriores con respecto a las mismas unidades que no incorporan inmovilizador. En los casos en que el motor no tenga unidad de mando como ocurre en algunos diésel, se incorpora el elemento DDS. Este componente se instala en la bomba de inyección, sobre la válvula de pare de la bomba de gasoil y cumple las mismas funciones que la ECU. [12]

Sistema Inmovilizador con Criptotransponder

La versión de inmovilizador dotada de criptotransponder solo presenta una única diferencia con respecto al inmovilizador que no lo tiene. Esta diferencia radica en el tipo de comunicación que hay entre el transponder de la llave y la unidad del inmovilizador. Como se ha explicado, en las primeras generaciones el código enviado por el transponder de la llave es siempre el mismo, y este es comparado con el memorizado.

En el sistema criptotransponder se añade un nuevo código a la centralita y es memorizado por el transponder durante la inicialización del sistema. Peugeot es una de las marcas que utiliza este tipo de tecnología. Cada vez que se acciona el encendido, ya no se envía este código, sino una clave que se obtiene a partir de la combinación del mismo con un número aleatorio creado cada vez que hay una nueva transmisión. Dado que este código es imposible de ser conocido, es muy difícil hacer copias de las llaves, aumentando por lo tanto la seguridad del sistema. [12]

Inmovilizador con Comando Remoto Infra Rojo

Ciertos vehículos como algunos Renault y Rover utilizan un control remoto que emite una señal para habilitar el arranque del motor además de destrabar y trabar las puertas.

Estos controles remoto no deben confundirse con los que solo manejan la traba de las puertas. Una fuente de información importante para identificar que tipo de control posee un vehículo, es el manual del propietario que normalmente se entrega junto con la documentación al comprar el mismo. Incluso en dicho manual podrá encontrar información de suma importancia en lo referente a la puesta en marcha de emergencia en caso de extravío de la llave o el transponder. El control remoto puede estar incorporado en el mango de la misma llave (Renault) o puede ser un control separado (Rover y Chrysler).

En este caso no existe antena. La unidad lectora es un receptor del código infrarrojo a veces ubicado en el plafón del espejo retrovisor.

El sistema se completa con la central electrónica del inmovilizador que puede manejar o no el cierre centralizado y a distancia de las puertas.

EL sistema puede actuar sobre el bloqueo electrónico de la unidad de control (Renault) o solamente sobre el relay principal (caso Rover) que inhibe al motor de arranque.

Cada marca tiene su forma de avisar al conductor si el sistema está en modo bloqueo.

Funciones adicionales: Se refiere a que algunos sistemas manejan directa o indirectamente el sistema de cierre centralizado de puertas e incluso alarmas sonoras y visuales (Mediante el encendido de las luces y el funcionamiento intermitente de la bocina o sirena). [12]

Inmovilizador con Teclado Numérico

Es factible encontrar en algunos Peugeot 406 y 306 de origen francés, un teclado en el habitáculo cercano a la ubicación del conductor.

Por ejemplo el Peugeot 306 Turbo Diésel lo trae visible en la consola central mientras que el Peugeot 406 puede traerlo bajo el volante del lado izquierdo abriendo una tapa semejante a un acceso a la central portafusiles.

Existen también varios tipos de inmovilizador que combinan sus componentes que se detallarán en el Módulo Peugeot de este manual.

El propietario del vehículo debe ingresar tecleando un código de 4 dígitos cada vez que intente dar arranque al motor.

El bloqueo también es por inhibición de la válvula de pare en los diésel convencionales y por bloqueo de la unidad de control en los vehículos con inyección electrónica diésel o naftera. [12]

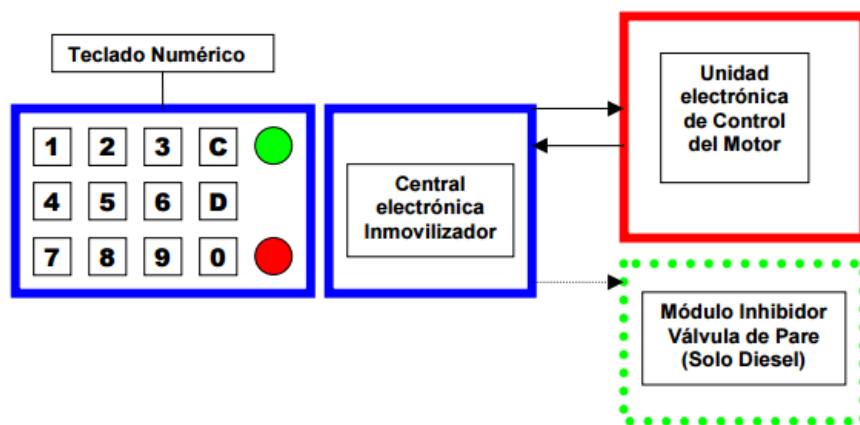


Figura 2.11 Diseño de bloques de inmovilizador con teclado
Fuente: Manual Técnico – Inmovilizadores. [12]

Las ventajas del sistema es que con solo recordar el código numérico es posible darle arranque con la llave convencional al motor.

Otra ventaja es la simplicidad del sistema ya que no hay receptores ni emisores de señales por radiofrecuencia como en el caso de los transponders o de infrarrojos como en el caso de los controles remoto.

Las desventajas son la necesidad de tener que ingresar el código cada vez que se desea poner en marcha el motor y que si olvida la clave de acceso será necesario reemplazar varios componentes costosos del sistema. [12]

b. Sistema de seguridad por contratación pública

Un sistema de alarma es un elemento de seguridad pasiva. Esto significa que no evitan una situación anormal, pero sí son capaces de advertir de ella, cumpliendo así, una función disuasoria frente a posibles problemas.

Por ejemplo:

La intrusión de personas. Inicio de fuego. El desbordamiento de un tanque. La presencia de agentes tóxicos. Cualquier situación que sea anormal para el usuario. Son capaces además de reducir el tiempo de ejecución de las acciones a tomar en función del problema presentado, reduciendo así las pérdidas.

Los sistemas de alarmas vehiculares están compuestos de los siguientes elementos y características:

Elementos que componen las alarmas del vehículo

- Cubre la rotura de los vidrios.
- Las puertas y el baúl se conectan con los pulsadores originales de la luz de cortesía o techo. En el caso del capot, se instala un pulsador adicional.
- Bloqueo de motor.
- LED disuasivo de alta luminosidad.
- Activación automática.
- Traba puertas.
- Valet
- Anti asalto Automático

Características de instalación

- Mazo pre cableado: "No hay que cablear relés afuera"
- 5 relés incorporados: Corte de corriente, Sirena, Luces, Traba Puertas, pistas de relés estañados reforzados.
- Cables volumétricos en fino coaxial negro. [13]

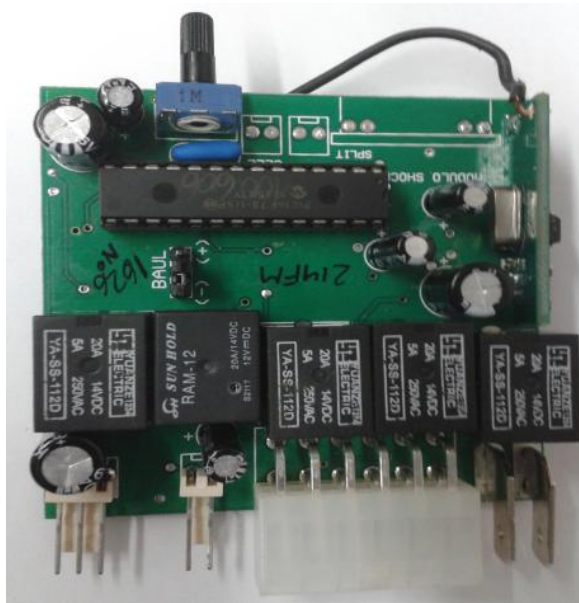


Figura 2.12 Módulo de alarma de un vehículo
Fuente: Alarmas Vehiculares - Alarmas Tausend. [14]

c. Sistemas de seguridad post robo privadas

Los sistemas de seguridad post robo privados son las siguientes empresas que se detallan a continuación las mismas que se han posicionado en el mercado local:

- Chevystar
- GPS Tracker
- Hunter SOS

CHEVYSTAR

Chevystar es un sistema de seguridad que está integrado por tecnología satelital y celular, el mismo que ofrece un sistema las 24 horas de rastreo y comunicación con el usuario. Este dispositivo viene en todos los vehículos livianos y pesados Chevrolet, como un valor agregado a su servicio. [14]

Funcionamiento:

En el panel de cada auto se encuentra ubicado un control tipo tablero debajo del retrovisor, que consiste en presionar un botón o llamado al Centro de asistencia al cliente y a la vez una serie de opciones de servicio que ofrece garantía y seguridad. [14]



Figura 2.13 Panel auto chevystar

Fuente:https://www.google.com.ec/search?q=fotos+chevystar&biw=1024&bih=471&source=lnms&tbn=isch&sa=X&ei=6DVCVaaU4SEsAXtsoC4Bw&ved=0CAcQ_AUoAg

GPS TRACKER

NET TRACKER S.A., es una empresa de origen argentino, dedicada al estudio, desarrollo y producción de soluciones tecnológicas de comunicaciones aplicables a todo tipo de vehículos terrestres y marítimos.

El resultado de importantes inversiones en investigación y desarrollo tecnológico permitió a G.C.S. S.A., incorporar en el mercado los equipos T 1800, T 2000 y ORBISAT 100; las más confiables herramientas de posicionamiento global satelital. Orientados a las necesidades del mercado y a los requerimientos de los clientes, brindamos servicios de capacitación, software de AVL, etc., con equipos de fabricación propia.

La empresa ha presentado y homologado ante la comunidad internacional el equipo ORBISAT 100. Este equipo, de características duales, combina las comunicaciones satelitales a las bases de AVL, utilizando indistintamente bandas de coberturas de telefonía celular GSM, o bien en caso de no obtener señal GSM, la comunicación se realiza mediante la utilización de la constelación de los satélites de la empresa ORBCOMM Inc. de U.S.A. y los módems satelitales DS 100 de su asociada Stellar Satellite Communications

Ambas empresas de Estados Unidos han homologado nuestro equipo ORBISAT 100 para su comercialización en todo el mundo. [15]



Figura 2.14 Dispositivo tracker GPS

Fuente:https://www.google.com.ec/search?q=fotos+chevystar&biw=1024&bih=471&source=lnms&tbn=isch&sa=X&ei=6DVCVauI4SEsAXtsoC4Bw&ved=0CAcQ_AUoAg#tbn=isch&q=gps+tracker

Especificaciones técnicas:

- Tamaño pequeño, impermeable
- Botón de SOS
- Alarmas de geo cercas, exceso de velocidad. micrófono espía.
- Ubicación vía SMS o vía web.
- GSM / GPRS / GPS
- Con Slot para memoria micro SD.

HUNTER SOS

Es un dispositivo GSM-GPS que puede ser monitoreado desde un celular o desde la página web. Especializado para personas que requieren asistencias especiales, con botón de emergencia y permite almacenar contactos importantes.

Tecnología

Hunter proporciona tecnología fija, móvil e inalámbrica; se ha mantenido líder en el mercado ecuatoriano con un servicio de altísima calidad en rastreo y ubicación de vehículos robados.

Diferencia entre Rastreo y Monitoreo

Rastrear con Lojack: Determina el lugar exacto de la posición del activo de riesgo, eliminando al máximo el perímetro de error. Ej. De una intersección, se determina el número exacto de vivienda, donde se encuentra el activo de riesgo.

- Uso exclusivo para rastreo de vehículos robados.
- Altísima efectividad en rastreo de vehículos robados.

Monitoreo con ubicación por GPS/GPRS: Da la zona aproximada donde se encuentra el vehículo, es una aproximación, indica la zona, pero no el lugar preciso donde se encuentra. Depende de la red de la compañía celular y de la cobertura que esta pueda darle. Pero en caso de estar en una intersección no puede indicarle el lugar exacto de vivienda, donde se encuentra el activo de riesgo, sólo el aproximado.

- Uso exclusivo para monitorear y administrar flotas.
- Tecnología GPS / GPRS.
- Monitoreo las 24 horas por web o software especial.
- Genera reportes de velocidad, paradas, zonas restringidas, etc. [16]



Figura 2.15 Logo Hunter SOS.

Fuente:https://www.google.com.ec/search?q=fotos+chevystar&biw=1024&bih=471&source=lnms&tbm=isch&sa=X&ei=6DVCVaaUI4SEsAXtsoC4Bw&ved=0CAcQ_AUoAg#tbm=isch&q=hunter+sos

Características técnicas y funcionamiento:



Figura 2.16 Funcionamiento Hunter SOS

Fuente: Tecnología . Hunter. [17]

2.2.6 Software Libre

Un programa informático es software libre si otorga a los usuarios todas estas libertades de manera adecuada. De lo contrario no es libre. Existen diversos esquemas de distribución que no son libres, y si bien podemos distinguirlos en base a cuánto les falta para llegar a ser libres, su uso bien puede ser considerado contrario a la ética en todos los casos por igual.

El software libre suele estar disponible gratuitamente, o al precio de coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar «software libre» a «software gratuito» (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente. Análogamente, el software gratis o gratuito incluye en ocasiones el código fuente; no obstante, este tipo de software no es «libre» en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa. [17]

2.2.7 Sistemas Operativos

Son aquellos sistemas que mantienen a uno o más equipos unidos a través de algún medio de comunicación (físico o no) con el objetivo primordial de poder compartir los diferentes recursos hardware y software.

Los sistemas operativos de red más ampliamente usados son: Novell NetWare, LAN Manager, Windows Server, UNIX, Linux, LANtastic, etc.

Al igual que un equipo no puede trabajar sin un sistema operativo, una red de equipos no puede funcionar sin un sistema operativo de red. Si no se dispone de ningún equipo que monte un sistema operativo de red, no existirá una gestión centralizada de recursos, por lo que la red se convertirá en lo que hemos denominado grupo de trabajo.

Sistema operativo Linux

Linux es un Unix libre, es decir, un sistema operativo, como el Windows o el MS-DOS (sin embargo, a diferencia de estos y otros sistemas operativos propietarios, ha sido desarrollado por miles de usuarios de computadores a través del mundo, y la desventaja de estos es que lo que te dan es lo que tu obtienes, dicho de otra forma no existe posibilidad de realizar modificaciones ni de saber cómo se realizó dicho sistema.), que fue creado inicialmente como un hobby por un estudiante joven, Linux Torvalds, en la universidad de Helsinki en Finlandia, con asistencia por un grupo de hackers a través de Internet. Linux tenía un interés en Minix, un sistema pequeño o abreviado del UNIX (desarrollado por Andy Tanenbaum); y decidido a desarrollar un sistema que excedió los estándares de Minix. Quería llevar a cabo un sistema operativo que aprovechara la arquitectura de 32 bits para multitarea y eliminar las barreras del direccionamiento de memoria.

Linux corre principalmente en PCs basados en procesadores 386/486/586, usando las facilidades de proceso de la familia de procesadores 386 (segmentación TSS, etc.) para implementar las funciones nombradas. [18]

Aplicaciones

Linux es un sistema operativo que permite obtener las siguientes aplicaciones y ventajas:

- Linux es uno de los sistemas operativos más robustos, estables y rápidos.
- Es austero: Linux funciona hasta en un 386.
- El manejo de la memoria de Linux evita que los errores de las aplicaciones detengan el núcleo de Linux.
- Linux es multitarea y multiusuario: Esta característica imprescindible está en Unix desde su concepción pero le llevó a Microsoft más de 20 años ofrecerlo en su sistema operativo de consumo.
- Linux soporta gran variedad de entornos gráficos (KDE, GNOME, XFCE...).
- Hay miles de programas libres para Linux, adaptados a muy diversos propósitos y disponibles en internet para usarlos con GNU/Linux.
- Linux permite navegar por Internet y conectar máquinas en red de manera natural (los protocolos TCP/IP o PPP por ejemplo, están incluidos como un módulo del básico del núcleo).
- Casi cualquier aplicación Unix puede usarse bajo Linux.
- Para Linux existe gran cantidad de documentación libre, aunque no siempre está traducida.
- Las libertades de copia y modificación permiten usar GNU/Linux para facilitar servicios sin depender de terceros.
- Al poder descargarse Linux de internet, el precio de las distribuciones debe mantenerse competitivo con el hágalo Vd. mismo y por lo tanto resulta un precio justo Pero no solamente el precio de adquisición de Linux es menor, el de implantación (debido a la posibilidad de emplearlo en tantas máquinas como se desee) también lo es, así como el Coste Total de Propiedad de Linux. [18]

Las distribuciones de Linux.

RED HAT: Red Hat es una empresa que forma parte de Linux, tiene dos tipos de distribución, un software comercial que funciona con licencia y tiene un costo, y otra

gratuita que se la utiliza con otro tipo de licencia. Su instalación es fácil puesto que el menú tiene una guía para que usuario pueda instalarla. [19]

DEBIAN: Es una distribución conocida en el mercado por su estabilidad y practicidad, además que es gratis pues no cuenta con licencia para los usuarios. Está formado por tipos de paquetes, y una potencia superior RPM. Cuenta con herramientas Alien lo que facilita la distribución, es de fácil instalación tiene una capacidad de acumular alrededor de 7000 paquetes. [19]

SUSE: Es una distribución de origen alemán, donde su principal componente es el RPM, incluye KDE y YAST. Es conocida por la alta demanda que tienen especialmente en el campo empresarial. [19]

SLACKWARE: Está compuesta por un paquete TGZ, su desventaja es que no tiene una configuración de programación puesto que es rudimentario su mantenimiento y configuración. [19]

MANDRAKE: Su origen es de Francia tienen elementos de red hat y se diferencia de esta en:

- Esta compilada para Pentium, con lo que todo funciona más fluido, en contra no podrás instalarla en un 486.
- Tiene el sistema de instalación más cómodo y automatizado, auto detectando y configurando la mayoría del hardware. [19]

GENTO: Esta distribución, apareció en el mercado recientemente, tiene un programa que permite instalar varios desde el código fuente y que asegura la máxima optimización, para su instalación es necesario tener mucho conocimiento de Linux. [19]

Comandos para Linux

Normalmente todos los programas permiten opciones, por ejemplo ls permitía -a -p -l, etc, las opciones se suelen colocar justo detrás del nombre del programa, antes que los otros parámetros, por ejemplo en el caso de ls sería ls -l /usr/doc, aunque suele

funcionar si lo colocas en otro sitio, lo normal es ahí, por facilidad a la hora de leer el comando y pensar lo que hace.

Se sugiere colocar junto todas opciones, poner ls -pla para usar las tres opciones, en ciertos casos especiales se podrá además omitir el guion, aunque lo normal es ponerlo.

También en algunos casos es posible utilizar opciones largas, así es lo mismo usar en ls -a que --all, en estos casos se utiliza un doble guion, para diferenciar --all de -a -l -l (-all). [19]

2.2.8 Hardware libre

Dado que el hardware tiene asociados a él costos variables directos, ninguna definición de software libre se puede aplicar directamente sin modificación. El término hardware libre se ha usado principalmente para reflejar el uso del software libre con el hardware y el lanzamiento libre de la información con respecto al hardware, a menudo incluyendo el lanzamiento de los diagramas esquemáticos, diseños, tamaños y otra información acerca del hardware. De todos modos, incluye el diseño del hardware y la distribución de los elementos en la tarjeta electrónica. [20]

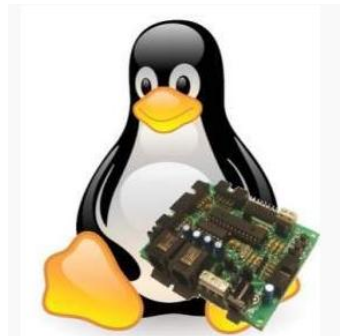


Figura 2.17 Logo de software libre

Fuente: Hardware libre, EcuRed, Cuba, Octubre de 2015

2.2.9 Pantallas táctiles

Una **pantalla táctil** (en inglés touch screen) es una pantalla que mediante un toque directo sobre su superficie permite la entrada de datos¹ y órdenes al dispositivo, y a su vez muestra los resultados introducidos previamente; actuando como periférico de

entrada y salida de datos, así como emulador de datos interinos erróneos al no tocarse efectivamente. Este contacto también se puede realizar por medio de un lápiz óptico u otras herramientas similares. Actualmente hay pantallas táctiles que pueden instalarse sobre una pantalla normal, de cualquier tipo (LCD, monitores y televisores CRT, plasma, etc.). [21]

Tipos de pantallas LCD a color con Touch.

TFT-LCD (Thin Film Transistor-Liquid Crystal Display, Pantalla de cristal líquido de transistores de película fina) es una variante de pantalla de cristal líquido (LCD) que usa tecnología de transistor de película delgada (TFT) para mejorar su calidad de imagen. Las LCD de TFT son un tipo de LCD de matriz activa, aunque esto es generalmente sinónimo de LCD. Son usados en televisores, visualizadores de pantalla plana y proyectores. En informática, los monitores de TFT han desplazado la tecnología de CRT, y están comúnmente disponibles en tamaños de 12 a 30 pulgadas. En el 2006 han entrado en el mercado de las televisiones. [22]

A continuación se muestran las diferentes tipos de pantallas:

3.5" TFT LCD de Pantalla Táctil Shield KIT + Estuche + Disipadores para Raspberry Pi

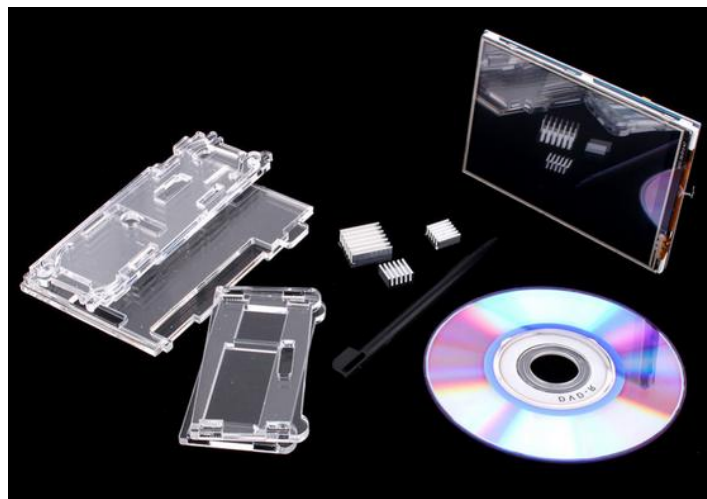


Figura 2.18 " TFT LCD de pantalla táctil Shield Kit

Fuente: <http://es.aliexpress.com/item/3-5-TFT-LCD-Shield-Touch-Screen-Kit-Display-Case-Heatsinks-for-Raspberry-Pi/32325005734.html>

Descripción:

1. Diseñado para Raspberry Pi Modelo B/B + una alternativa ideal solución para monitor HDMI.
2. 320 × 480 de resolución, mejor pantalla.
3. Conveniente interfaz hombre-máquina para Raspberry Pi, combinado con la energía portátil, DIY en cualquier lugar en cualquier momento.
4. Soporta Sistema Raspbian, permite a su sistema para:
 - Reproducir vídeos (soporta múltiples formatos, MP4 y así sucesivamente).
 - Tomar fotos tocando.
 - Software de ayuda (sistema interacción sin teclado/ratón). [23]

Parámetros clave:

- Tipo de lcd: TFT
- Lcd interfaz: SPI
- Tipo de pantalla táctil: resistiva
- Controlador de pantalla táctil: XPT2046
- Colores: 65536
- Luz de fondo: LED
- Resolución: 320*480 (pixel)
- Relación de aspecto: 2: 3
- Consumo de energía: TBD
- Luz de fondo: TBD
- Temperatura de funcionamiento. (O c): TBD. [23].

Interface

Tabla 2.1 Descripción de los pines de la 3.5 " TFT LCD de pantalla táctil

Pin No. .	Símbolo	Descripción
1, 17	3.3 V	Positivo de alimentación (3.3 V entrada de energía)
2, 4	5 V	Positivo de alimentación (5 V entrada de energía)
3, 5, 7, 8, 10, 12, 13, 15, 16	NC	NC
6, 9, 14, 20, 25	GND	Tierra
11	TP_IRQ	Panel táctil de interrupción, bajo nivel, mientras que el panel táctil detecta conmovedora
18	LCD_RS	Registro de instrucciones/datos selección
19	LCD_SI/tp_si	Entrada de datos SPI de lcd/panel táctil
21	TP_SO	SPI salida de datos de panel táctil
22	RST	Restablecer
23	Lcd_sck/tp_sck	SPI reloj de lcd/panel táctil
24	LCD_CS	LCD selección de chip, bajo activo
26	TP_CS	Panel táctil selección de chip, bajo activo

Fuente: <http://es.aliexpress.com/item/3-5-TFT-LCD-Shield-Touch-Screen-Kit-Display-Case-Heatsinks-for-Raspberry-Pi/32325005734.html>

TONTEC 7" Raspberry Pi LCD Touch Screen Display TFT AT070TN90 con tarjeta de mando de entrada VGA KIT pantalla táctil HDMI



Figura 2.19 Tontec 7" Raspberry Pi LCD Touch Screen Display TFT AT070TN90

Fuente: <http://es.aliexpress.com/item/7-Raspberry-Pi-LCD-Touch-Screen-Display-TFT-Monitor-AT070TN90-with-Touchscreen-Kit-HDMI-VGA-Input/1782782511.html>

En la figura 2.19 se muestra el Kit LCD para Raspberry Pi en la cual se detallan los siguientes componentes.

- LCD 7"
- Touch
- CD de Intalación
- Placa convertidora de señal HDMI a TTL
- Control Remoto
- Control Manual
- Cable de comunicación USB
- Bus de datos

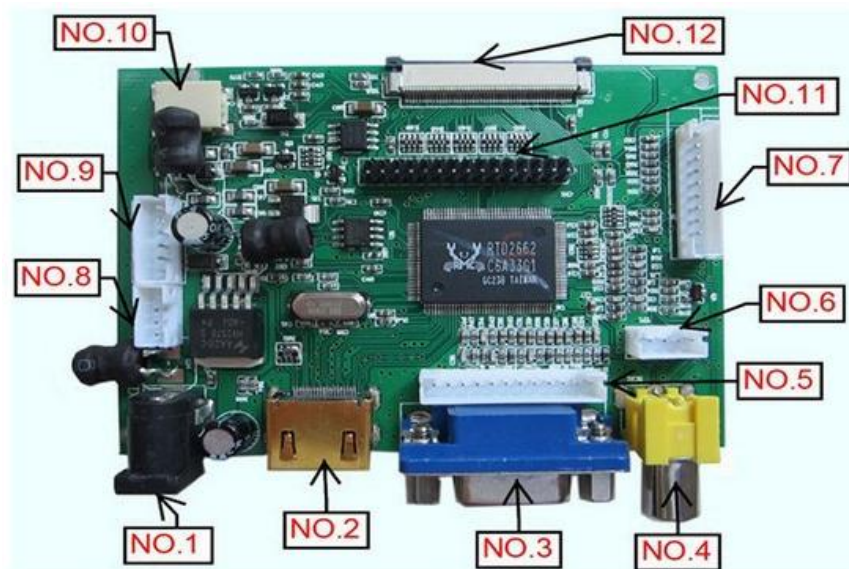


Figura 2.20. Placa controladora AT070TN90

Fuente: <http://es.aliexpress.com/item/7-Raspberry-Pi-LCD-Touch-Screen-Display-TFT-Monitor-AT070TN90-with-Touchscreen-Kit-HDMI-VGA-Input/1782782511.html>

Tabla 2.2 Descripción de los pines de la tarjeta AT070TN90

No.	Description	No.	Description
1	Power input	7	Key board &IR &LED indicator
2	HDMI signal Input	8	Power input
3	VGA signal input	9	Power and control signal for extra inverter board
4	AV1 signal input	10	Backlight voltage output connector
5	Extra VGA signal input	11	LVDS signal output connector
6	Extra AV1 and AV2 signal input	12	TTL signal output connector

Fuente: <http://es.aliexpress.com/item/7-Raspberry-Pi-LCD-Touch-Screen-Display-TFT-Monitor-AT070TN90-with-Touchscreen-Kit-HDMI-VGA-Input/1782782511.html>

Interfaces de tablero de conductor:

1. Power Abastecimiento (5 ~ 12V 2A)
2. HDMI
3. VGA Conector
4. AVI
5. AV1 + AV2
6. Keyboard
7. TTL Señal 50pin
8. LVDS señal 1/2 ch 6bit 8bit
9. 12V contraluz
10. Inverter Conector. [24].

Contenido del paquete:

- 1 * HDMI + VGA + 2AV Controller Junta VS- TY2662 -V1 (Apoyo automáticamente cambiar a AV2)
- 1 * 7inch 800 * 480 LCD con pantalla AT070TN90
- 1 * Interruptor de 5 botones y el panel de ajuste
- Kit de pantalla táctil 1 * 7 pulgadas. [24].

10.1" Digital pantalla del IPS (1280 X 800) + tablero de la impulsión para Pi / PCDUINO / CUBIEBOARD - NEGRO # N101ICG-L21



Figura 2.21. " Digital pantalla del IPS (1280 x 800) kit

Fuente: <http://es.aliexpress.com/item/10-1-Digital-IPS-Screen-1280-x-800-Drive-Board-for-Raspberry-Pi-Pcduino-Cubieboard-Black/1782794578.html>

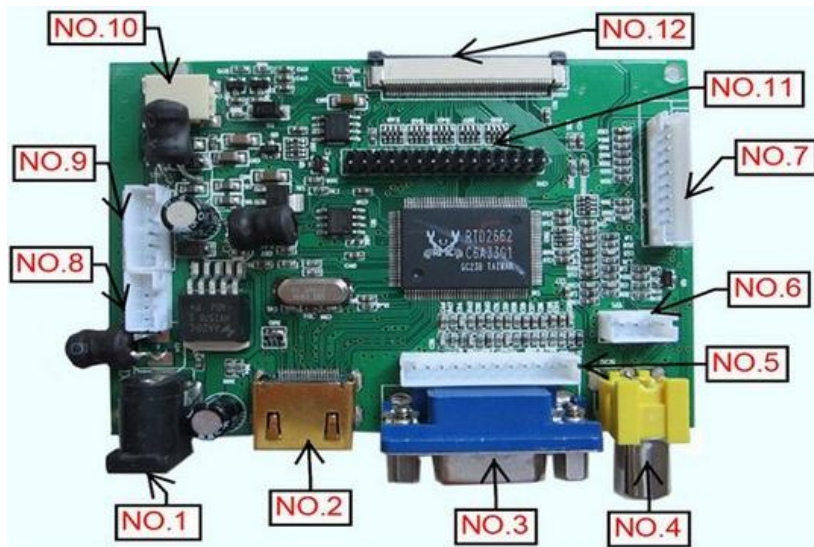


Figura 2.22. Placa controladora VS-TY2662-V1

Fuente: <http://es.aliexpress.com/item/10-1-Digital-IPS-Screen-1280-x-800-Drive-Board-for-Raspberry-Pi-Pcduino-Cubieboard-Black/1782794578.html>

Tabla 2.3. Descripción de la placa controladora VS-TY2662-V1

No.	Description	No.	Description
1	Power input	7	Key board & IR & LED indicator
2	HDMI signal Input	8	Power input
3	VGA signal input	9	Power and control signal for extra inverter board
4	AV1 signal input	10	Backlight voltage output connector
5	Extra VGA signal input	11	LVDS signal output connector
6	Extra AV1 and AV2 signal input	12	TTL signal output connector

Fuente: <http://es.aliexpress.com/item/10-1-Digital-IPS-Screen-1280-x-800-Drive-Board-for-Raspberry-Pi-Pcduino-Cubieboard-Black/1782794578.html>

Descripción

- Tamaño de la pantalla: 10.1 pulgadas
- Panel: a-Si tft-pantalla LCD
- En tasa distintiva: 1280x800
- Brillo del panel: 350 cd/m² (típico)
- Para que los grados: 800: 1 (típico)
- Pantalla Color: 262 K
- Tiempo de respuesta: 14/11 (típico) (Tr/TD) MS

- Ángulo visual: 85/85/85/85 (típico) (CR 10 o superior)
- Frecuencia de barrido: 60Hz
- Tipo de lámpara: WLED
- El tipo de señal: LVDS (1 ch, 8-bit). [25].

Interfaces de tablero de conductor:

1. Power alimentación (5 ~ 12 V 2a)
2. HDMI Conector
3. VGA
4. AVI
5. AV1 + AV2
6. Keyboard
7. TTL Señal 50Pin
8. LVDS Señal 1/2 6bit ch 8bit
9. 12 V luz de fondo
10. Inverter Conector. [25].

Contenido del paquete:

- HDMI + VGA + AV Controller Board VS-TY2662-V1 (apoyo cambiar automáticamente a av2)
- 1*10.1 pulgadas 1280*800 panel LCD N101ICG-L21
 - Botones de interruptor y panel de ajuste
- 1* Unidad de control remoto. [25].

2.2.10 UDOO

Es un único ordenador de a bordo que se puede utilizar tanto con Android y Linux, se combina con un procesador compatible con Arduino. Se trata de una placa de prototipado de gran alcance para el desarrollo de software y diseño; es fácil de usar y permite el desarrollo de proyectos con conocimientos mínimos de diseño de hardware. [26]



Figura 2.23 Placa UDOO

Fuente: http://shop.udoo.org/other/product/udoo-quad.html?__from_store=other&popup=no

Características

- CPU Freescale i.MX 6 ARM Cortex-A9 Quad core 1GHz
- GPU Vivante GC 2000 + Vivante GC 355 + Vivante GC 320
- Integrated accelerators for 2D, OpenGL® ES2.0 3D and OpenVG™
- Atmel SAM3X8E ARM Cortex-M3 CPU (same as Arduino Due)
- RAM DDR3 1GB
- 76 fully available GPIO: 62 digital + 14 digital/analog
- Arduino-compatible R3 1.0 pinout
- HDMI and LVDS + Touch
- 2 Micro USB (1 OTG type a+b)
- 2 USB type A (x2) and 1 USB connector (requires a specific wire)
- Analog Audio and Mic
- CSI Camera Connection
- on board Micro SD card reader (boot device)
- Input Voltage (recommended) 6-15 V
- Ethernet RJ45 (10/100/1000 MBit)
- WiFi Module
- SATA

2.2.11 Raspberry Pi

Es un proyecto de hardware libre que se ha hecho extremadamente popular entre los aficionados al mundo de la computación y la electrónica aunque su objetivo principal es equipar las escuelas de todo el mundo con ordenadores que cuestan menos de 30 dólares y que permiten, por tanto, democratizar el acceso a la tecnología y a la enseñanza de materias técnicas. Gracias Raspberry Pi, y la fundación que está detrás del proyecto, sería posible desplegar un aula de informática en centros educativos de todo el mundo sin que el presupuesto sea una barrera de entrada. [27]



Figura 2.24 Módulo Raspberry pi

Fuente: <http://alt1040.com/2013/04/usos-de-raspberry-pi>

Hardware de la Raspberry Pi

La aceleración por hardware para la codificación de vídeo (H.264) se hizo disponible el 24 de agosto de 2012, cuando se informó que la licencia permitiría su uso gratuitamente; antes se pensó en anunciarlo cuando se lanzara el módulo de cámara. También se puso a la venta la capacidad para poder usar la codificación-decodificación de MPEG-2 y Microsoft VC-1. Por otro lado se hizo saber que el ordenador soportaría CEC, permitiendo que pudiera ser controlado mediante un mando a distancia de televisión.

El 5 de septiembre de 2012, se anunció una revisión 2.0 de la placa, que ofrecía un pequeño número de correcciones y mejoras, como unos agujeros de montaje, un circuito para hacer reset, soporte para depuración JTAG, etc.

El 15 de octubre de 2012, la fundación anunció que todos los Raspberry Pi Modelo B serían enviados a partir de ese momento con 512 MB de RAM en vez de 256 MB. [27]

Software de la Raspberry Pi

El Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse.

Slackware ARM (también llamada ARMedslack) versión 13.37 y posteriores arranca sin ninguna modificación. Los 128-496 MiB de memoria RAM disponible en la Raspberry Pi, cubren los necesarios 64 MiB de RAM para arrancar esta distribución en sistemas ARM y i386 sin usar interfaz gráfica (el administrador de ventanas Fluxbox que funciona bajo X Window System requiere 48 MiB de memoria RAM adicional). Por otro lado, se están creando distribuciones más específicas y ligeras como IPfire (distribución para ser usada como firewall), o OpenELEC y Raspbmc (distribuciones con el centro multimedia XBMC). [27]

Tipos De Raspberry Pi

a. Raspberry Pi 1 Modelo A+

El modelo A + es la variante de bajo costo de la Raspberry Pi. Sustituyó el original modelo A en noviembre de 2014. En comparación con el modelo A tiene:

- **Más GPIO.** La cabecera GPIO ha crecido hasta 40 pines, mientras que conserva el mismo pinout para los primeros 26 pines como el modelo A y B.
- **Micro SD.** El antiguo zócalo de la tarjeta SD ajuste por fricción ha sido reemplazado con un push-push versión mucho más agradable micro SD.
- **Menor consumo de energía.** Mediante la sustitución de reguladores lineales con los de conmutación hemos reducido el consumo de energía entre 0.5W y 1W.
- **Mejor audio.** El circuito de audio incorpora una fuente de alimentación de bajo ruido dedicado.

- **Más pequeño, factor de forma más ordenada.** Hemos alineado el conector USB con el borde de la placa, se trasladó de vídeo compuesto al conector de 3,5 mm, y añadió cuatro orificios de montaje colocados en ángulo recto. Modelo A + es de aproximadamente 2 cm más corto que el modelo A.

Recomendamos el modelo A + para los proyectos y los proyectos integrados que requieren de muy baja potencia, y que no requieren de Ethernet o varios puertos USB. [28].

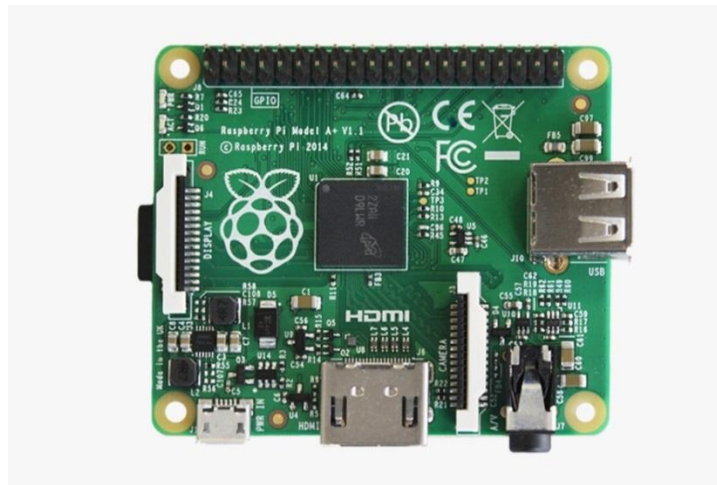


Figura 2.25 Raspberry Pi 1 Model A+

Fuente: <https://www.raspberrypi.org/raspberrypi-model-a-plus-on-sale/>

b. Raspberry Pi 1 Model B+

El modelo B + es la revisión final de la Raspberry Pi originales. Se sustituye el modelo B y fue reemplazado por el Raspberry Pi 2 Modelo B. En comparación con el modelo B tiene:

- **Más GPIO.** La cabecera GPIO ha crecido hasta 40 pines, mientras que conserva el mismo pinout para los primeros 26 pines como el modelo A y B.
- **Más USB.** Ahora tenemos 4 puertos USB 2.0, frente a 2 en el Modelo B, y una mejor conexión en caliente y el comportamiento de sobre corriente.
- **Micro SD.** El antiguo zócalo de la tarjeta SD ajuste por fricción ha sido reemplazado con un push-push versión mucho más agradable micro SD.

- **Menor consumo de energía.** Mediante la sustitución de reguladores lineales con los de conmutación hemos reducido el consumo de energía entre 0.5W y 1W.
- **Mejor audio.** El circuito de audio incorpora una fuente de alimentación de bajo ruido dedicado.
- **Más ordenado factor de forma.** Hemos alineados los conectores USB con el borde de la placa, se trasladó de vídeo compuesto al conector de 3,5 mm, y añadió cuatro orificios de montaje colocados en ángulo recto.

El modelo B + es perfectamente adecuado para su uso en las escuelas: ofrece una mayor flexibilidad para los estudiantes que el modelo más delgado A o A +, que son más útiles para los proyectos y los proyectos integrados que requieren de muy baja potencia, y tiene más puertos USB que el modelo B. [29].

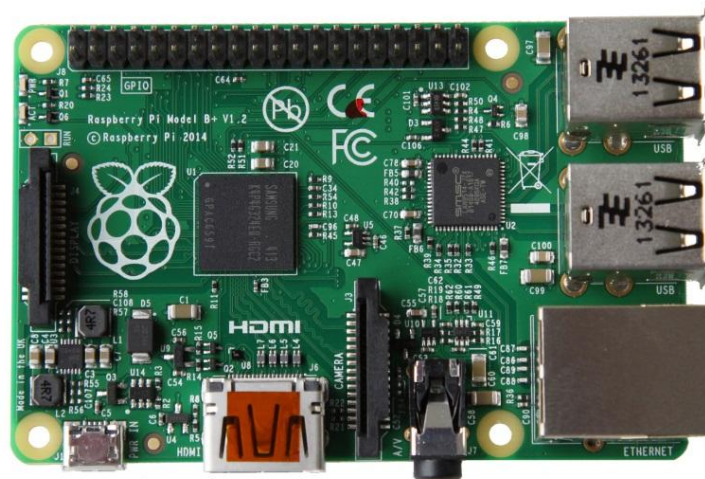


Figura 2.26 Raspberry Pi 2 Model B +
Fuente: <https://www.raspberrypi.org/products/model-b-plus/>

c. Raspberry Pi 2 Modelo B

El Raspberry Pi 2 Modelo B es la segunda generación de Raspberry Pi. Sustituyó el original Raspberry Pi 1. En comparación con el Raspberry Pi 1 tiene:

- Un 900MHz CPU de cuatro núcleos ARM Cortex-A7.
- 1 GB de RAM.

Al igual que el (Pi 1) Modelo B +, también tiene:

- 4 puertos USB
- 40 pines GPIO
- Puerto HDMI Full
- Puerto Ethernet
- Conector de audio de 3,5 mm combinado y vídeo compuesto
- Interfaz de la cámara (CSI)
- Interfaz de pantalla (DSI)
- Ranura para tarjeta Micro SD
- Núcleo de gráficos VideoCore IV 3D

Debido a que tiene un procesador ARMv7, puede funcionar la gama de distribuciones ARM GNU / Linux, incluyendo Snappy Ubuntu Core, así como Microsoft Windows 10

El Raspberry Pi 2 tiene un factor de forma idéntica a la anterior (Pi 1) Modelo B + y tiene compatibilidad completa con Raspberry Pi 1.

Recomendamos el Raspberry Pi 2 Modelo B para su uso en las escuelas: ofrece una mayor flexibilidad para los estudiantes que la más ligera (Pi 1) Modelo A + , que es más útil para los proyectos y los proyectos integrados que requieren de muy baja potencia. [30]

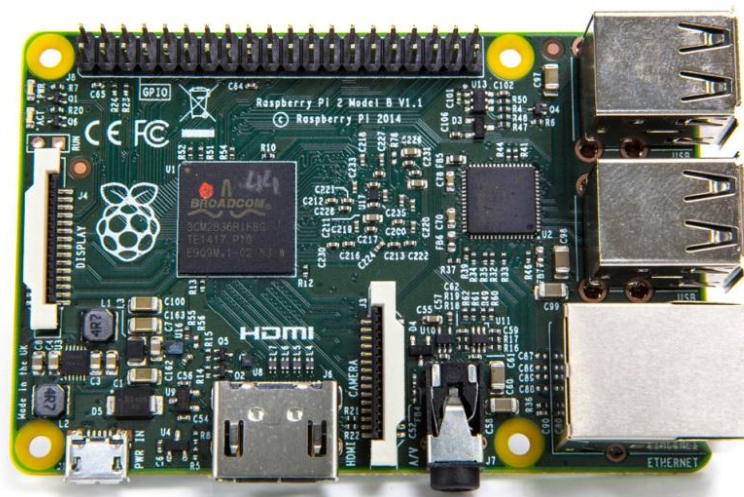


Figura 2.27 Raspberry Pi 2 Modelo B

Fuente: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

2.2.12 Plataforma Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP). [31]

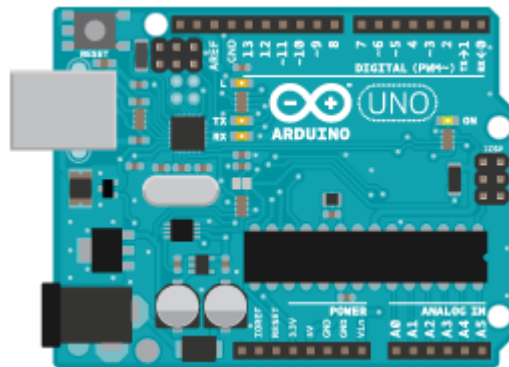


Figura 2.28 Aspecto de la placa Arduino.
Fuente: <https://www.arduino.cc/>

Tipos de placas Arduino

a. Arduino UNO Rev. 3

Es una placa electrónica basada en el ATmega328 cuenta con 14 entradas/salidas digitales pines (de los cuales pueden ser utilizados como salidas PWM), 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera de ICSP, y un botón de reinicio.

Contiene todo lo necesario para apoyar a la micro, solo tiene que conectarlo a un ordenador con un cable USB o el poder con un adaptador AC-DC o la batería para empezar. [32].



Figura 2.29 Placa Arduino UNO

Fuente: <http://www.electronicaembajadores.com/Admin/Content/eyontzqw.pdf>

Especificaciones.

- Microcontrolador: ATmega328
- Voltaje de operación: 5V
- Voltaje suplementario: 7-12V
- Maximum Supply voltaje: 20V no recomendado
- Digital I/O Pins: 14 (6 para PWM)
- Entradas analógicas:6
- DC Current per I/O PIN 40mA
- DC current for 3.3V pin: 50mA
- Flash Memory:32 KB
- SRAM: 2KB
- EEPROM: 1KB
- Clock speed: 16MHz

b. Arduino Leonardo

Muy similar al Arduino UNO, se trata de una evolución del mismo con mejores prestaciones y un precio similar.

Una de las ventajas de este nuevo microcontrolador es que dispone de USB nativo por hardware y por lo tanto no necesita de ninguna conversión Serie-USB. [32].



Figura 2.30 Placa Arduino Leonardo

Fuente: <http://www.electronicaembajadores.com/Admin/Content/eyontzqw.pdf>

Especificaciones

- Microcontrolador: ATmega32u4.
- Tensión de funcionamiento: 5V.
- Alimentación recomendada: 7-12V.
- Pines I/O Digitales: 20.
- Canales PWM: 7.
- Entradas analógicas: 12.
- Corriente Máxima de los pines I/O: 40 mA.
- Corriente máxima de los pines 3.3V: 50 mA.
- Memoria Flash: 32KB.
- SRAM: 2.5 KB.
- EEPROM interna: 1 KB.
- Velocidad: 16 MHz.

c. Arduino Mega

El Arduino MEGA es probablemente la placa con mayores prestaciones de la familia Arduino. Cuenta con 54 pines digitales, que funcionan como Entrada/Salida, además de sus 16 entradas analógicas. Es la placa más grande y potente de Arduino, es totalmente compatible con las shields Arduino UNO, y cuenta con una memoria que duplica su capacidad en comparación con el resto de las placas, en la figura se muestra la placa Arduino Mega. [32]



Figura 2.31 Placa Arduino Mega 2560

Fuente: <http://www.electronicaembajadores.com/Admin/Content/eyontzqw.pdf>

Especificaciones

- Microcontrolador: ATmega 2560
- Tensión de alimentación: 5V
- Tensión de entrada recomendada: 7.12V
- Pines I/O Digitales: 54
- Canales PWM: 14
- Entradas analógicas: 16
- Corriente Máxima de los pines I/O: 40 mA
- Corriente máxima de los pines 3.3V: 50 mA
- Memoria Flash: 256KB
- SRAM: 8 KB
- EEPROM Interna: 4 KB
- Velocidad: 16 MHz

d. Arduino Ethernet

Es la combinación en una sola placa de un arduino UNO y un ETHERNET Shield para los proyectos que necesitan de mejor espacio físico. Dispone también de un zócalo para tarjetas de memoria MicroSD que pueden ser utilizados para leer y escribir datos, en a figura se muestra la placa arduino Ethernet. [32]



Figura 2.32 Placa Arduino Ethernet

Fuente: <http://www.electronicaembajadores.com/Admin/Content/eyontzqw.pdf>

Especificaciones

- Microcontrolador: ATmega 328
- Tensión de alimentación: 5V
- Tensión de entrada recomendada: 7.12V
- Pines I/O Digitales: 14
- Canales PWM: 4
- Entradas analógicas: 6
- Corriente Máxima de los pines I/O: 40 mA
- Corriente máxima de los pines 3.3V: 50 mA
- Memoria Flash: 32KB
- SRAM: 2 KB
- EEPROM Interna: 1 KB
- Velocidad: 16 MHz
- W5100 TCP/IP Embedded Ethernet Controller
- Power Over Ethernet ready Magnetic Jack
- Micro SD card.

e. Arduino Nabduino 1.2.

La placa está diseñada por la empresa Navto y tiene el aspecto de Arduino. El corazón de la misma es el PIC8F6160 de microchip y lleva incorporada la conexión Ethernet. La idea es que pueda emplearse los mismo shields de Arduino, pero el software entre ambos no son compatible.

Es por tanto unos accesos remoto web para dispositivos pequeños marco embebido. [32]



Figura 2.33 Placa Nabduino 1.2

Fuente: <http://www.electronicaembajadores.com/Admin/Content/eyontzqw.pdf>

Así que en lugar de las herramientas de Arduino, las herramientas que se utilizaran con la placa Nabduino son:

- MPLAB.X
- Herramienta flash Ethernet
- Nabto plugins del navegador
- ICD3 o depurador Pickit (solo firmware opcional, puede ser fraseado a travez del puerto Ethernet de un PC)
- La plataforma de hardware, el código fuente y el firmware están abiertos y el código fuente puede ser descargado y adjuntado. [32]

Entorno de desarrollo Arduino

El entorno de desarrollo en Arduino (IDE) es el encargado de la gestión de la conexión entre el PC y el hardware de Arduino con el fin de establecer una comunicación entre ellos por medio de la carga de programas. Como podemos ver en la figura 3.16, el IDE de Arduino se compone de: [33]

Un editor de texto.- donde escribir el código del programa.

Un área de mensajes.- a través del cual el usuario tendrá constancia en todo momento de los procesos que se encuentren en ejecución, errores en código, problemas de comunicación, etc.

Una consola de texto.- mediante la que podremos comunicarnos con el hardware Arduino y viceversa.

Una barra de herramientas.- donde podremos acceder a una serie de menús y a los botones con acceso directo a las principales funcionalidades de Arduino. [33]

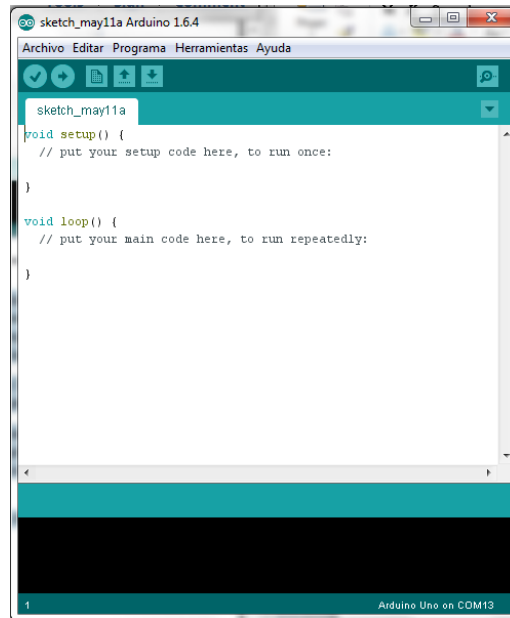


Figura 2.34 Entorno de desarrollo Arduino

Fuente: El Investigador

A través de la IDE de Arduino, podemos escribir el código del programa software y crear lo que se conoce por "sketch" (programa). ¿Por qué lo llamamos sketch y no programa? Pues porque el IDE de Arduino viene de Processing, y en este lenguaje de programación enfocado al mundo gráfico, cada código es considerado un boceto, en inglés "sketch".

El sketch permite la comunicación con la placa Arduino. Estos programas son escritos en el editor de texto, el cual admite las posibilidades de cortar, pegar, buscar y reemplazar texto.

En el área de mensajes se muestra, tanto la información mientras se cargan los programas, como los posibles errores que tengamos a la hora de compilar, ya sea por problemas en el código del sketch, por fallo en la detección de nuestro Arduino en el puerto USB, o por cualquier otro problema que sea detectado.

La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones.

Desde la barra de herramientas tenemos acceso directo a las principales funcionalidades que ofrece el IDE de Arduino, como por ejemplo: verificar el proceso de carga, crear un nuevo sketch, abrir un sketch ya existente, guardar los programas, abrir el Monitor Serial, etc.

A continuación pasamos a describir la utilidad de cada uno de los iconos que aparecen en la pantalla principal del entorno de desarrollo de Arduino:

“Verificar”.- Esta funcionalidad se encarga de verificar el código del sketch en busca de posibles errores. A través del área de mensajes se le notificará al usuario el resultado de dicha verificación. En el caso de que se detecten errores en el código, éstos se detallarán junto con el número de línea en la que han sido detectados. Sólo cuando la comprobación resulta libre de errores podremos proceder a la carga del código en nuestra placa Arduino. [33]

- **“Cargar”**.- Permite compilar el código del sketch y lo carga en Arduino. Cuando la carga a terminado se informa al usuario a través del área de mensajes, y podremos proceder a la apertura del monitor serial.
- **“Nuevo”**.- Para la creación de un nuevo sketch. Abre una nueva hoja de texto donde escribiremos el código correspondiente al sketch.
- **“Abrir”**.- Permite abrir un sketch ya existente que ha sido previamente guardado.

También puedes abrir cualquiera de los sketches que trae instalados por defecto el IDE de Arduino.

- **“Guardar”**.- Esta funcionalidad nos permite almacenar el sketch que estemos desarrollando en ese momento. Te permite elegir la ruta en la que quieres guardarlo, y te crea automáticamente una carpeta con el mismo nombre que le des al sketch, guardando éste dentro de la misma.
- **“Monitor Serial”**.- Al pinchar sobre este icono, el entorno de desarrollo de Arduino abre una nueva ventana a través de la cual podemos ver la

comunicación establecida por el puerto serie entre la placa Arduino y el PC durante la ejecución del programa. Contiene una barra de escritura mediante la que podemos comunicarnos con Arduino a través de su puerto serie, por ejemplo, para seleccionar distintas opciones que contemple un posible menú creado por el usuario dentro de un código, o para enviar directamente comandos AT a una shield

GPRS/GSM que tengamos montada sobre el Arduino. También contempla la opción de seleccionar el envío de algunos caracteres junto con el texto que introduzcamos en la barra de entrada del mismo, como el carácter de nueva línea, retorno de carro, o los dos. En la figura 3.17 podemos ver la pantalla correspondiente al Monitor Serial y la pestaña desplegable en la que podemos seleccionar las distintas opciones referentes a los caracteres de fin de línea. [33]

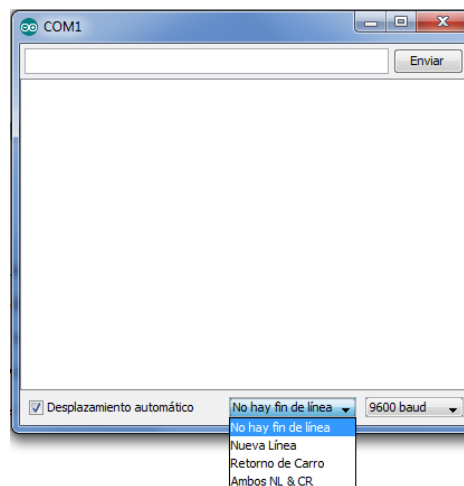


Figura 2.35 Selección de caracteres de fin de línea en la ventana "Monitor Serial"
Fuente: El Investigador

Shields Arduino

Si queremos ampliar las funcionalidades de nuestra plataforma Arduino, siempre podemos recurrir a una gran variedad de shields compatibles prácticamente con cualquiera de sus modelos. De este modo, podemos dotar al dispositivo de funciones adicionales dedicadas específicamente a ofrecer algún tipo de servicio concreto.

Un shield es un módulo de expansión en forma de placa impresa que se puede conectar a la parte superior de la placa Arduino para ampliar sus capacidades, permitiendo

además ser apiladas unas encima de otras manteniendo un diseño modular, tal como podemos ver en la Figura 3.36. [33]

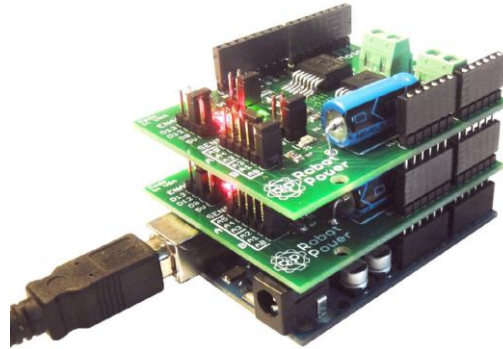


Figura 2.36 Ejemplo de estructura modular con varias shields para Arduino

Fuente: <http://www.robotshop.com/en/arduino-compatible-mega-motor-shield-1a-5-28v.html>

A continuación se muestran los tipos de Shields

a. Módulo GPRS+GPS Quadband para Arduino/Raspberry Pi (Sim908)

Este es un modelo de la shield GPRS para Arduino. Gracias a que cuenta con un módulo SIM908 integrado en la propia placa, ofrece la posibilidad de utilizar la tecnología GPS para posicionamiento en tiempo real, resultando muy útil para aquellas aplicaciones en las que necesitemos conocer la ubicación de nuestro dispositivo. En la figura 3.22 se adjunta una imagen de dicha shield. [33].



Figura 2.37 Módulo GPRS+GPS Quadband para Arduino y Raspberry PI (SIM 908)

Fuente: Sistema de control de temperatura a través de Arduino y la tecnología GPRS/GSM

b. Módulo 3G/GPRS+GPS para Arduino/Raspberry Pi

Este es el modelo más completo entre todas las shields GPRS disponibles. A parte del sistema GPRS, gracias a su módulo SIM5218, integra también servicios 3G y tecnología GPS, pero es cierto que admite más funcionalidades en comparación con el resto de shields que hemos visto hasta ahora, incluso permite la conexión de una cámara para la toma de imágenes. En la figura 3.24 podemos ver el aspecto que presenta esta shield. [33]



Figura 3.38 Módulo 3G/GPRS+GPS para Arduino/Raspberry PI

Fuente: Sistema de control de temperatura a través de Arduino y la tecnología GPRS/GSM

c. Shield SIM 900 GSM/GPRS para Arduino

Esta shield puede convertir nuestra placa Arduino en un plataforma capaz de ofrecer conectividad GPRS/GSM. Integra un módulo SIM900 que nos permite establecer llamadas con otros dispositivos móviles, enviar SMS, incluso la comunicación de datos a través de los protocolos TCP, UDP, HTTP o FTP. En la figura 3.26 se adjunta una imagen de dicho módulo: [33].



Figura 2.39 Shield Sim 900

Fuente: <http://www.geekfactory.mx/tienda/shields-arduino/shield-gsm-gprs-sim900-para-arduino/>

Junto con el modulo, se debe adquirir una antena y una fuente de alimentación externa, ya que los 5V de Arduino no son suficientes para alimentar tanto al módulo como a los componentes que conectamos a nuestra plataforma, las características del SHIELD SIM900 Se muestran en la anexo. [33].

d. Phone Shield SIM908 GSM/GPRS/GPS

Módulo SIM908 es un módulo industrial GSM / GPRS / GPS de alto rendimiento. Las interfaces de esta placa de desarrollo son ricas funciones, perfecto, sobre todo es adecuado para la necesidad de expresar / SMS / GPRS / GPS de servicio de datos de navegación de todo tipo de campos. Placa de desarrollo SIM908 no sólo es hermoso, pero la función, la interfaz rica también completó. El tamaño del módulo (no incluyendo la antena parte) es 77.978 mm * 73.787 mm. El módulo ha agujeros de montaje que son muy pequeñas, y propicio para instalar y se puede aplicar fácilmente a todo tipo de diseño del producto. [34].



Figura 2.40 Shield GSM/GPRS/GPS Sim 908

Fuente: <http://www.wvshare.com/product/Phone-Shield-SIM908.htm>

2.2.13 Comunicaciones inalámbricas

La comunicación inalámbrica o sin cables es la comunicación que se da entre un emisor y receptor pero que no se da de manera física entre ambos, sino a través de un medio de propagación, con ondas que se encuentran en el espacio y por medio de aparatos móviles que no utilizan cables. . En este sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales encontramos: antenas, computadoras portátiles, PDA, teléfonos móviles, etc. [35]

Estándar GSM

Las siglas GSM vienen de las palabras anglosajonas Global System for Mobile communications. Como su propio nombre indica, pues, el GSM no es más que un estándar de comunicación para la telefonía móvil, implementado mediante la combinación de satélites y antenas terrestres. A los móviles que usan la tecnología GSM también se les conoce por móviles 2g o de segunda generación desde master magazine

En la figura 2.2 podemos ver la arquitectura de red correspondiente al sistema GSM. Está compuesta por múltiples estaciones base (BTS), que a su vez, se conectan a un controlador de estaciones base (BSC), encargado de la administración de la red. A éste sistema compuesto por el BSC y sus correspondientes estaciones base conectadas al mismo, se le conoce como BSS (Subsistema de estaciones base). [33]

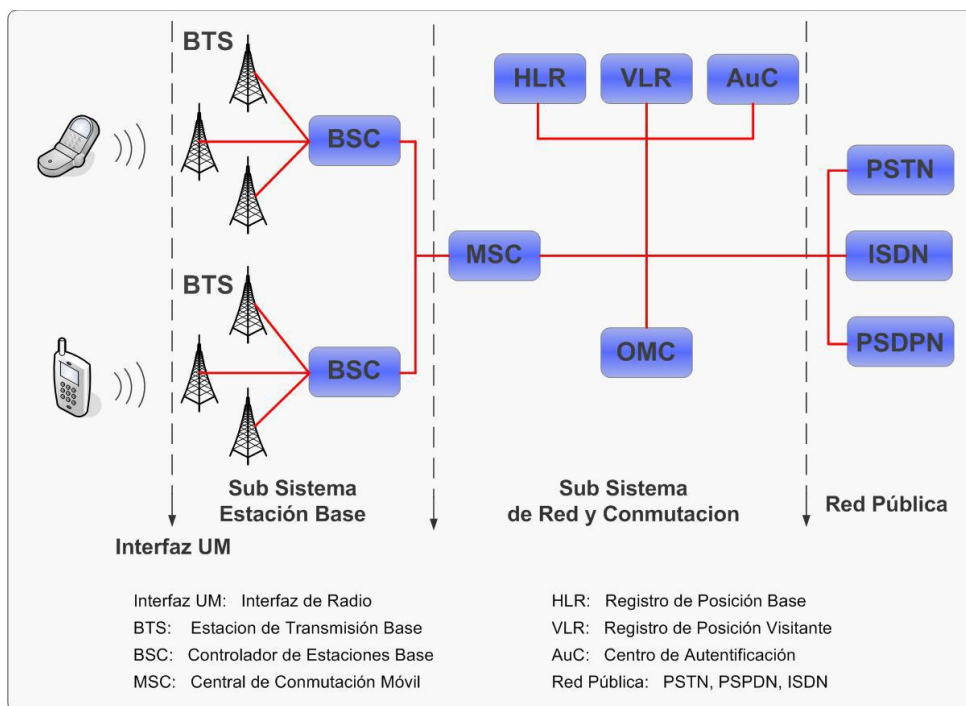


Figura 2.41 Arquitectura de red del sistema GSM

Fuente: Sistema de control de temperatura a través de Arduino y la tecnología GPRS/GSM

Estándar GPRS

El estándar GPRS o Servicio General de Paquetes vía Radio (en inglés, General Packet Radio Service) es una evolución del sistema GSM. Es también conocido como GSM++, pero dado que se trata de un estándar de telefonía móvil intermedio entre la segunda generación (2G) y la tercera (3G), a menudo recibe la nomenclatura de 2.5G.

GPRS extiende la arquitectura del estándar GSM para permitir la transferencia de datos mediante conmutación de paquetes con velocidades de transferencia que rondan los 114 Kbps.

Al contrario de lo que ocurre en conmutación de circuitos, en el estándar GPRS, gracias a su modo de transferencia de paquetes, las transmisiones de datos sólo utilizan la red cuando es necesario, permitiendo la tarificación por volumen de información transmitida en lugar de por tiempo de conexión, por lo tanto, el usuario puede

permanecer conectado sin costo adicional, ya que sólo utilizará la red cuando envíe o reciba un paquete de datos.

Para el acceso a la red de datos, el estándar GPRS utiliza el protocolo IP, mientras que para el transporte de voz, emplea la arquitectura de la red GSM.

A parte de actualizar algunos servicios con los que ya contaba GSM, la tecnología GPRS admite otra serie de características que no estaban disponibles en 2G: [33].

- Servicios de mensajes cortos (SMS)
- Servicios de mensajes multimedia (MMS)
- Servicio punto a punto (PTP); para la conexión cliente-servidor en una red IP
- Servicio punto a multipunto (PTMP); para el envío de multidifusión.

En la figura 2.42 se muestra la estructura funcional del sistema GPRS, basada en la adición de nuevos nodos sobre la infraestructura correspondiente a GSM. A dichos nodos se les conoce como GSN (nodos de soporte GPRS):

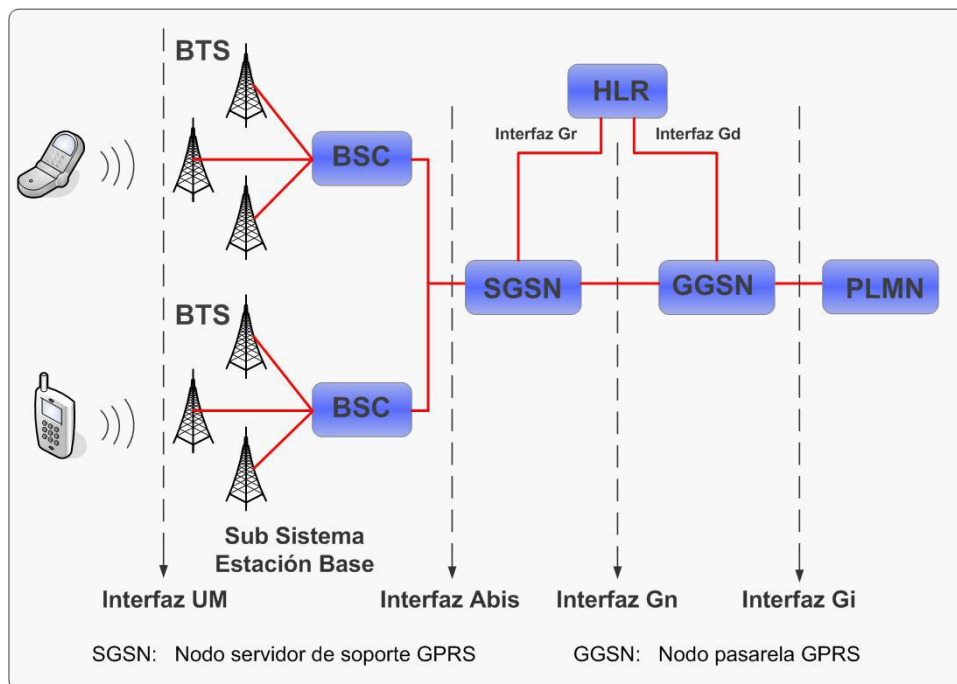


Figura 2.42 Arquitectura de red del sistema GPRS

Fuente: Sistema de control de temperatura a través de Arduino y la tecnología GPRS/GSM

2.2.14 Comandos AT

Los comandos AT, también conocidos como comandos Hayes (en honor a su desarrollador Dennis Hayes), son una serie de instrucciones que conforman un interfaz de comunicación entre usuario y modem. Su abreviatura AT por la que son mundialmente conocidos estos comandos proviene de la palabra ‘attention’.

Aunque la finalidad principal de los comandos AT fue la comunicación con módems, la telefonía móvil GSM/GPRS también adoptó este lenguaje como estándar de comunicación.

En la actualidad, todos los terminales móviles GSM poseen una serie específica de comandos AT que nos permiten configurarlos por medio de estas instrucciones e indicarles una serie de acciones que queremos que ejecuten, tales como marcar un número de teléfono, enviar o leer un SMS, consultar el estado de conexión a la red, leer o escribir en la agenda de contactos, etc.

Gracias a que la transmisión de comandos AT no depende del canal de comunicación a través del cual estos sean enviados (cable, infrarrojos, Bluetooth, etc.), podremos utilizar. [33]

2.2.15 Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java. [36]

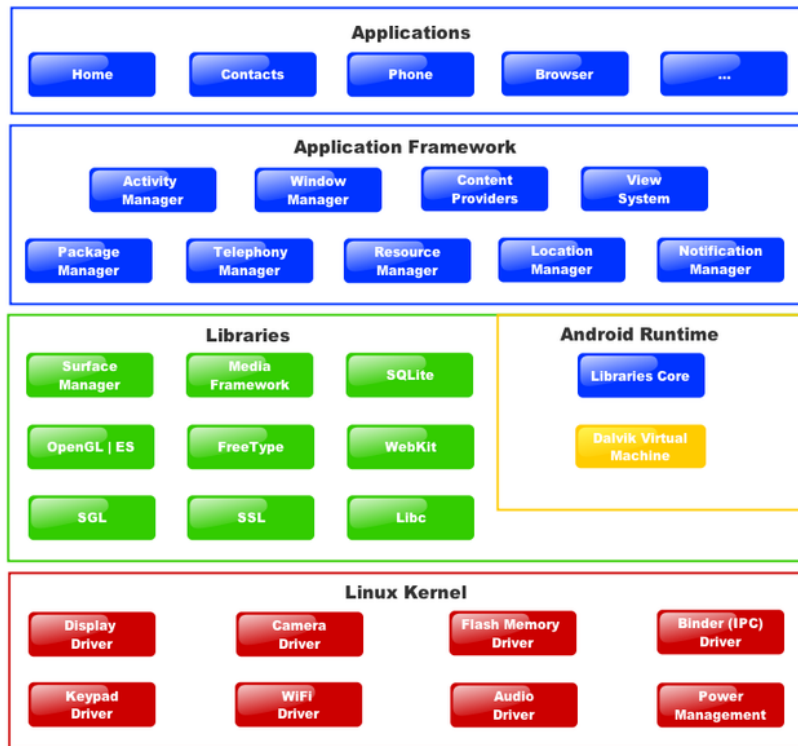


Figura 2.43 Diagrama de bloques del funcionamiento de Android
Fuente: <http://www.xatakandroid.com/sistema-operativo/que-es-android>

2.2.16 APP-Inventor

App inventor es un framework creado inicialmente por el MIT (Instituto tecnológico de Massachusetts) y fue cogido por google, para que cualquier persona con interés pueda crearse su propia aplicación móvil, ya sea para su empresa, para su casa o por otros intereses. [37]

Para crear una aplicación con app inventor hay que realizar tres pasos:

El diseño de la aplicación, en la que se seleccionan los componentes para su aplicación.

- El editor de bloques, donde irás escogiendo los bloques que te sean necesarios según la aplicación que tengas pensada de hacer.
- Tu aplicación aparecerá paso a paso de en la pantalla del teléfono a medida que añada piezas a la misma, para que puedas probar tu trabajo. Cuando haya terminado, puedes empaquetar la aplicación y producir una aplicación independiente para instalar. [37].

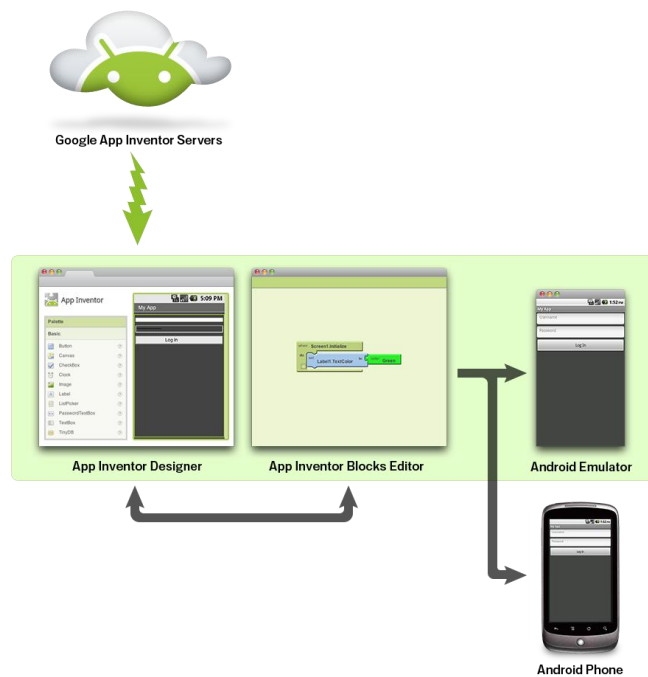


Figura 2.44 Pasos para crear una aplicación Android en App-Inventor
Fuente:<http://s445852199.mialojamiento.es/peponees/appinventorspanish/learn/whatis/index.html>

Si no tienes un teléfono Android, puedes construir tus aplicaciones utilizando el emulador de Android, el software que se ejecuta en tu computadora y se comporta como el teléfono.

El entorno de desarrollo de App Inventor es compatible con Mac OS X, GNU / Linux y sistemas operativos de Windows, y varios modelos de teléfonos Android populares. Las aplicaciones creadas con App Inventor se pueden instalar en cualquier teléfono Android.

Antes de poder utilizar App Inventor, es necesario configurar el ordenador e instalar los archivos de instalación de App Inventor en el equipo. [37]

2.3 Propuesta de solución

Con la construcción de un ordenador electrónico vehicular con sistema de seguridad y GPS utilizando Raspberry Pi y plataforma de hardware libre, se potencio la seguridad de los vehículos ante la delincuencia con un sistema avanzado en tecnología inalámbrica.

CAPÍTULO III

METODOLOGÍA

3.1 Modalidad de la investigación

Investigación Bibliográfica

Para el presente proyecto se utilizó investigación Bibliográfica, debido a que la definición científica del tema de investigación está enfocado a la tecnología, además se realizó mediante la consulta en libros, pertenecientes a la rama de la electrónica y las comunicaciones soluciones al proyecto.

Investigación Aplicada

El presente proyecto permitió realizar una investigación aplicada debido a que se pudo emplear todos los conocimientos adquiridos para desarrollar la construcción de un ordenador electrónico vehicular con sistema de seguridad y GPS utilizando Raspberry Pi y hardware libre.

Investigación Experimental

El presente proyecto permitió realizar una investigación Experimental debido a que se manipulo y programó objetos electrónicos con el fin de establecer una relación fundamental entre la inexistencia de un ordenador y un sistema de seguridad vehicular.

3.2 Población y muestra

Al realizarse el presente trabajo de investigación no es necesario establecer cuál va a ser la población y muestra, debido a que se construirá un producto nuevo que antes de

ser lanzado al mercado debe comprobarse la utilidad del mismo y también deberá ser sometido a pruebas previas a su montaje en vehículos.

3.3 Recolección de información

Para alcanzar los objetivos de la investigación se aplicó técnicas de levantamiento de información de datos en los módulos de comunicación electrónicos, utilizando documentos o guías para el almacenamiento de información sobre pruebas de programación realizadas.

3.4 Procesamiento y análisis de datos

Una vez obtenida la información necesaria, se procede a una revisión crítica y se realizó la corrección necesaria. Después la información será parte de un proceso estadístico, mediante la tabulación de datos, de forma ordenada y sistemática. De acuerdo a cuadros estadísticos se presentó el análisis de los resultados, enfatizando las tendencias o relaciones fundamentales de acuerdo al objetivo y a las variables de la hipótesis encontrada

3.5 Desarrollo del proyecto

- Recolección de información del principio y funcionamiento de sistemas de seguridad anti delincuenciales actualmente existentes en el Ecuador para vehículos.
- Estudio sobre el sistema electrónico de encendido de un vehículo.
- Recolección de información sobre el sistema GPS para vehículos.
- Diseño del diagrama de flujo del sistema de mando.
- Programación del ordenador Raspberry Pi con las funciones necesarias.
- Diseño de la interface entre el sistema de Seguridad y el ordenador Electrónico.
- Diseño e implementación de la interface entre el GPS y el Ordenador Electrónico.
- Implementación del sistema controlador en el vehículo

- Evaluación y comprobación del funcionamiento del sistema.
- Elaboración y presentación del informe final.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1 Análisis de los Sistemas de seguridad para vehículos actuales.

4.1.1 Sistema de seguridad por contratación pública

Son sistemas encargados de brindar seguridad al vehículo para cualquier intento de manipulación por parte de persona externas no autorizadas para su uso.

Utiliza los siguientes dispositivos que se detallan a continuación:

- Modulo principal de control
- Control inalámbrico de mando
- Sensores para puertas.
- Motores actuadores para bloqueo de puertas
- Sensor de vibración
- Relés para control
- Sirena auditiva

En la figura 4.1. Se muestra la distribución de cada uno de los dispositivos hacia el módulo de mando.

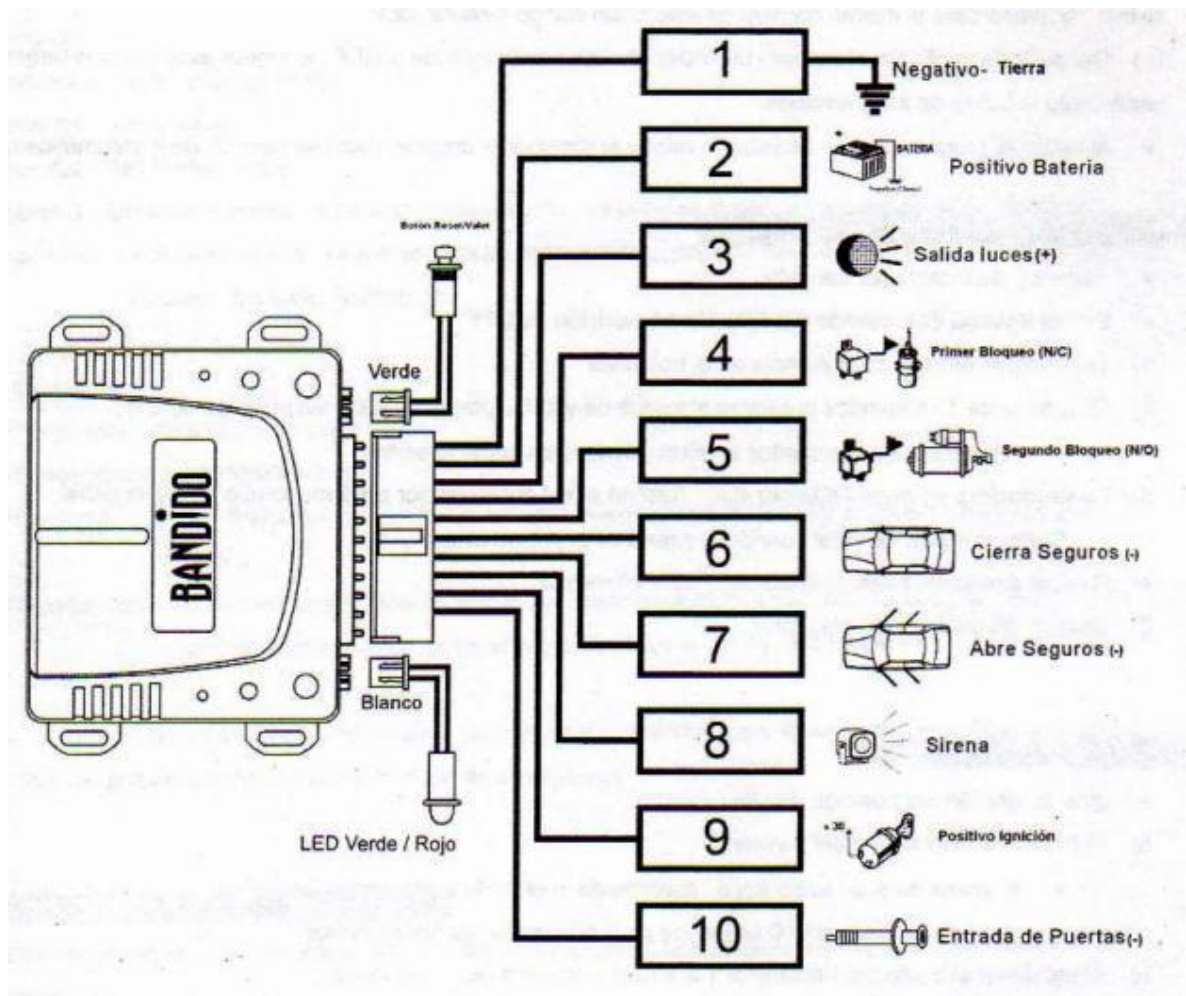


Figura 4.1 Diagrama de conexión de un sistema de seguridad vehicular
Fuente: <http://diagramasde.com/blog/133112/alarmas-1332270822-154952446-6-curso-de-isntalacion-de-alarma-para-auto-mexico-jpg.html>

Funcionamiento:

El sistema es activado y desactivado a través de un control de mando, cuando es sistema está armado el modulo administra los sensores instalados en cada sitio estratégico para que, cuando sea activado envíe una señal hacia el controlador el cual activará la alarma ubicada dentro del automotor.

Dicho sistema solo podrá ser desactivado con el control de mando.

4.1.2 Sistema de seguridad por contratación privada.

Son aquellos sistema los cuales están sujetos a suscripciones ya sea mensual o anualmente exclusivos para la línea Chevrolet, los cuales brindan soporte y seguridad

desde las centrales realizando un seguimiento constante de la ubicación del vehículo a través de las comunicaciones inalámbricas sean estas satelitales o utilizando la red GSM/GPRS,

A continuación se analiza las empresas que prestan su servicio en el Ecuador.

CHEVYSTAR

Brinda los siguientes beneficios.

- Localización y recuperación de su vehículo en caso de robo.
- Apertura remota de las puertas en caso de olvido en su interior.
- Asistencia médica y mecánica.
- Comunicación celular manos libres.
- Comunicación con el Centro de Atención al Cliente las 24 horas del día los 365 días del año desde el vehículo.
- Monitoreo de su vehículo cuando esté en carretera.
- Monitoreo en la ciudad
- Bluetooth
- Perfil de manejo

En la figura 4.2 se muestra el diagrama de conexión de un sistema de alarma Chevystar

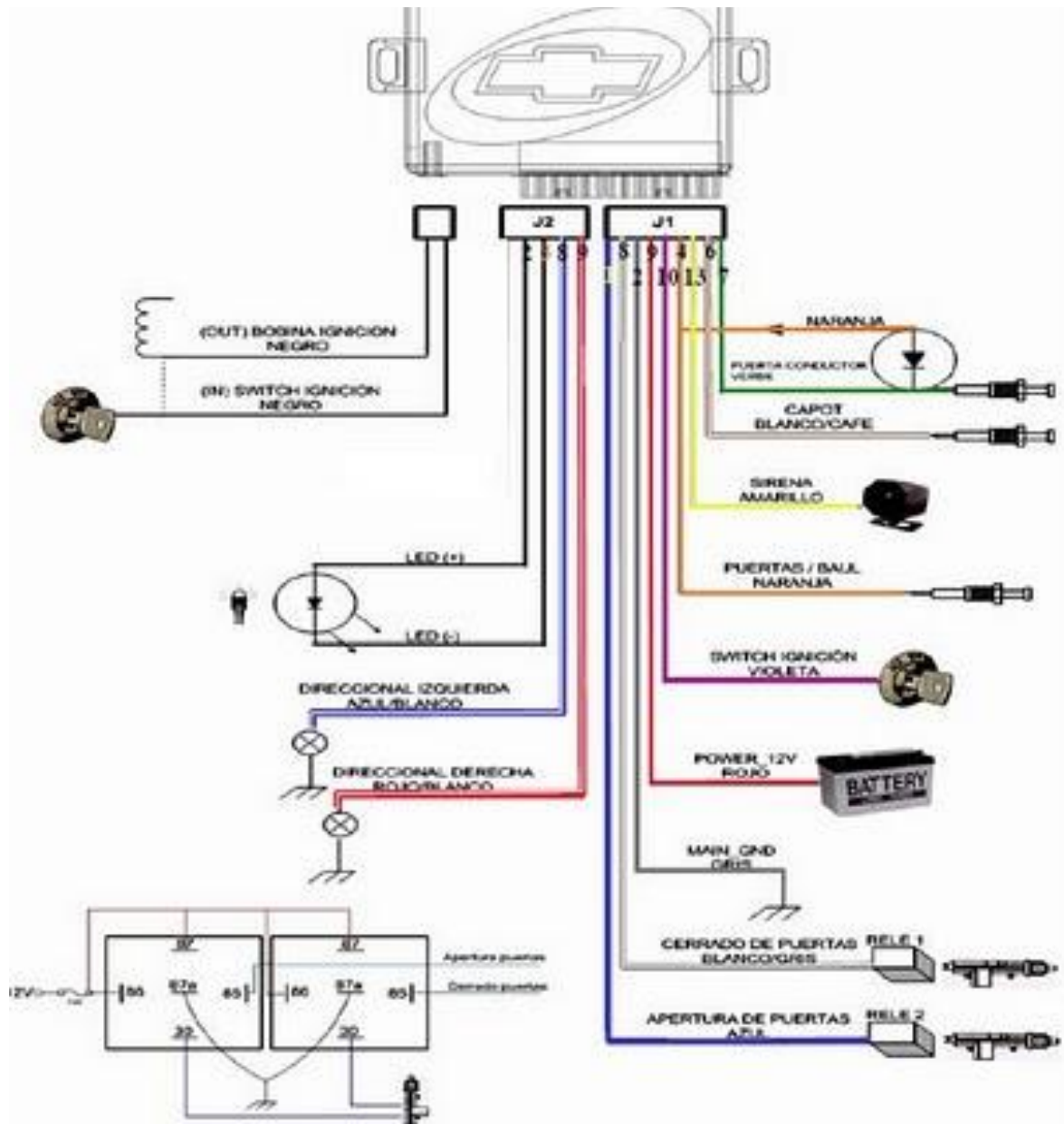


Figura.4.2 Diagrama de conexión de alarma ChevyStar

Fuente: http://mecanica.yoreparo.com/electronica_automotriz/alarma-chevy-de-spark-bloqueada-t680725.html

Funcionamiento

En la figura 4.3 se muestra el funcionamiento del servicio ChevyStar para el usuario



Figura 4.3 Descripción de las opciones de CheyStar

Fuente:https://www.globalbuypower.com/content_data/LAAM/EC/es/GBPEC/001/BRANDSITE/site/cheystar/como-funciona-cheystar.html

TRACKER GPS

En la figura 4.4 se muestra el diagrama de comunicación del sistema de seguridad Tracker GPS.



Figura 4.4 Diagrama de comunicación TRACKER GPS

Fuente:http://es.made-in-china.com/co_cncoban/product_GPS-Tracking-System-for-Car-Taxi-Truck-Motor-Asset-Transportation-with-GPRS-Web-Based-Fleet-Management-Software-102-_essyyiong.html

Características:

- Orientación del soporte sin el servidor orientado: Puede descubrir longitud y latitud del Verdadero-tiempo, tiempo y velocidad. Usando software público de la correspondencia y entrando en longitud y latitud manualmente, usted puede descubrir la información en tiempo real de la posición.
- Puede utilizar la plataforma: Al usar la plataforma, puede seguir blancos en tiempo real en correspondencia electrónica, o investigar la posición de blanco por el teléfono móvil.
- Seguimiento de la función por medir el tiempo y frecuencia
- Función agregada y suprimida del número autorizado
- Función ocultada soporte de la alarma y del monitor
- Geo-cerca: Fijar una geo-cerca para que la unidad restrinja sus movimientos dentro de un distrito. La unidad enviará el mensaje a los números autorizados cuando practica una abertura el distrito.
- Alarma del movimiento: Usted puede fijar barreras cuando paran al perseguidor. Una vez que está movido, puede dar una alarma al número autorizado.
- Sobre alarma de la velocidad: Usted puede hacer la blanco examinada ejecutada según velocidad fija. Alarma cuando se excede la velocidad.
- Función baja de la alarma de la batería

HUNTER SOS

Es un sistema de monitoreo que reporta la ubicación del vehículo y viene con cámara que capta fotos cada minuto con el fin de saber que está ocurriendo en el interior del vehículo.

Existen dos servicios que brinda que son:

a. Hunter FULL

Sistema de monitoreo con doble tecnología (celular y Lojack), reporta la ubicación del vehículo por medio de la página web o celular y la ubicación del vehículo para rastreo en caso de robo.

b. Hunter Monitoreo Satelital

Sistema de monitoreo satelital, que garantiza que no habrá pérdida de señal en ningún momento, aunque se encuentre en alta mar o zonas de poca accesibilidad.

Hunter proporciona tecnología fija, móvil e inalámbrica; se ha mantenido líder en el mercado ecuatoriano con un servicio de altísima calidad en rastreo y ubicación de vehículos robados.

Monitoreo con ubicación por GPS/GPRS: Da la zona aproximada donde se encuentra el vehículo, es una aproximación, indica la zona, pero no el lugar preciso donde se encuentra. Depende de la red de la compañía celular y de la cobertura que esta pueda darle. Pero en caso de estar en una intersección no puede indicarle el lugar exacto de vivienda, donde se encuentra el activo de riesgo, sólo el aproximado.

- Uso exclusivo para monitorear y administrar flotas.
- Tecnología GPS / GPRS.
- Monitoreo las 24 horas por web o software especial.
- Genera reportes de velocidad, paradas, zonas restringidas, etc.
- Se recomienda su uso para administración de flotas de vehículos.

4.2 Análisis del Sistema

Después de haber investigado el sistema electrónico de encendido y analizado los sistemas de seguridad vehiculares se ha determinado la realización del proyecto enfocado al Sistema de Seguridad Antirrobo con hardware libre el cual no esté inscrito a ninguna suscripción y se disponga de un control y monitoreo del mismo desde cualquier sitio a través de una aplicación móvil personal.

El ordenador electrónico vehicular con sistema de seguridad y GPS tiene como finalidad brindar al propietario del vehículo mejor manejo del medio que compone el sistema de seguridad en el vehículo tanto interno y externo, utilizando las comunicaciones inalámbricas obteniendo control a cualquier distancia de la activación o desactivación del mismo y una parte fundamental de los sistemas de rastreo para la ubicación en tiempo real del vehículo también denominado GPS, estas funciones se las acopla en una aplicación móvil la cual permite al usuario disponer de un mejor manejo y

entendimiento del sistema. El sistema consta de tres partes fundamentales que son: la primera compuesta por sensores de puertas, sensor de encendido del vehiculo, actuadores que conforma el sistema de alarma, dos placas principales: una placa la cual es el controlador del sistema, y la segunda la cual realiza la funcion de usuario-maquina y maquina-usuario a traves de un entorno grafico de control de desbloqueo principal dentro del vehiculo atravez de una pantalla touch, por ultimo una aplicaci3n movil que realiza la funcion de control y monitoreo del sistema.

En la figura 4.5 se muestra el diagrama de bloques de los elementos que conforman el Ordenado Electronico Vehicular.

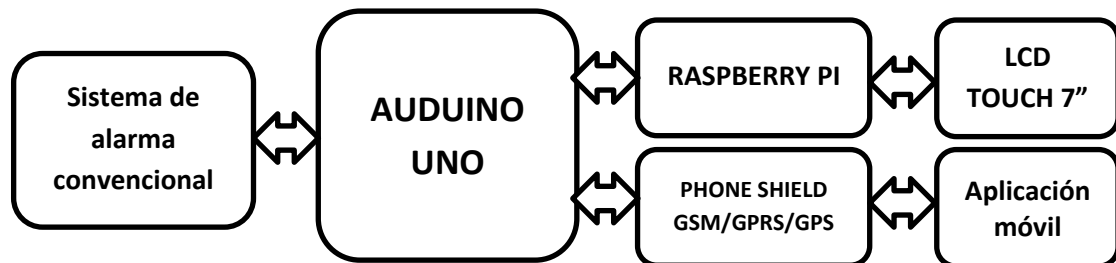


Figura.4.5 Componentes principales que conforman el Ordenador Vehicular
Fuente: El Investigador

El modo de operaci3n consta de dos partes fundamentales cuando el sistema entra en funcionamiento: La primera parte esta constituida de un control de monitoreo dentro del vehiculo el cual reconoce si las puertas del vehiculo se encuentran abiertas o cerradas informando al conductor este comportamiento en un entorno grafico, ademas de un control de seguridad el cual protege el movimiento del vehiculo atravez de una clave personal. La segunda parte esta enfocada en la proteccion antirobo y ubicaci3n del vehiculo, este control se lo realiza atravez de una aplicaci3n movil y procedimiento se los detallan acontinuacion.

Cuando el usuario activa el sistema el ordenador vehicular bloquea los seguros de las puertas y activa la funcion de los sensores.

Cuando el usuario desactiva el sistema el ordenador vehicular desbloquea los seguros de las puertas y desactiva la funcion de los sensores.

En el caso que el sistema se encuentre activado y alguna puerta es forzada para abrirla, el ordenador vehicular envia un mensaje de texto con el detalle del evento al numero o numeros de celular registrados.

Cuando el usuario solicita la ubicación del vehiculo, el ordenador vehicular envia las cordenadas de ubicación del mismo al numero registrado y la aplicación se encarga de mostrarla atravez de la app Google Maps.

Es evidente que dado los presentes requerimientos del sistema, el controlador debera contar de un dispositivo que permita la comunicación de mensaje de textos, lo cual puede realizar con el shield GSM para arduino.

Este comportamiento del sistema propuesto se lo muestra en la siguiente figura:

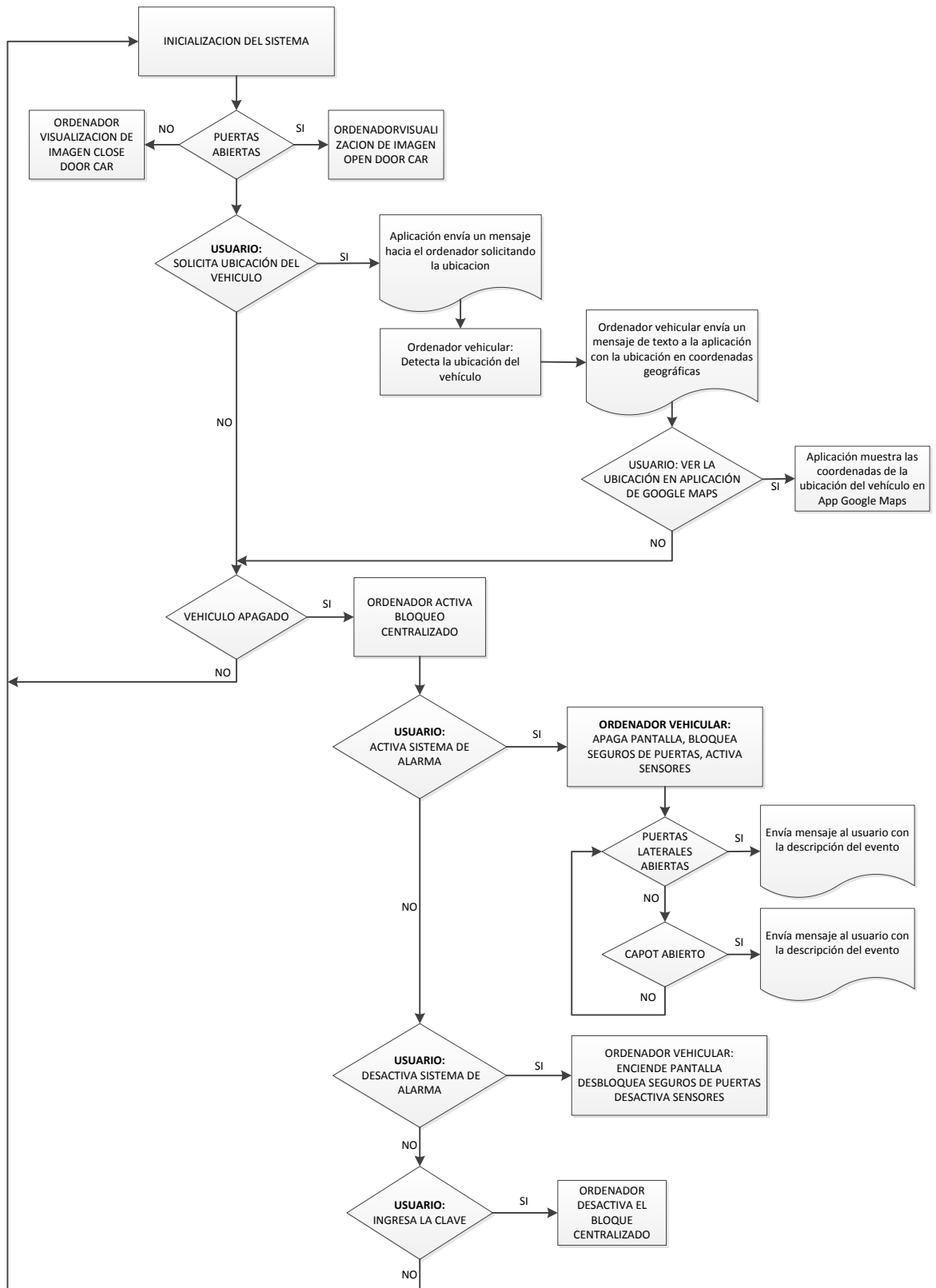


Figura 4.6 Diagrama de flujo del comportamiento del Sistema propuesto.
Fuente: El Investigador.

4.3 Comparación de las placas de la plataforma de Arduino

Dentro de la familia de Arduino existen diferentes series que se detallan a continuación:

Tabla 4.1. Tabla de las Características de las Series Arduino.

	Arduino Uno	Arduino Leonardo	Arduino MEGA 2560	Arduino Mega ADK rev 3
Microcontroller	ATmega 328	ATMega 32U4	Atmega 2560	Atmega 2560
Operating Voltage	5V-12V	5V-12V	5V-12V	5V-12V
Digital I/O Pins	14	20	54	54
Analog Inputs Pins	6	12	16	12
Puertos TX/RX	1	1	4	6
Canales PWM	0	7	14	14
Flash memory	32 KB	32 KB	256kB	256kB
SRAM	2 KB	2.5 KB	8 KB	8 KB
EEPROM	1 KB	1 KB	4 KB	4 KB
Clock speed	16 Mhz	16 Mhz	16 Mhz	16 Mhz

Fuente: El Investigador.

Bajo criterios de elección basados en funcionalidad e implementación se determina que el Arduino Uno es el cual presenta las mejores características técnicas en lo que se refiere al número de pines de comunicación así como la interface de comunicación.

En el Anexo A se encuentra la descripción técnica del módulo Arduino Uno.

4.4 Comparación de los ordenadores de placa reducida

Entre las particularidades de mini ordenadores citados en la fundamentación teórica y existentes en el mercado se realizó una tabla comparativa y seleccionar el que mejores características dispone y así aplicarlo al proyecto.

En la tabla 4.2 se describe las características de mini ordenadores.

Tabla 4.2. Tabla comparativa de los Ordenadores de placas reducidas

	RASPBERRY PI	UDDO Quad
GPU	Broadcom VideoCore IV,,60 OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia),58 1080p30 H.264/MPEG-4 AVC3	Vivante GC 2000 para 3D + Vivante GC 355 para 2D (gráficos vectoriales) + Vivante GC 320 para 2D (composición).
ALMACENAMIENTO	SD / MMC / ranura para SDIO	SD
ALIMENTACION	5 V vía Micro USB o GPIO header	(6-15V)
SISTEMAS OPERATIVOS	GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux.RISC OS2	UDOOuntu (Oficial Distribución Lubuntu 12.04 LTS), Android 4.3 Jelly Bean
PRECIO	\$60	\$300

Fuente: El Investigador

Mediante un juicio crítico se ha seleccionado el ordenador Raspberry Pi, debido a sus características técnicas y bajo costo, además de la compatibilidad del software necesario para el proyecto.

4.5 Selección del ordenador de placa reducida Raspberry Pi

Una vez determinado el ordenador se hace necesario un análisis de los distintos tipos de modelos y características que ofrece la gama de Raspberry Pi como se muestra en la tabla siguiente.

Tabla 4.3. Tabla comparativa de la Placas Raspberry Pi

Características	MEMORIA RAM	Puertos USB	GPIO	Ethernet 10/100 MBPS	Precio
Raspberry Pi Modelo A	256 MB LPDDR SDRAM 400 MHZ.	1	25	NO	Bajo
Raspberry pi Modelo B +	256 MB LPDDR SDRAM	4	40	Si	Bajo

	400 MHZ				
Raspberry pi 2 Modelo B	256 MB LPDDR SDRAM	4	40	Si	Medio
	400 MHZ				

Fuente: El Investigador

Mediante un juicio crítico se determinó que le Raspbery Pi modelo B+ es el más indicado para la implementación del ordenador debido a que posee mayor memoria RAM y dispone de 4 entradas de puertos USB los cuales pueden servir para aplicaciones en futuro.

Las características técnicas se muestran en el Anexo B.

4.6 Comparación de las pantallas

En la tabla 4.4. Se detallan las características de cada pantalla

Tabla 4.4. Tabla comparativa de las pantallas Tóuch compatibles para Raspberry Pi.

Características	3.5" TFT LCD DE PANTALLA TÁCTIL	TONTEC 7" RASPBERRY PI LCD TOUCH SCREEN DISPLAY TFT AT070TN90	10.1" DIGITAL PANTALLA DEL IPS
Interface	HDMI	HDMI	HDMI
Resolución	320 × 480	800 * 480 LCD	1280x800
Tipo de pantalla táctil	Resistiva	Resistiva	Resistiva
Consumo de energía	TDB	5-12V	5-12V
Precio	\$40	\$55	\$88

Fuente: El Investigador.

Dentro de la gama de pantallas analizadas en el marco teórico y mostrado en la tabla se determinó que la más apropiada es la TONTEC 7" RASPBERRY PI LCD TOUCH SCREEN DISPLAY TFT AT070TN90 debido a la resolución, precio, características técnicas y la facilidad de adquisición.

En el Anexo C se muestra las características y componentes de la TONTEC 7" Raspberry PI LCD Tóuch Screen Display TFT AT070TN90.

4.7 Selección del módulo Shield

A continuación se muestra una tabla comparativa sobre los diferentes Shields comparativas para Arduino.

Tabla 4.5. Tabla comparativa de los Shields para Arduino

	SHIELD GPRS+GPS QUADBAND	SHIELD 908 GPRS+GPS QUADBAND	PHONE SHIELD SIM908	SHIELD GSM/GPRS QUADBAND
BANDAS	GSM/GPRS	GPRS/GPS	GSM/GPRS/GPS	GSM/GPRS
TIPO	MODEM	MODULO	MODULO	MODULO
MODULO	HILO SAGEN	SIM 908	SIM 908	SIM900
ANTENA	EXTERNA	SEPARADA	SEPARADA	EXTERNA
PRECIO	86	120	120	60
ADQUISICION	DIFICIL	DIFICIL	SI	DIFICIL

Fuente: El Investigador

Se ha seleccionado el Modulo PHONE SHIELD908 GSM/GPRS/GPS debido a las características técnicas, compatibilidad para Arduino y posee la comunicación GPS necesaria para el proyecto.







En el anexo D se puede apreciar las características técnicas del Phone Shield 908 GSM/GPRS/GPS.

4.8 Selección del Sistema Operativo utilizado en la placa Raspberry Pi

Para la instalación del sistema operativo en la Raspberry Pi se ha desarrollado una tabla comparativa entre los Sistemas operativos que soportan procesadores ARM basados en Linux y que están se encuentran disponibles.

En la tabla 4.6. Se muestra la descripción de los sistemas operativos disponibles para Raspberry Pi.

Tabla 4.6. Tabla descriptiva de los S.O. disponibles para Raspeberry Pi.

	Imagen	Descripción
RASPBIAN		Probablemente la distribución más conocida y utilizada en Raspberry Pi y se basa en Debian, con todas las ventajas que ello conlleva. Podemos utilizarla desde la consola, a través de SSH, conectada a un monitor, o a través de un escritorio remoto con VNC.
UBUNTU MATE		Distribución de Linux basada en Ubuntu, modelo de desarrollo es de software libre y de código abierto.
SNAPPI UBUNTU CORE		Es un distribución base, el sistema de gestión de software no es el típico APT sino que es el concepto diferente llamado "SNAPPY"
OSMC		El punto fuerte de OSMC es su ligereza, rapidez y fácil manejo, lo que lo convierten en la opción ideal si "solo" quieres utilizar tu Raspberry como mediacenter.
OPENELEC		Igual que RaspBMC, sirve para la creación de centros multimedia. Ofrece un muy buen rendimiento, aunque prefiero RaspBMC, que tiene más posibilidades que este
PINET		Es un proyecto libre y de código abierto, posee una red basada en cuentas de usuario y sistema operativo basado en red.

Fuente. El Investigador.

Se ha seleccionado el sistema operativo Raspbian ya que cumple con mejores requerimientos técnicos para el desarrollo del proyecto y es compatible para los softwares de entornos gráficos.

Preparación de la placa Raspberry Pi con Sistema Operativo (RASPBIAN)

Para la instalación del sistema operativo en la Raspberry Pi se ejecutó los siguientes pasos que se mencionan a continuación.

1. Descargamos desde una PC el sistema operativo Raspbian de la página Oficial de Raspberry PI.
2. Descomprimos el archivo en una carpeta llamada RARPBERRYISO ubicada en el escritorio de la PC.
3. Copiamos el contenido de la carpeta en una tarjeta SD de 16GB.
4. Insertamos la tarjeta SD con el S.O. en la ranura de la Raspberry PI.
5. Conectamos nuestra Raspberry Pi hacia un periférico de Salida en nuestro caso una pantalla LCD por puerto HDMI.

4.9 Diseño y construcción del Hardware

El presente sistema inicialmente fue concebido mediante el diagrama de bloques de la figura 4.6 razón por la cual se procede de forma similar para el diseño de cada uno de sus componentes.

En la figura 4.7 se detalla la ubicación de los componentes principales del sistema.

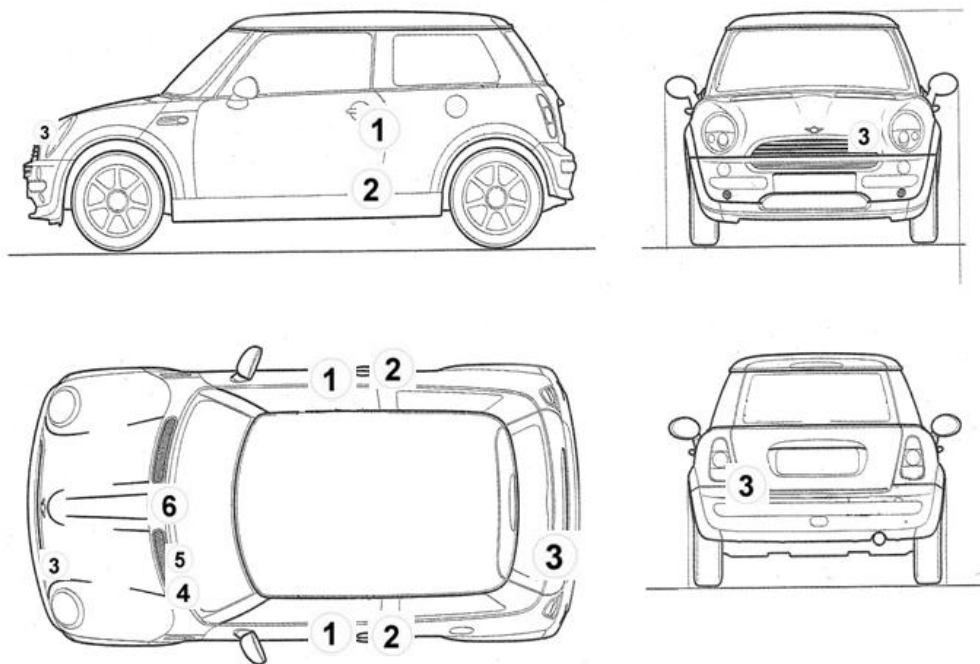


Figura 4.7 Diagrama de ubicación de los componentes del sistema.

Fuente: El Investigador.

1. Motores bloqueo central.
2. Pulsadores de puertas laterales.

3. Pulsador de puertas frontal y posterior.
4. Bloqueo centralizado.
5. Sensor de encendido del vehículo.
6. Ordenador vehicular.

4.9.1 Control de entradas/salidas del ordenador

La interface de entrada tiene por objetivo acoplar los niveles de voltaje proveniente de los sensores externos, a niveles de voltaje que puedan ser manejados por la placa del microcontrolador denominado Arduino, a fin de que estos valores puedan ser censados correctamente por los pines de entrada al Arduino; es decir, valores de 0 lógico y 1 lógico.

La interface de salida tiene por finalidad acoplar los niveles lógicos de voltaje de los pines de salida del Arduino a niveles propios de los actuadores que se quiere controlar.

Como se puede observar en la figura 4.8. Se utiliza los pines digitales D4, D5, D6 de Arduino Uno para el ingreso de las señales de los sensores externos, los pines D8, D9, D10, D11 como salidas del Arduino para conectar a los actuadores, y los pines de Arduino Uno correspondientes a 5V y GND como voltaje de referencia.

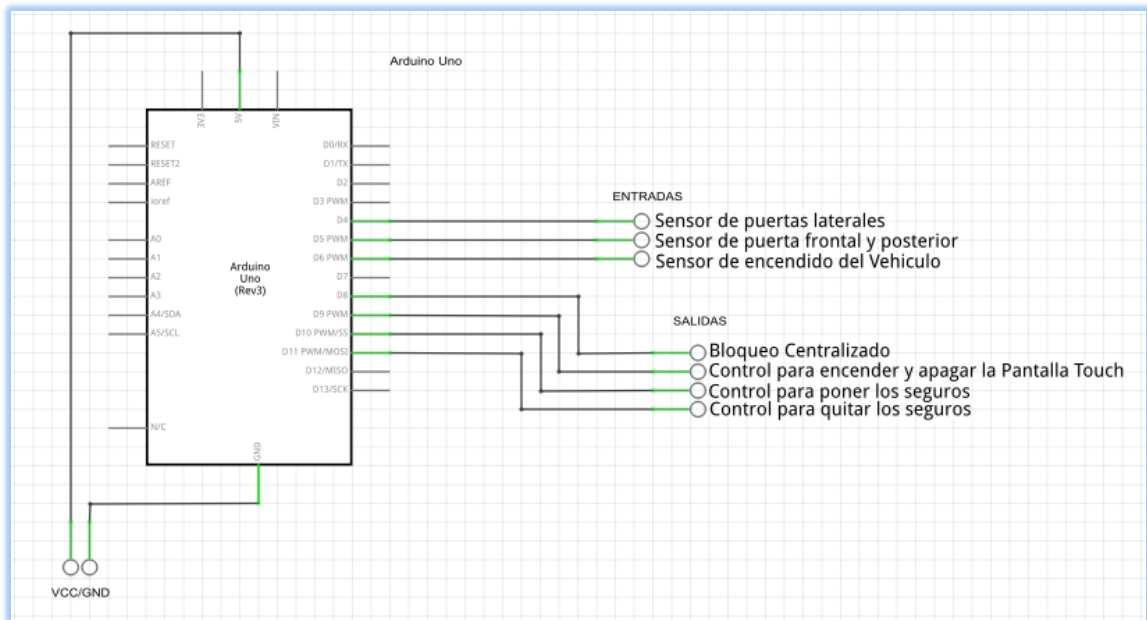


Figura 4.8. Distribución de los pines del Arduino Uno.

Fuente: El Investigador.

En la tabla 4.7 se detalla la función que realiza los pines utilizados de Arduino Uno.

Tabla 4.7. Tabla de función de los pines del Arduino Uno.

Puertos	Función	Numero de pin/Puerto
E1	Recibe la señal del sensor de las puertas laterales	4 digital
E2	Recibe la señal del Sensor del compartimiento frontal y posterior	5 digital
E3	Recibe la señal del sensor de foco de encendido del vehículo	6 digital
S1	Envía una señal para desactivar el bloqueo centralizado	8 digital
S2	Controla el encendido y apagado de la pantalla touch	9 digital
S3	Activar los motores de los seguros de las puertas	10 digital
S4	Desactivar los motores de los seguros de las puertas	11 digital

Fuente: El Investigador.

4.9.2 Diseño de circuitos de acoplamiento

Dado que el microcontrolador ATM328 perteneciente a la placa Arduino requiere cierta flexibilidad en cuanto a los niveles de voltaje de entrada y de salida se hace necesario realizar circuitos de acoplamiento para interactuar con los sensores y actuadores del sistema de alarma.

Realizando un análisis de los sensores y actuadores que se utilizan en el desarrollo se tiene:

Líneas de entrada

Sensor de puertas laterales entrega el valor de 0 lógico cuando la puerta está abierta y circuito abierto cuando la puerta está cerrada.

Sensor de compartimiento frontal y posterior entrega el valor de 0 lógico cuando el compartimiento está abierto y circuito abierto cuando el compartimiento está cerrado.

Sensor de foco de encendido del vehículo entrega un valor de 12V cuando el motor esta encendido y 0V cuando el motor está apagado.

Líneas de Salida

El bloqueo centralizado permite el paso de corriente hacia el motor de arranque con un voltaje de 12V.

La pantalla tóuch dispone de un pulsador que permite encender y apagar la pantalla.

Los motores de bloqueo son servomotores que su función depende de la polarización y trabajan con 12V.

De acuerdo a lo analizado se realizan los siguientes diseños electrónicos:

Circuito para control de sensores de las puertas

Para acoplar los pulsadores de las puertas laterales se realiza un circuito utilizando la lógica de resistencia PULL-DOWN y un control de Relé.

Para alimentar el circuito de Resistencia PULL-DOWN utilizamos el voltaje de 5V que entrega la placa Arduino Uno y la referencia utilizamos el pin GND de la misma placa Arduino Uno, la función del pulsador la realiza el Relé TECHMAN DC12V el cual es activado y desactivado dependiendo el estado de los pulsadores acoplados en las puertas laterales conectados en serie que cierran el circuito como se muestra en la figura 4.9.

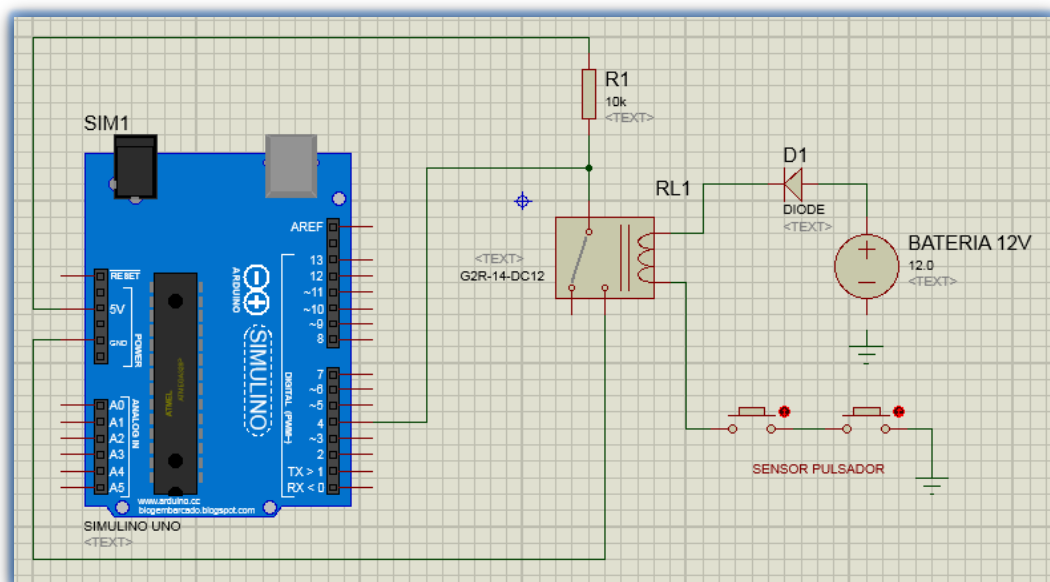


Figura 4.9 Circuito de acoplamiento de los sensores de puertas
Fuente: El Investigador

En el anexo E se encuentra las características del circuito resistencia PULL.DOWN.

Circuito para control de sensor de encendido del vehículo

Para realizar el diseño del circuito de control del encendido del vehículo se utiliza la misma lógica del circuito de la figura anterior, con la diferencia que la señal que se controla es de 12 Voltios de corriente continua cuando el vehículo está encendido y 0 voltios cuando el vehículo está apagado, el diseño se muestra en la figura 4.10.

En la salida del sensor de encendido se conecta un diodo rectificador para evitar corrientes inversas hacia el circuito eléctrico del vehículo, esta corriente permite que el relé TECHMAN DC12V se active cerrando el circuito de resistencia PULL-DOWN, permitiendo así cambiar la lógica entre 0 lógico y 1 lógico.

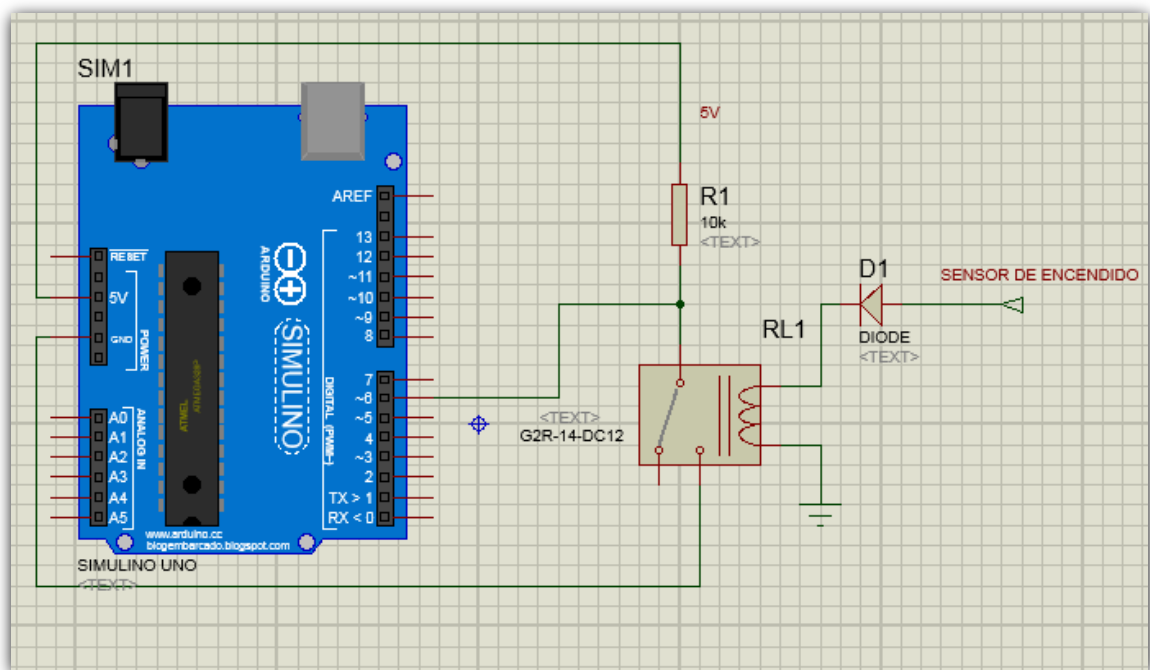


Figura 4.10 Circuito de acoplamiento con la señal de encendido

Fuente: El investigador

En el anexo E se encuentra las características del circuito resistencia PULL.DOWN.

Circuito para el control del bloqueo Central

Para el control de bloqueo central realizamos un circuito de potencia para acoplar con la salida del Arduino Uno, con la finalidad de proteger al microcontrolador.

La salida del Arduino Uno conectamos a un transistor PNP a través de una resistencia para disminuir la corriente, el valor lógico proveniente de la salida del Arduino permite abrir y cerrar el relé, cerrando el circuito y energizando la bobina del relé mediante el transistor PNP como se muestra en la figura siguiente.

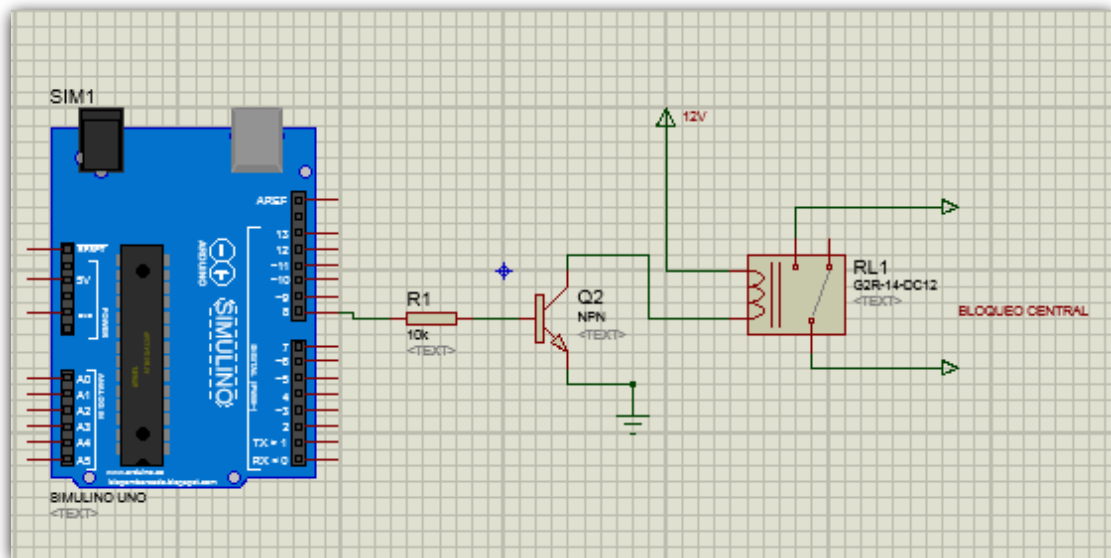


Figura 4.11 Diagrama para control del Bloqueo Central

Fuente: El Investigador

Circuito para el control de motores de bloqueo Central

Para realizar la acción de abrir y cerrar los seguros de las chapas de las puertas se basa en el cambio de giro de un servomotor, se utiliza la misma lógica de la figura anterior con dos Relés, los cuales permiten polarizar directa e inversamente a los motores dependiendo de los valores 0 lógico o 1 lógico de la salida de la placa Arduino y de esta manera cumplir con el objetivo.

En la siguiente figura se muestra el circuito del control de los motores que realizan la función de cierre centralizado de las chapas de las puertas.

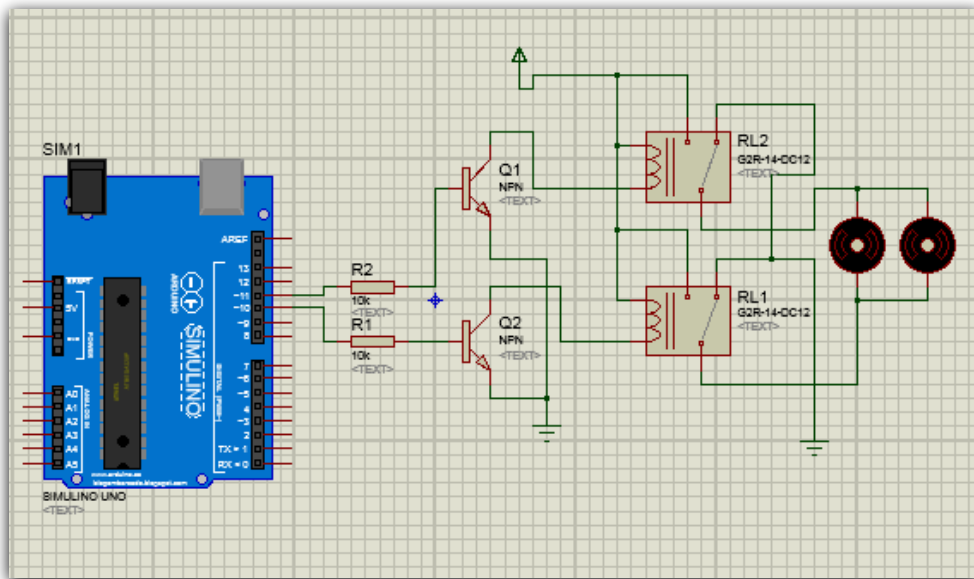


Figura 4.12 Diagrama de conexión para abrir y cerrar los seguros de las puertas
Fuente: El Investigador

Circuito para el control de encendido y apagado de la pantalla touch

Para el encendido y apagado de la Pantalla Touch se utilizó la misma lógica del circuito anterior como se muestra en la figura 4.13.

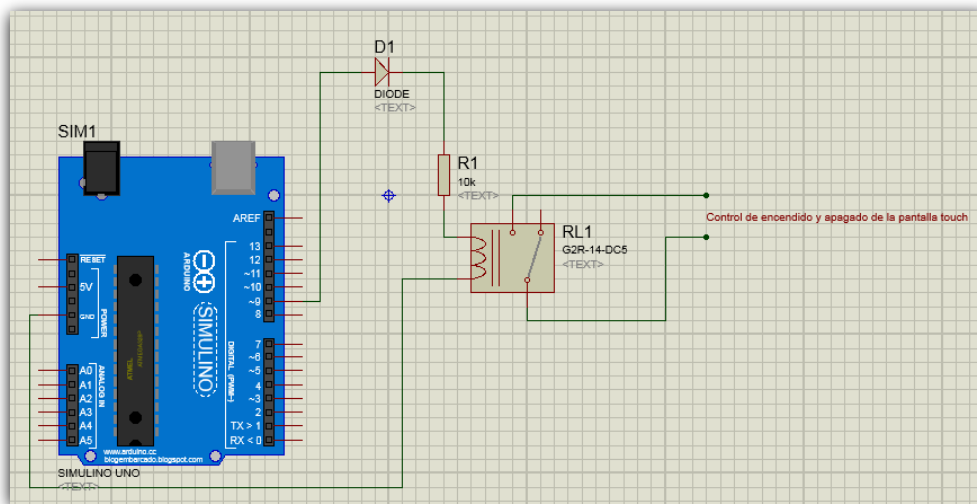


Figura 4.13. Diagrama de control de encendido y apagado de la pantalla touch
Fuente: El Investigador

Circuito esquemático entre el sistema de alarma y Arduino Uno

A continuación se presenta el circuito esquemático final, en el cual se encuentran todos los circuitos interconectados mostrados en la siguiente figura.

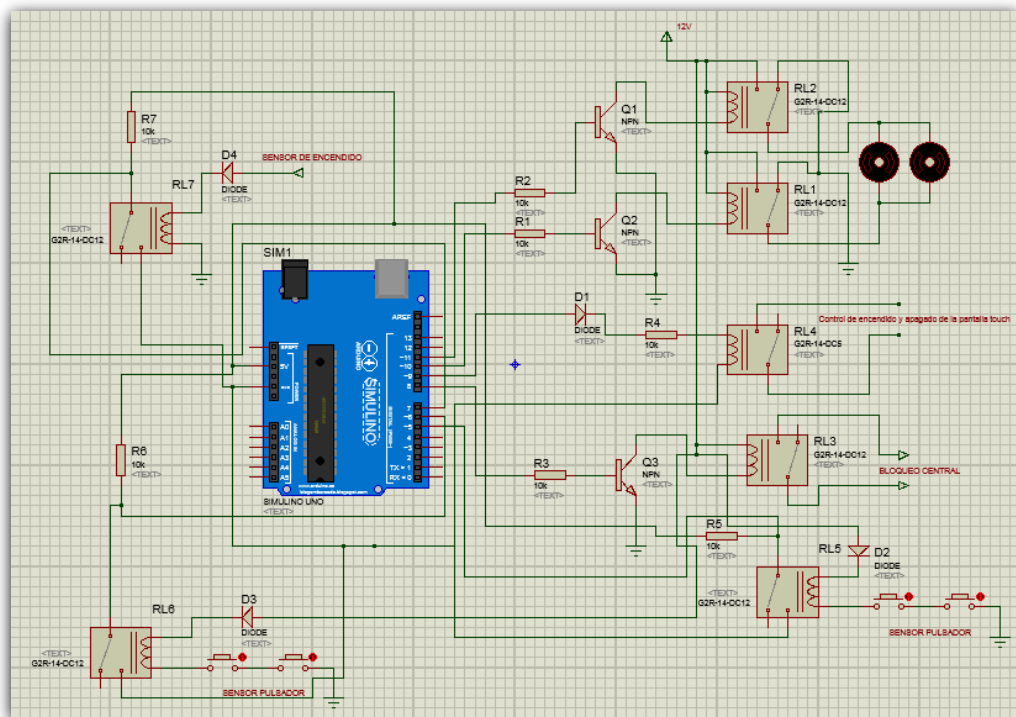


Figura 4.14. Diseño esquemático entre el sistema de alarma y Arduino Uno
Fuente: El Investigador

4.9.3 Conexión entre Arduino Uno y Raspberry Pi

Para establecer la comunicación entre las placas de Arduino Uno y Raspberry Pi, la realizamos a través de un cable USB como se muestra en la siguiente figura.

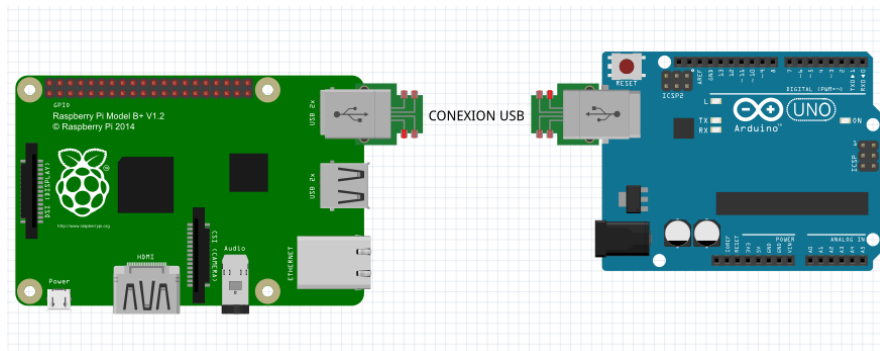


Figura 4.15 Conexión entre Raspberry Pi y Arduino Uno
Fuente: El Investigador

4.9.4 Conexión de Arduino Uno con Phone Shield Sim908 GSM/GPRS/GPS

Insertamos el módulo Phone Shield sim 908 GSM/GPRS/GPS sobre la placa Arduino Uno. Podemos ver el montaje en la figura 4.16.

La “ceja” que separa los bloques de los pines del Arduino Uno determina la ubicación correcta de los pines de Phone Shield Sim908 sobre la placa del Arduino Uno.



Figura 4.16 Plataforma Arduino Uno + Shield Sim908 GSM/GPRS/GPS.
Fuente: El Investigador.

4.9.5 Conexión de Raspberry Pi CON TONTEC 7" Raspberry Pi LCD Touch Screen Display TFT AT070TN90

Para conectar nuestra Raspberry Pi con TONTEC 7" Raspberry Pi LCD Touch Screen Display TFT AT070TN90, lo realizamos a través de un cable HDMI entre la Raspberry Pi y la placa controladora y el touch a través de una placa como se muestra en la figura 4.17.



Figura 4.17 Interface entre Raspberry Pi con LCD touch 7".
Fuente: <http://www.aliexpress.com/item-img/7-Inch-TFT-Touch-Screen-LCD-Monitor-For-Raspberry-Pi-Driver-Board-HDMI-VGA-2AV-Provide/1846532213.html>.

4.9.6 Diseño de circuito de alimentación

Para energizar el ordenador Vehicular se analiza el consumo de corriente y el voltaje que soporta cada uno de los módulos que forman parte del sistema, para esto se toma la información que se adquirió en el Capítulo II obteniendo la siguiente Tabla.

Tabla 4.8. Tabla de Rango de voltajes de los Modulos del Sistema

MODULO	VOLTAJE (V)	CORRIENTE (A)
Arduino Uno	7 - 12 V	0.5 - 1.5 A
Raspberry Pi Modelo B+	0 - 10 V	0.5 - 2 A
Phone Shield Sim 908 GSM/GPRS/GPS	6 - 9 V	0.5 - 1 A
TONTEC 7" Raspberry Pi LCD Touch Screen Display TFT AT070TN90	7 - 24 V	0.5 - 2 A

Fuente: El Investigador

La corriente que entrega el vehículo es de 12 V por ende se alimenta directamente a los módulos *Arduino Uno* y *TONTEC 7" Raspberry Pi LCD Touch Screen Display TFT AT070TN90*

En la figura 4.18. Se muestra el diagrama acondicionamiento del voltaje de 12 Voltios que genera el vehículo para alimentar el Ordenador Vehicular.

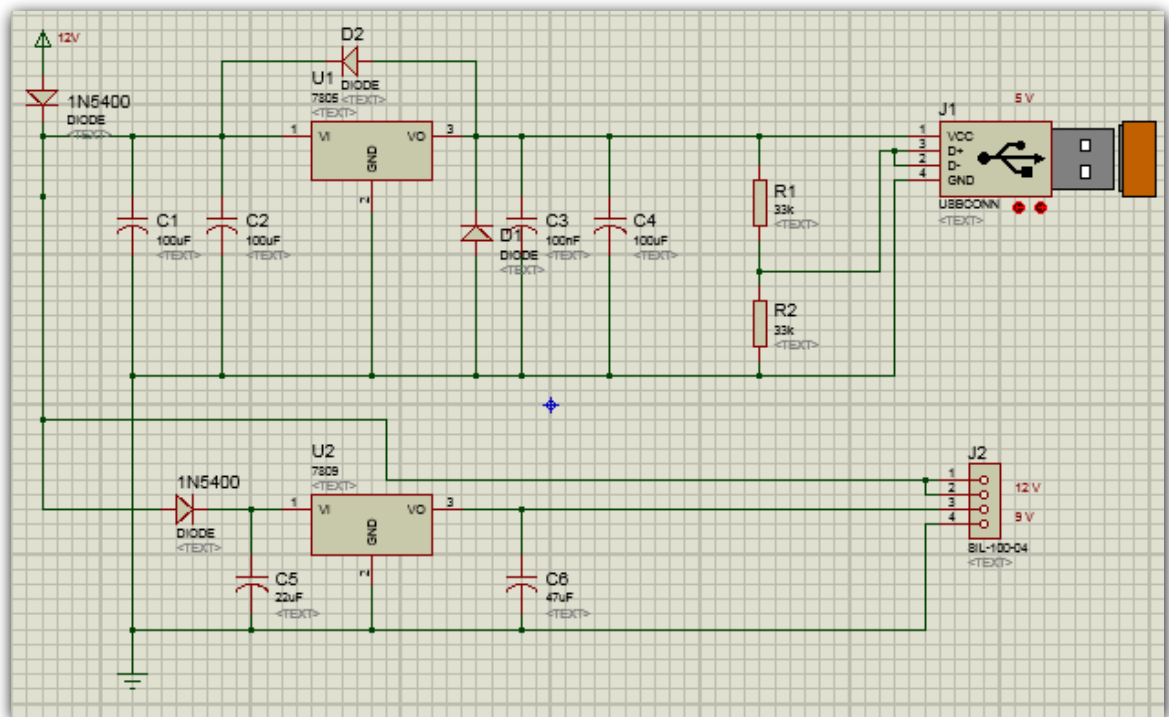


Figura 4.18. Acondicionamiento de alimentación para el Sistema
Fuente: El Investigador

4.10 Desarrollo del software del sistema. Modos de funcionamiento

En la presente parte se inicia dando una visión general de las funcionalidades que se busca en el Sistema del Proyecto, de donde se parte para el desarrollo de los correspondientes diagramas de flujo que son utilizados en la programación de cada uno de los módulos.

Para brindar flexibilidad al módulo de funcionamiento del sistema se considera tres modos de trabajo los cuales permiten al usuario interactuar con el ordenador electrónico Vehicular.

A continuación se detalla los tres modos de funcionamiento.

Modo armado

Este modo de funcionamiento ha sido diseñado para trabajar conjuntamente con los sensores y actuadores pertenecientes a la alarma del vehículo. En este modo se tendrá la opción de monitorear y controlar los eventos de los sensores y actuadores, mediante

comando AT que el ordenador reciba e interactuar con celular del usuario cuando este haya ejecutado el comando de armado previamente el sistema.

Modo desarmado

En este modo de funcionamiento el sistema permite obtener la ubicación GPS del vehículo a través de la interacción entre el usuario y el ordenador vehicular usando comandos AT hacia una aplicación móvil, por otra parte el usuario mantendrá una comunicación directa con el ordenador vehicular permitiendo desactivar el bloqueo central el cual se activa automáticamente cuando el vehículo es apagado. Además, el ordenador no enviara información adicional mediante comando AT al celular del usuario aunque el sistema de alarma correspondiente a los sensores se haya activado.

Modo Independiente

El presente modo de funcionamiento ha sido diseñado para mantener un control interno para el usuario sobre el estado del sistema de seguridad en un entorno gráfico, esto implica la apertura y cierre de las puertas.

4.10.1 Instalación del Sistema Operativo Rasbian en la Raspberry Pi

Cuando ponemos en marcha la Raspbian la primera ventana que observamos es la selección del Sistema operativo como se muestra en la figura 4.19.

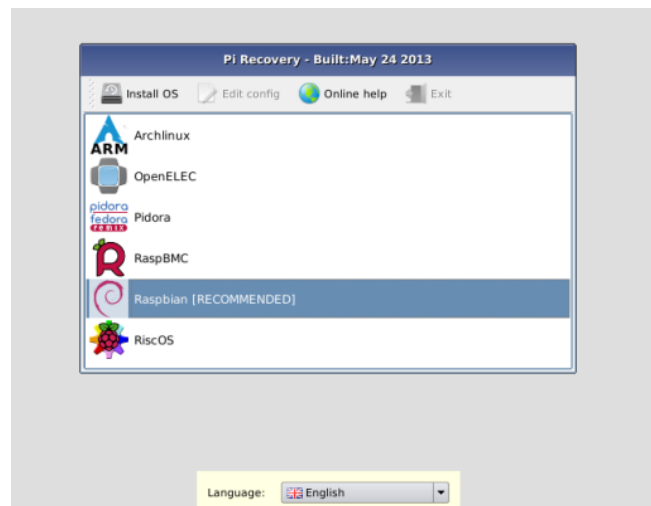


Figura 4.19. Sistemas operativos disponibles para la instalación
Fuente: El Investigador

Una vez seleccionado el sistema operativo se procede a configurar los siguientes parámetros que se expresan en un menú como se muestra en la figura 4.20

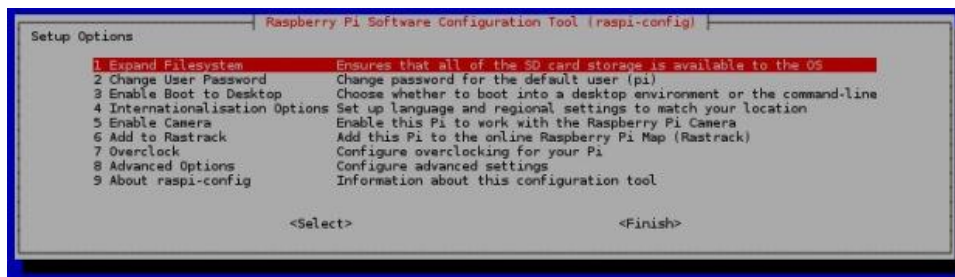


Figura 4.20 Menú principal de las opciones de configuración de Raspberry pi
Fuente: El Investigador

El sistema operativo de Raspberry Pi posee ciertas opciones al momento de configuración las cuales se enlistan a continuación

- Opción 1 – Expandir el sistema de archivos (Expand Filesystem)
Esta opción permitió expandir el sistema operativo para que utilice todo el espacio disponible en la tarjeta.
- Opción 2 – Cambiar la contraseña del usuario Pi (Change User Password)
Esta opción permitió cambiar la contraseña de usuario pi la cual viene de fábrica.
- Opción 3. Activar el escritorio al iniciar (Entable Boot to Desktop)
Esta opción permitió que el Raspberry Pi después de iniciar el sistema, comience inmediatamente el escritorio modo gráfico.
- Opción 4. Opciones de internacionalización (Internationalisation Options)
Esta opción permitió modificar el lenguaje del sistema operativo, la zona horaria y la distribución de su teclado.
- Opción 5 – Activar la cámara (Enable camera).
- Opción 6 – Adicionar a Rastrack (Add to Rastrack)
Esta opción permitió que Raspberry Pi sea rastreado por el sitio web Rastrack (<http://rastrack.co.uk>), este sitio no pretende registrar o recolectar información alguna.
- Opción 7 – Overclockig
Esta opción permitió aumentar la velocidad del procesador.
- Opción 8. Opciones avanzadas (Advanced Options)

Esta opción presenta un otro submenú con las siguientes opciones.

- La opción A1 overscan sirvió para borrar las líneas negras en algunos monitores o televisores.
- A2 Hostname, sirvió para identificar su Raspberry Pi en su red local.
- La opción A3 – permitió Distribuir la memoria (Memory Split).
- La opción A4 – permitió para activar SSH (Enable SSH).
- La opción A5 – Actualizar (update).

Después de configurar las opciones básicas de Raspberry PI se mostrará el entorno gráfico como se muestra en la figura 4.21.

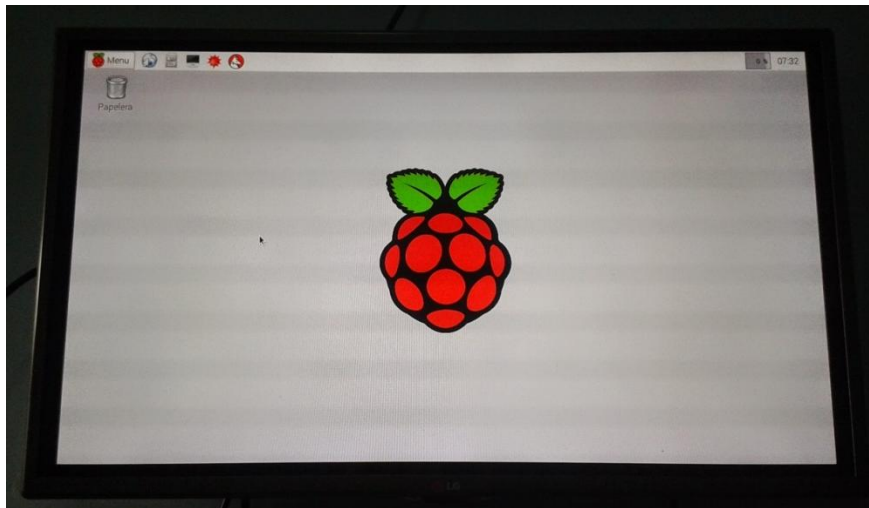


Figura 4.21 Entorno grafico de Raspberry PI

Fuente: El Investigador.

4.10.2 Instalación de Processing en Raspberry Pi

Para la instalación de Processing en nuestro Raspberry Pi seguimos una serie de instrucciones que se muestran a continuación:

1. Conectamos nuestra raspberry pi a internet por medio de un cable Ethernet a un Router
2. Ejecutamos el modo de comandos
3. Instalamos ORACLE JAVA ejecutando el comando:
`sudo apt-get install oracle-java7-jdk`
4. Creamos un directorio personal en nuestra raspberry Pi llamado **Procesing**.

5. Descargamos el archivo ejecutable de la página <https://processing.org/download/?processing> la versión de Linux de 32 bits, y copiamos el archivo en el directorio creado en nuestro caso **Procesing**
6. Extraemos el archivo desde el directorio ejecutando el siguiente comando
`procesamiento-2,1-linux32.tgz tar xvzf`
7. Ahora quitamos el tiempo de ejecución de Java *x86* incluido, y reemplazarlo con *armhf* del Raspberry Pi uno:
`rm -rf ~/procesamiento-2.1/java ln -s /usr/lib/jvm/jdk-7-oracle-armhf ~/procesamiento-2.1/java`
8. Para instalar el conector de serie simple de java descargamos el archivo `jSSC-2.6.0-Release.zip` de la página <https://code.google.com/p/java-simple-serial-connector/downloads/detail?name=jSSC-2.6.0-Release.zip&can=2&q=>
9. Ahora sobrescribir el `jssc.jar` que se incluye con procesamiento con el que se acaba de descargar:
`mv JSSC-2.6.0-LANZAMIENTO/jssc.jar ~/procesamiento-2.1/modos/java/bibliotecas/serie/biblioteca/`
10. Ejecutamos Processing. La figura 4.22. muestra el entorno de desarrollo de processing.

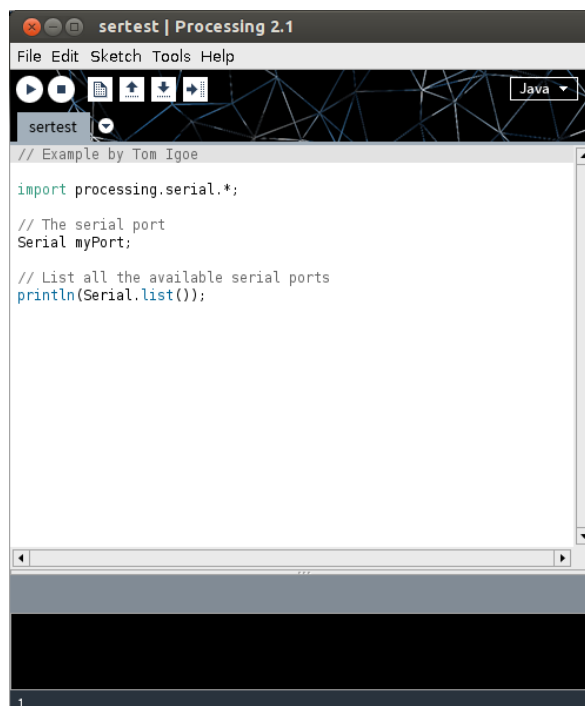


Figura 4.22 Entorno de programación de Processing
Fuente: El Investigador

4.10.3 Configuración de Arduino Uno

Arduino permite mantener comunicación entre Usuario-Maquina y Maquina-Usuario acoplado al ordenador de placa reducida Raspberry Pi. Esta comunicación se la realiza a través de distintos métodos, y las funciones se basan en la figura 4.6 la cual fue contemplada y analizada.

Arduino permite obtener datos del GPS y enviarlos al usuario además de controlar el sistema de alarma del vehículo.

Para conseguir este propósito seguimos los siguientes pasos que se detallan a continuación:

a. Configuración del Phone Shield Sim908 para uso de GSM

Para poder utilizar el servicio GSM que dispone el Phone Shield Sim908 GSM/GPRS/GPS establecemos una comunicación entre el pin GSM RX con el pin D3, y el pin GSM TX con el pin D2 de la interface de control de la placa del Phone Shield SIM908 GSM/GPRS/GPS, de esta manera conseguimos activar el servicio GSM como se muestra en la siguiente figura.

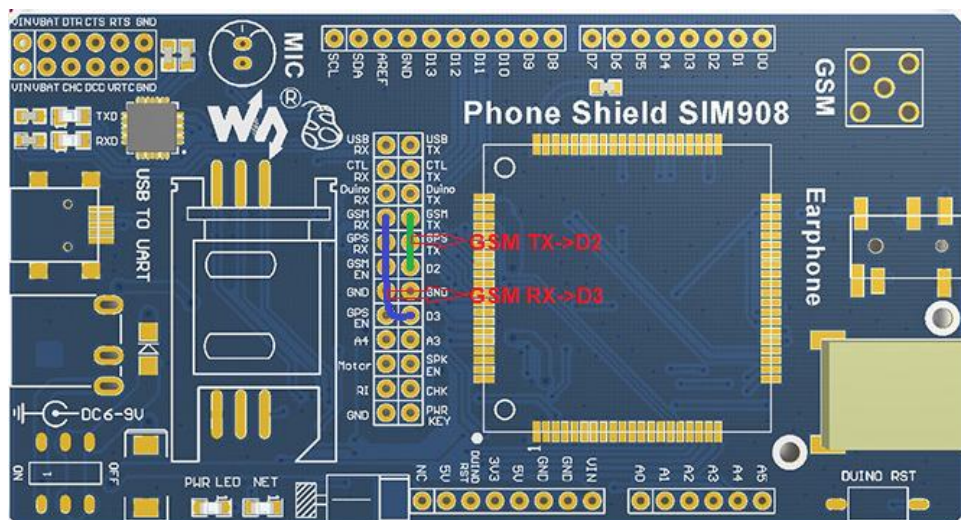


Figura 4.23. Configuración de la placa Phone Shield SIM908 para uso del servicio GSM.

Fuente El Investigador.

b. Configuración del Phone Shield Sim908 para uso del GPS

Para poder utilizar el servicio GPS que dispone el Phone Shield Sim908 GSM/GPRS/GPS establecemos una comunicación entre el pin GPS EN con el pin GND de la Interface de Control de la placa del Phone Shield SIM908 GSM/GPRS/GPS, de esta manera conseguimos activar el servicio GPS como se muestra en la siguiente figura.

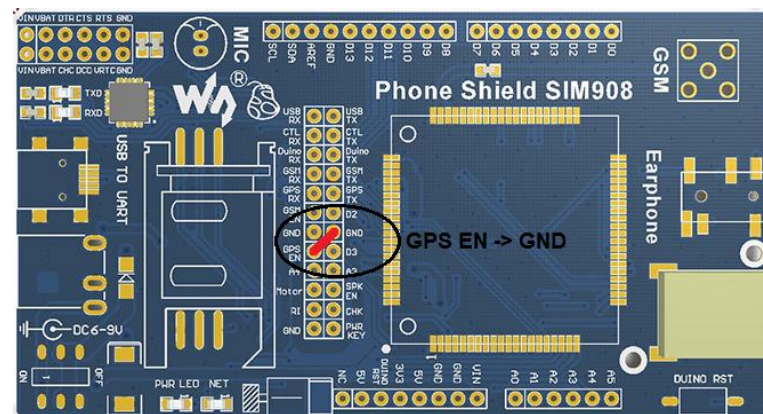


Figura 4.24 Configuración de la placa Phone Shield Sim908 para uso del servicio GPS.
Fuente: El Investigador

c. Montaje de la placa Arduino Uno con el Phone Shield Sim908

Conectamos la antena GSM y la antena GPS en la Phone Shield Sim908 y montamos sobre el Arduino Uno de tal manera que generamos comunicación entre las dos placas como se muestra en la figura 4.25.



Figura 4.25. Montaje de la Phone Shield Sim908 con Arduino Uno
Fuente: El Investigador

d. Creación del Sketch para manejos de datos GPS a través de comandos AT

Generamos un sketch utilizando la librería GPS de Arduino mediante la cual podemos adquirir los datos que recepta el Phone Shield Sim908 y los envía a través de comandos AT.

En el Anexo F se muestra el sketch generado para Arduino.

El código generado nos permite extraer los datos del GPS del Phone Shied Sim908 que son *latitud* y *longitud* y enviarlos cuando el usuario los solicite mediante comando AT.

Mediante el sketch creado, el editor de texto está compuesto de tres partes fundamental:

- La primera parte consta del encabezado del programa en el cual importamos las librerías y declaramos los tipos de variables que se utilizaran en el desarrollo del esquema.
- La segunda parte está compuesta por la función llamada *void setup*, que constituye la preparación del programa, en la cual abrimos comunicación con el GPS del Phone shield SIM908.
- La tercera parte consta de la función *void loop*, en esta parte permite realizar la ejecución del programa, en la cual se realiza la condiciones para enviar las Coordenadas por comandos AT.

Una vez escrito dicho código, pulsamos sobre el icono “Verificar” para que lo analice en busca de posibles errores. Si todo es correcto, en el área de mensajes aparecerá la palabra ‘Compilado’, tal y como se muestra en la figura 4.26.

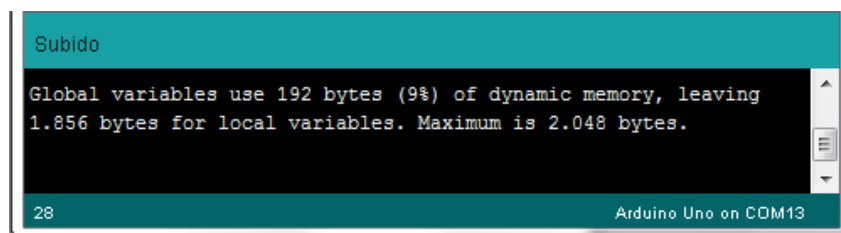


Figura 4.26. Verificación de posibles errores en el ID de Arduino

Fuente: El Investigador

Una vez cargado el sketch, abrimos el “Monitor Serial” y observamos el comportamiento del programa, en este caso se muestra los datos de *latitud* y *longitud* como se muestra en la figura 4.27.

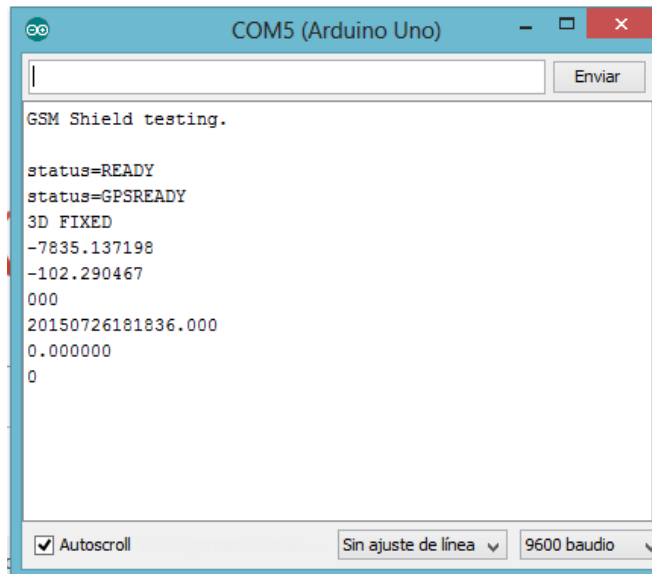


Figura 4.27. Datos del GPS mostrados en el Monitor Serial del ID de Arduino

Fuente: El Investigador

e. Creación del Sketch para control del Sistema de Seguridad.

Un análisis sobre el comportamiento del Arduino basado en la figura 4.6 el cual consta de dos modos de comunicación los cuales se detalla a continuación:

a. Comunicación de Arduino Uno con un Smartphone a través de comandos AT.

Desde Smartphone envía los siguientes comandos:

- Activar
- Desactivar
- Ubicación

Al recibir por mensaje la palabra “Activar”, Arduino ejecuta las siguientes funciones:

- Bloquea los seguros de las puertas de acceso al vehículo.
- Bloquea la pantalla con eso el acceso que desactiva el bloqueo central desde el vehículo.

Condiciones: Si alguna de las puertas o Capo del vehículo es forzado el Arduino devuelve un mensaje hacia el Smartphone con la descripción del sensor activado como.

- Puerta lateral abierta.
- Puerta frontal o posterior abierta.

Al recibir por mensaje la palabra “Desactivar”, Arduino ejecuta las siguientes funciones:

- Desbloquea los seguros de las puertas de acceso al vehículo
- Activa la pantalla permitiendo la manipulación y visualización para el usuario

Al recibir por mensaje la palabra “Ubicación” Arduino Uno ejecuta la siguiente función

- Devuelve los datos de latitud y longitud entregados por el GPS del Phone Shield Sim908 GSM/GPRS/GPS por comandos AT hacia el celular del Usuario.

b. Arduino Uno se encarga de establecer comunicación con Raspberry Pi a través del puerto USB y cumple las siguientes funciones.

- Recibe un dato específico de Raspberry Pi que permite desactivar el bloqueo centralizado.
- Transmite la información del comportamiento de los sensores Ubicados en el Vehículo.

Generamos un sketch en base a las condiciones establecidas como se muestra en el Anexo G.

El código generado nos permite la activación y desactivación del sistema de alarma mediante comando AT además de transmitir datos de los sensores a través del puerto serie.

Mediante el sketch creado, el editor de texto está compuesto de tres partes fundamentales:

- La primera parte consta del encabezado del programa en el cual importamos las librerías y declaramos los tipos de variables que se utilizarán en el desarrollo del esquema.
- La segunda parte está compuesta por la función llamada *void setup*, que constituye la preparación del programa, en la cual abrimos comunicación entre

Arduino y Processing y la transmisión de datos a través de SMS mediante el Phone shield SIM 908.

- La tercera parte consta de la función *void loop*, esta parte contiene el código que ejecuta continuamente (lectura de entradas, activación de salidas y transmisión de datos), esta función es el núcleo del programa y la realiza la mayor parte del trabajo ya que interactúa con el usuario a través de comandos AT.

Una vez escrito dicho código, pulsamos sobre el icono “Verificar” para que lo analice en busca de posibles errores. Si todo es correcto, en el área de mensajes aparecerá la palabra ‘Compilado’, tal y como se muestra en la figura 4.28

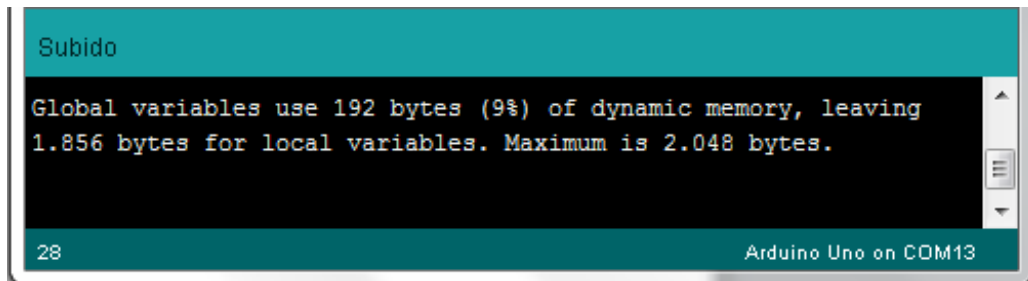


Figura 4.28 Verificación de posibles errores del sketch en el ID de Arduino

Fuente: El Investigador

Una vez cargado el sketch, abrimos el “Monitor Serial” y observamos el comportamiento del programa, en este caso se observa los datos de la comunicación serial como se muestra en la figura 4.29

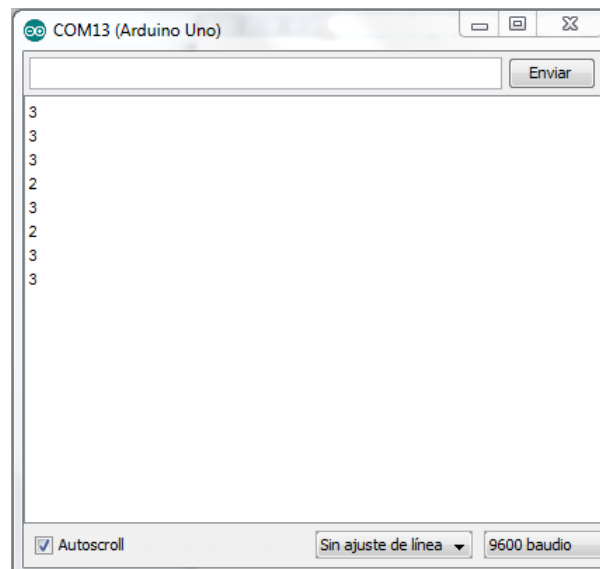


Figura 4.29 Datos de sensores mostrados en el Monitor Serial del ID de Arduino

Fuente: El Investigador

4.10.4 Programación del Sketch en Processing

Para la programación de entorno grafico en Processing, tomamos los datos que Arduino transmite a través por el puerto USB y los procesamos generando un código dentro del Sketch de Processing de tal manera que creamos un entorno grafico que el usuario pueda interpretarlo, el cual consta de un control de ingreso clave que al ser el correcto desactiva el bloqueo centralizado del vehículo permitiendo el encendido del mismo.

En la figura 4.30 se muestra la interface Usuario-Maquina

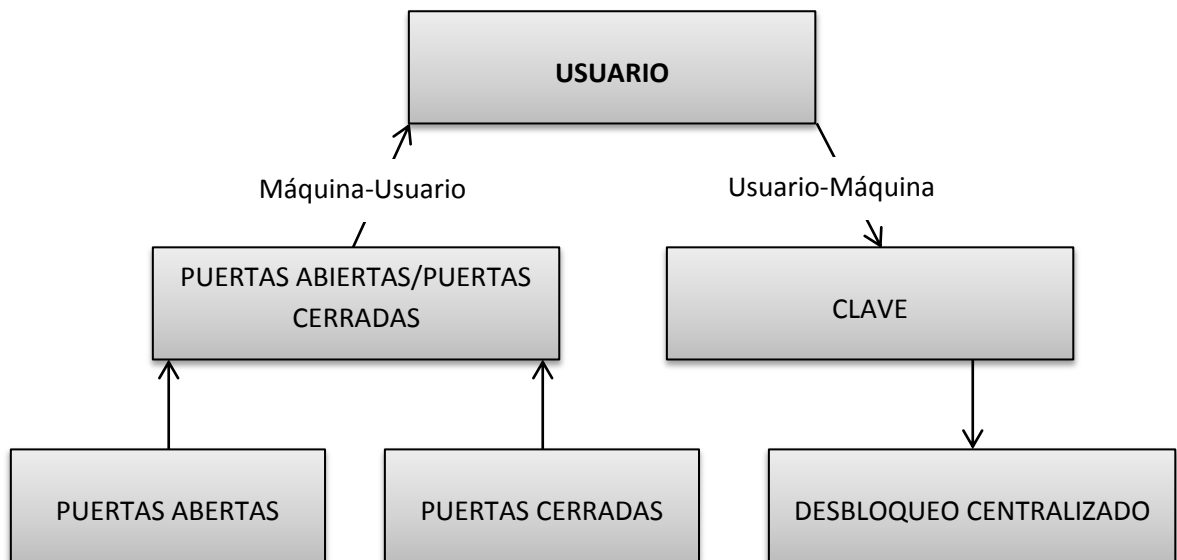


Figura 4.30 Interface Usuario-Máquina, Máquina-Usuario
Fuente: Investigador

Una vez realizado y analizado el diagrama de flujo lo generamos en un sketch de programación con líneas de comando en lenguaje Java para Processing.

En el Anexo H se muestra el Sketch de programación para Processing en el cual se interpreta el diagrama de flujo mostrado en la figura 4.30.

Mediante el sketch creado, el editor de texto está compuesto de Cuatro partes fundamental:

- La primera parte consta del encabezado del programa en el cual importamos las librerías y declaramos los tipos de variables que se utilizarán en el desarrollo del esquema.
- La segunda parte está compuesta por la función llamada *void setup*, que constituye la preparación del programa, en la cual abrimos comunicación con el puerto serial de comunicación al Arduino, cargamos las imágenes que se encuentran en la carpeta *data* perteneciente al mismo programa que fue creada automáticamente al momento de crear el archivo, guardamos valores en las variables declaradas en el encabezado.
- La tercera parte consta de la función *void drawn*, en esta parte recogemos los datos que Arduino envía a través del puerto *Serial* para compararlos y mostrarlos en un entorno gráfico, además graficamos figuras circulares para que el usuario las pueda visualizar y diferenciar.
- La cuarta parte consta de la función *void update*, en esta parte a las figuras creadas en el entorno gráfico de la parte anterior, las determinamos como zonas en el cual el programa pueda identificar cuando el usuario las seleccione y de esta manera creamos la secuencia correcta de la clave que el usuario ingrese.

Ingresamos este sketch en el entorno de Processing como se muestra en la figura 4.31.

```

File Edit Sketch Tools Help
tesis1
import processing.serial.*;
Serial puerto;
PrintWriter output;
int valor;
int aux;
int salto;
int dato;
PImage autoabierto;
PImage fondo;
PImage nivelfuel;
PImage temp;
PImage aceite;
PImage bat;
PImage engine;
PImage close;
PImage dooropen;
PImage capopen;

int circleX, circleY; // Position of circle button
int circleX1, circleY1; // Position of circle button
int circleX2, circleY2; // Position of circle button
int circleX3, circleY3; // Position of circle button
int circleX4, circleY4; // Position of circle button
int circleX5, circleY5; // Position of circle button
int circleSize = 50; // Diameter of circle
int circleSize1 = 50; // Diameter of circle

```

Figura 4.31 Sketch en el software de Processing
Fuente: El Investigador.

Para codificar el sketch creado y mostrar en un entorno grafico compilamos el programa

presionando  que corresponde a “RUN” disponible en el entorno de Processing.

El entorno grafico de Processing corresponde a dos partes, una en la cual se visualizará el estado de los sensores, y la segunda parte en la cual el usuario podrá seleccionar una secuencia que corresponda a la clave que permita desbloquear el bloqueo centralizado del vehículo como se muestra en la figura 4.32.

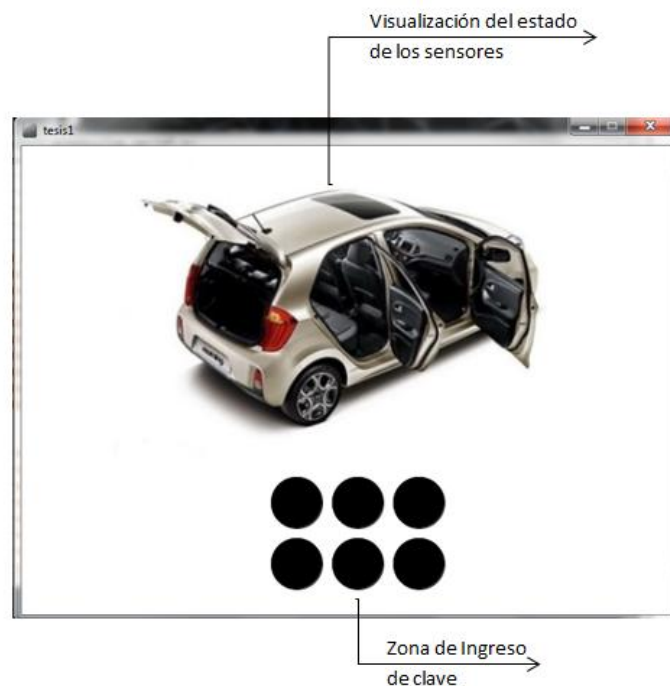


Figura 4.32 Entorno Grafico generado por el sketch en Processing.
Fuente: El Investigador

4.10.5 Creación de una Aplicación Móvil con App-Inventor

Para la creación de la aplicación móvil utilizamos la herramienta de programación App-Inventor beta disponible en la web, para realizar la comunicación entre la aplicación y el ordenador Vehicular, lo realizamos mediante comandos AT bajo las siguientes condiciones:

- Al presionar el botón denominado “Activar”, envía un mensaje de texto hacia el ordenador vehicular el cual ejecutara la acción determinada.
- Al presionar el botón denominado “Desactivar”, envía un mensaje de texto hacia el ordenador vehicular el cual ejecutara la acción determinada.

- Al presionar el botón denominado “Obtener Ubicación” envía un mensaje de texto hacia el ordenador vehicular el cual interpretara el código y reenviara un mensaje de texto con las coordenadas de la ubicación en tiempo real del vehículo.
- Al presionar el botón denominado “buscar en Google Maps” la aplicación introduce el código de la ubicación que envía el ordenador vehicular y lo muestra en la aplicación de Google Maps.

En la figura 4.33. Se muestra el diagrama de bloques del funcionamiento de la aplicación Móvil.

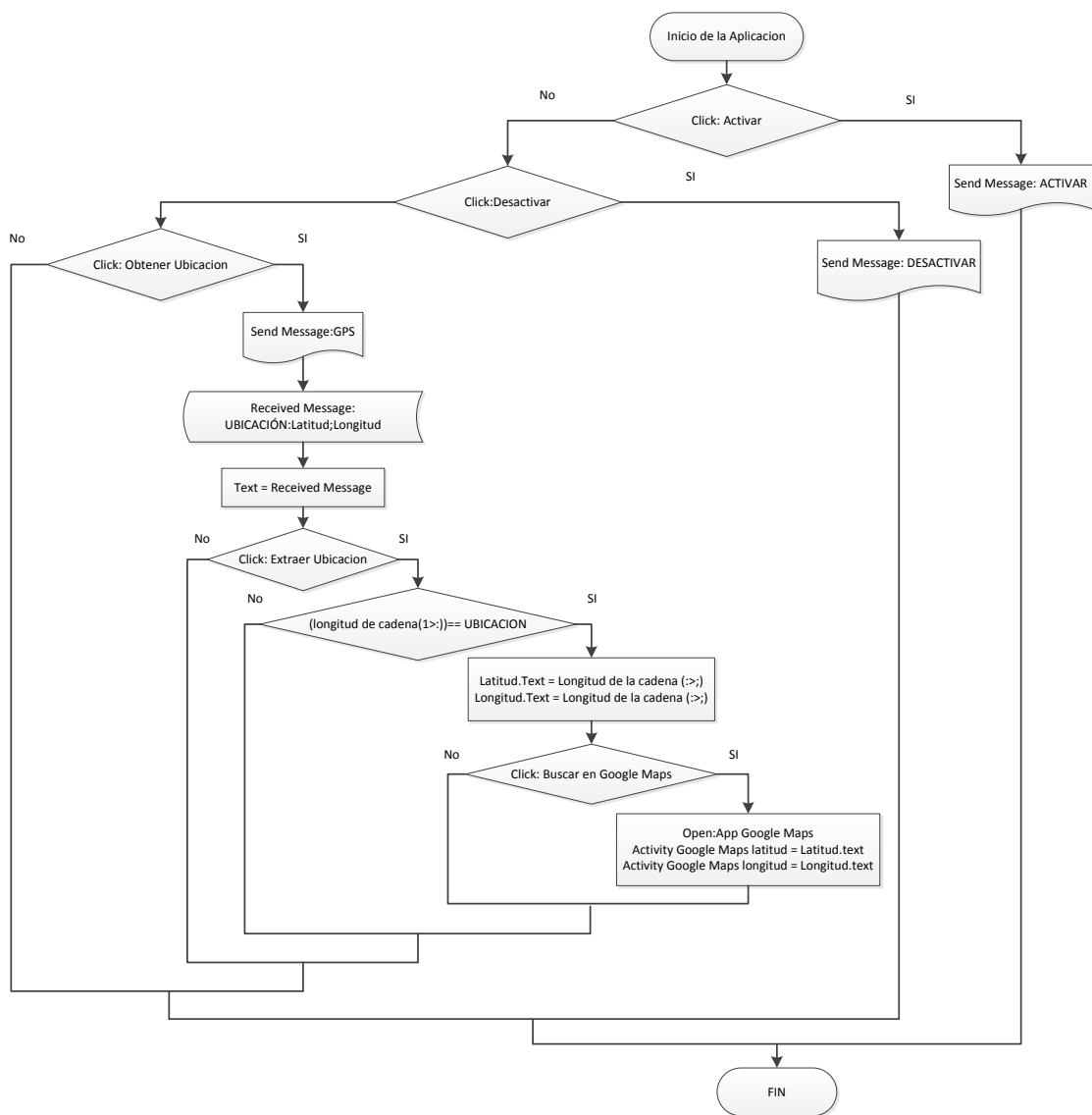


Figura 4.33 Diagrama de flujo del funcionamiento de la Aplicación Móvil

Fuente: El Investigador

Para la creación de la aplicación se procede a realizar los siguientes pasos.

1. Ingresamos a la herramienta de programación App-inventor beta mediante la dirección web *beta.appinventor.mit.edu*.
2. Click en Acceder y se obtendrá el entorno del diseño de nuestra aplicación como se muestra en la figura 4.34.

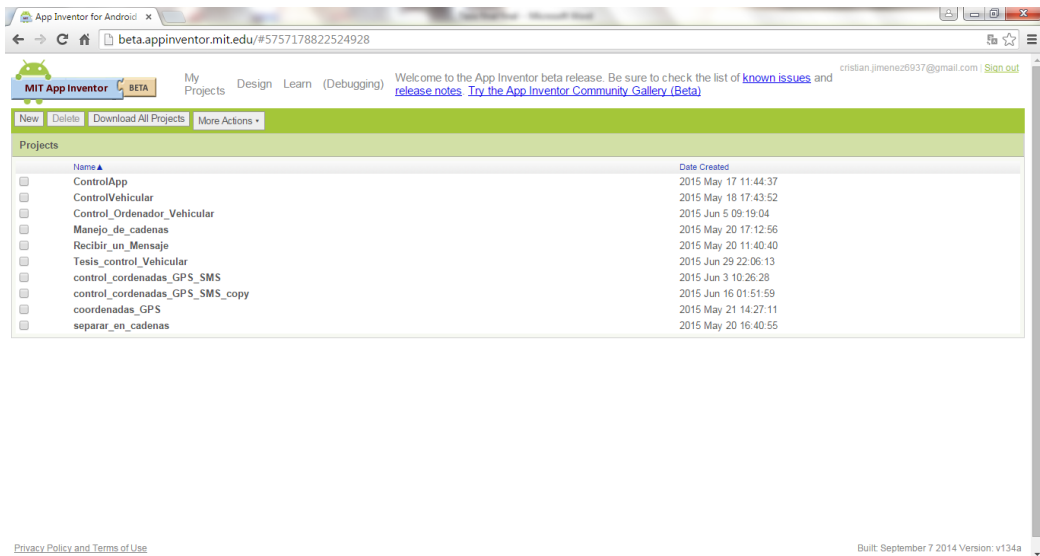


Figura 4.34 Versión beta de app inventor
Fuente: El Investigador

3. Click en New e ingresamos el nombre del proyecto en nuestro caso Tesis_Ordenador_vehicular como se muestra en la figura 4.35.

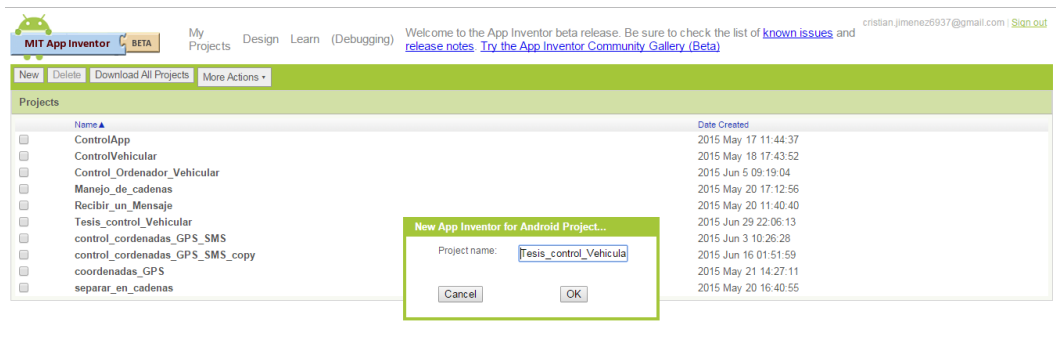


Figura 4.35 Creación de un nuevo proyecto en App Inventor Beta
Fuente: El investigador

4. Para diseñar la interface del usuario creamos un proyecto llamado Tesis_Ordenador_vehicular y nos aparece una imagen el blanco como se muestra en la figura 4.36

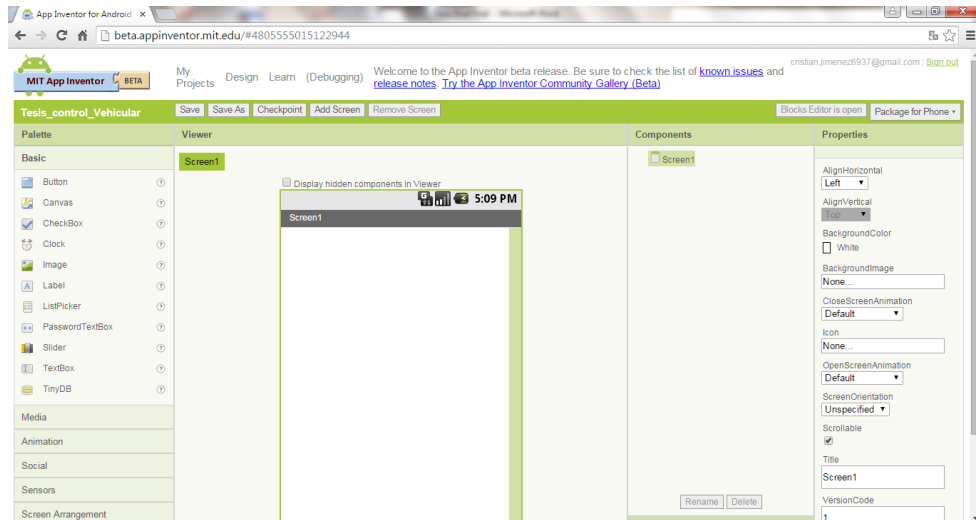


Figura 4.36 Pantalla de diseño de App-Inventor beta
Fuente: El Investigador

5. Diseñamos la pantalla, para lo cual utilizamos los siguientes componentes: *Image1*, *HorizontalArrangement1*, *Button1*, *Label1*, *Label2*, *Button2*, *VerticalArrangement2*, *Button3*, *Label1*, *HorizontalArrangement2*, *Label2*, *Label3*, *HorizontalArrangement3*, *Label4*, *Label5*, *Button4*, *ActivityStarter1*, *Texting1*. Nuestra pantalla inicial con estos componentes agregados lucen aun sin configurar (salvo *VerticalArrengement1*,) como se muestra en la Figura 4.37 El componente *TableArrengement* fue configurado para dar una mejor organización a los componentes.

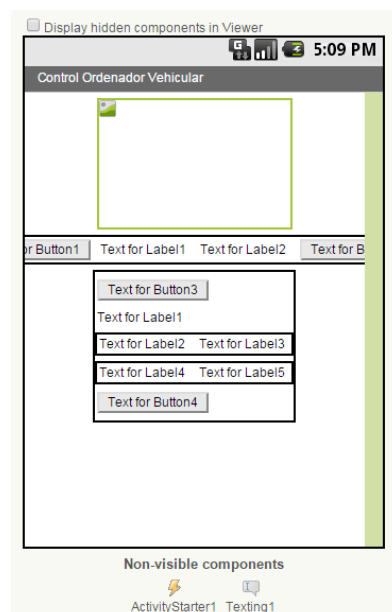


Figura 4.37. Componentes agregados aun sin configurar
Fuente: El Investigador

- Configuramos el componente *ActivityStarter1* con los comandos mostrados en la tabla. En la tabla 4.9 se muestra el componente *ActivityStarter1* con los cambios realizados.

Tabla 4.9. Lista de campos para configurar el componente ActivityStarter1

Nombre del Campo	Campo
Action	android.intent.action.VIEW
ActivityClass	com.google.android.maps.MapsActivity
ActivityPackage	com.google.android.apps.maps

Fuente: El Investigador

En la figura 4.38 se muestra los campos llenos del componente ActivityStarter1.

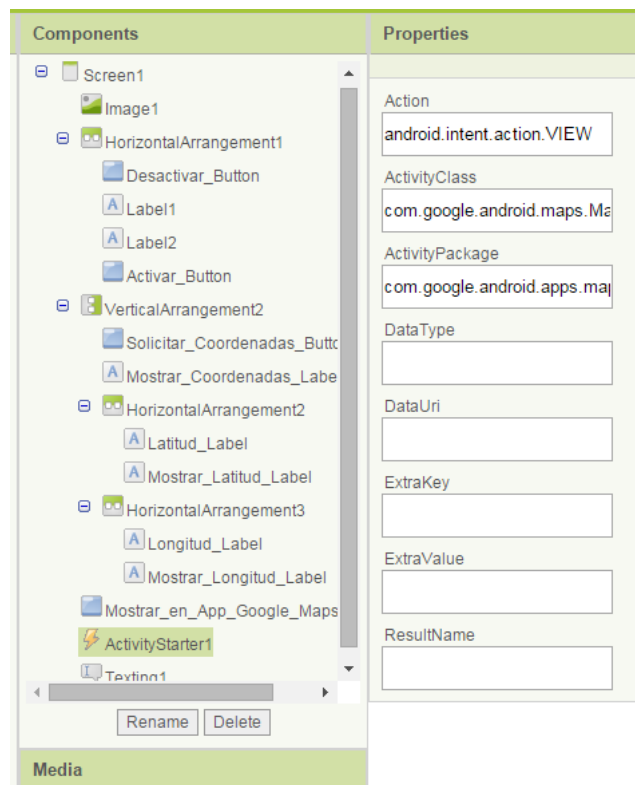


Figura 4.38 Configuración del componente ActivityStarter1

Fuente: El Investigador

- Personalizamos los objetos de una forma gráfica como se muestra en la figura 4.39



Figura 4.39 Pantalla Parcial mostrando los componentes Personalizados.
Fuente: El Investigador.

En la tabla 4.10. Se detallan los componentes utilizados en la aplicación móvil Control Vehicular con los cambios personalizados.

Tabla 4.10. Listado de componentes usados en la aplicación Móvil

Componente	Nombre asignado	Otra personalización	Objetivo
Image1	Image1		Mostrar un entorno de animación a la aplicación.
Button1	Activar	Se elimina el contenido de la propiedad <i>text</i> ..	Enviar el texto al destinatario.
Butoon2	Desactivar	Se elimina el contenido de la propiedad <i>text</i> ..	Enviar el texto al destinatario.
Label1	Espacio.Label1	Se elimina el contenido de la propiedad <i>text</i> .	Separa con un espacio los objetos en la aplicación
Label2	Mostrar_Msj.Label2	La propiedad <i>text</i> se cambia a "Mensaje".	Muestra el mensaje recibido.
Label3	Espacio.Label3	Se elimina el contenido de la propiedad <i>text</i> .	Separa con un espacio los objetos en la aplicación.

HorizontalArrangement3	HorizontalArrangement3		Panel para organizar los elementos de la interface.
Button3	Obtener_Ubicación	La propiedad <i>text</i> se cambia a “Obtener Ubicacion”.	Permite separar los datos Latitud y Longitud.
Label4	Espacio.Label5	Se elimina el contenido de la propiedad <i>text</i> .	Separa con un espacio los objetos en la aplicación.
Label5	Latitud.Label6	La propiedad <i>text</i> se cambia a “Latitud”.	Muestra el dato Latitud.
Label6	Espacio.Label7	Se elimina el contenido de la propiedad <i>text</i> .	Separa con un espacio los objetos en la aplicación.
Label7	Longitud.Label6	La propiedad <i>text</i> se cambia a “Longitud”.	Muestra el dato Longitud
Button4	Ver_en_el_Mapa	La propiedad <i>text</i> se cambia a “Ver_google_Maps”.	Muestra la ubicación del vehiculo en el mapa de google.
ActivityStarter1	ActivityStarter1	<p>Agregamos en los campos:</p> <p>Action:android.intent.action.VIEW,</p> <p>ActivityClass:com.google.android.maps.MapActivity,</p> <p>ActivityPackage:com.google.android.apps.maps</p>	Crea un acceso directo desde nuestra aplicacion hacia la App Google Maps.
Texting1	Texting1		Permite enviar mensajes a un número determinado

Fuente: El Investigador

- Para agregar el comportamiento de nuestra aplicación seleccionamos el botón *Open the Blocks Editor*. Lo cual nos lleva a la descarga y ejecución la última versión disponible del editor de bloques. al ejecutarlo aparecerá una ventana como se muestra en la figura 4.40. Donde se muestran las diferentes secciones

de bloques predefinidos y una sección (My Blocks) donde se encuentran los componentes ahora llamados bloques usados en el paso anterior. Los bloques se arrastran al área principal donde vamos armando nuestros conjuntos de bloques generando la lógica de la aplicación.

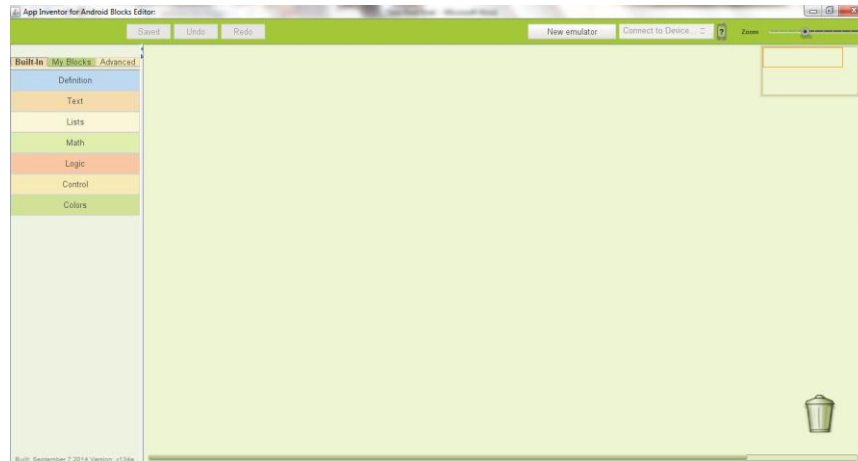


Figura 4.40 Vista del Android blocks Editor

Fuente: El Investigador

9. Configuramos la lógica de los componentes *Activar* y *Desactivar*, en esta parte le indicamos a los componentes los pasos a seguir cuando el usuario realiza la acción *Click* en algún componente.
 - Cuando Presionamos *Activar* la aplicación envía un mensaje de texto con la palabra *Activar* al número de destinatario asignado en el *Ordenador_Vehicular*. Para indicar este comportamiento en nuestra aplicación, seleccionamos el componente *ACTIVAR* de la barra *MyBlocks*. De dicho componente elegimos el bloque *Click*. A continuación definimos el comportamiento específico dentro del evento *Click*. Primero, asignamos el número telefónico a *Numero_texting1.PhoneNumber*. del mismo modo asignamos el mensaje al componente *Numero_Texting1.Message*, luego lo enviamos ejecutando el evento *Numero_Texting1.SendMessage* como se muestra en la figura 4.41.

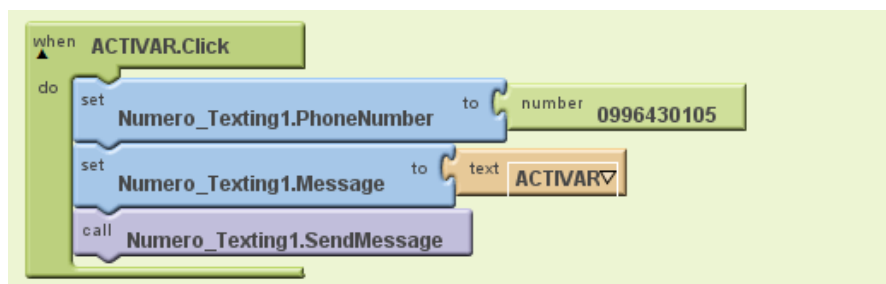


Figura 4.41 Comportamiento del Componente Button ACTIVAR
Fuente: El Investigador

- Cuando Presionamos *Desactivar* la aplicación envía un mensaje de texto con la palabra *Desactivar* al número de destinatario asignado en el Ordenador_Vehicular. Para indicar este comportamiento en nuestra aplicación, seleccionamos el componente *DESACTIVAR* de la barra MyBlocks. De dicho componente elegimos el bloque Click. A continuación definimos el comportamiento específico dentro del evento Click. Primero, asignamos el número telefónico a *Numero_texting1.PhoneNumber*. del mismo modo asignamos el mensaje al componente *Numero_Texting1.Message*, luego lo enviamos ejecutando el evento *Numero_Texting1.SendMessage* como se muestra en la figura 4.42.

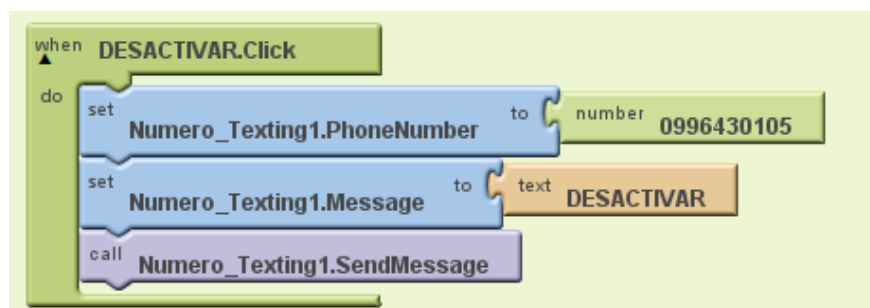


Figura 4.42 Comportamiento del Componente Button DESACTIVAR
Fuente: El Investigador

10. Configuramos la lógica de la aplicación para obtener la ubicación del Vehículo mediante los siguientes pasos.
 - Cuando Presionamos *OBTENER_UBICACION* la aplicación envía un mensaje de texto con la palabra *Ubicación* al número de destinatario asignado en el Tesis_Ordenador_Vehicular. Para indicar este comportamiento en nuestra aplicación, seleccionamos el componente *Obtener_Ubicacion* de la barra MyBlocks. De dicho componente elegimos el bloque Click. A continuación definimos el comportamiento específico

dentro del evento Click. Primero, asignamos el número telefónico a *Numero_texting1.PhoneNumber*. del mismo modo asignamos el mensaje al componente *Numero_Texting1.Message*, luego lo enviamos ejecutando el evento *Numero_Texting1.SendMessage* como se muestra en la figura 4.43.

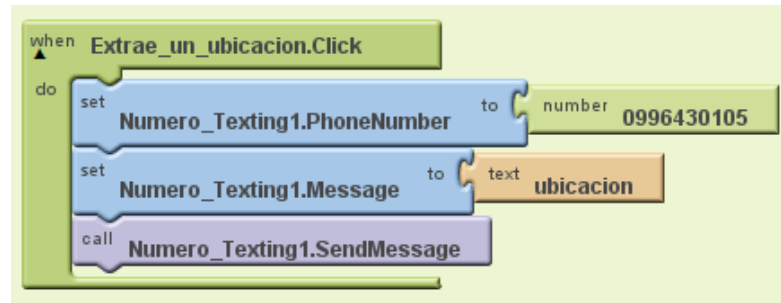


Figura 4.43 Comportamiento del Componente Button Obtener Ubicación
Fuente: El Investigador

- Una vez realizada esta acción el Ordenador Vehicular devolverá un Mensaje con el siguiente formato: Ubicación:Latitud;Longitud, y lo muestra en la pantalla de la aplicación mediante el componente *Mostrar_msj.text* contenido dentro del objeto *Numero_Texting.MessageReceived* como se muestra en la figura 4.44.



Figura 4.44 Comportamiento del Componente MessageReceived
Fuente: El Investigador

- Mediante la función *make text* almacenamos el mensaje recibido contenido en el objeto *MessageText* en la variable llamada *texto* como se muestra en la figura 4.45

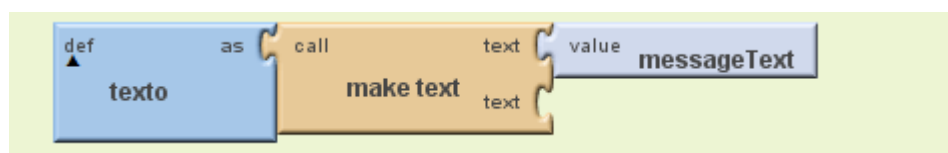


Figura 4.45 Uso de la función *Make text* para almacenar el mensaje recibido en la variable *Text*.

Fuente: El Investigador

- Configuramos la lógica del componente *Extraer_ubicacion*, en esta parte le indicamos al componente los pasos a seguir para verificar si el mensaje recibido corresponde a los datos del GPS y separar los datos de latitud y longitud cuando el usuario realiza la acción *Click*, primero seleccionamos el componente *Extraer_Ubicacion* de la barra MyBlocks. De dicho componente elegimos el bloque *Click*, a continuación definimos el comportamiento específico dentro del evento *Click*. Primero, incluimos un condicional *if* para verificar si la información corresponde a los datos enviados por el GPS, con la función *Segment* extraemos una subcadena de la cadena contenida en la variable *Texto* y comparamos con la palabra clave que viene incluida con los datos del GPS mediante la función *Text=*. Si la condición es correcta separamos los datos de latitud y longitud utilizando la lógica anterior y lo mostramos en los componentes *Latitud.text* y *Longitud.text*

A continuación se muestra la lógica en la figura 4.46.

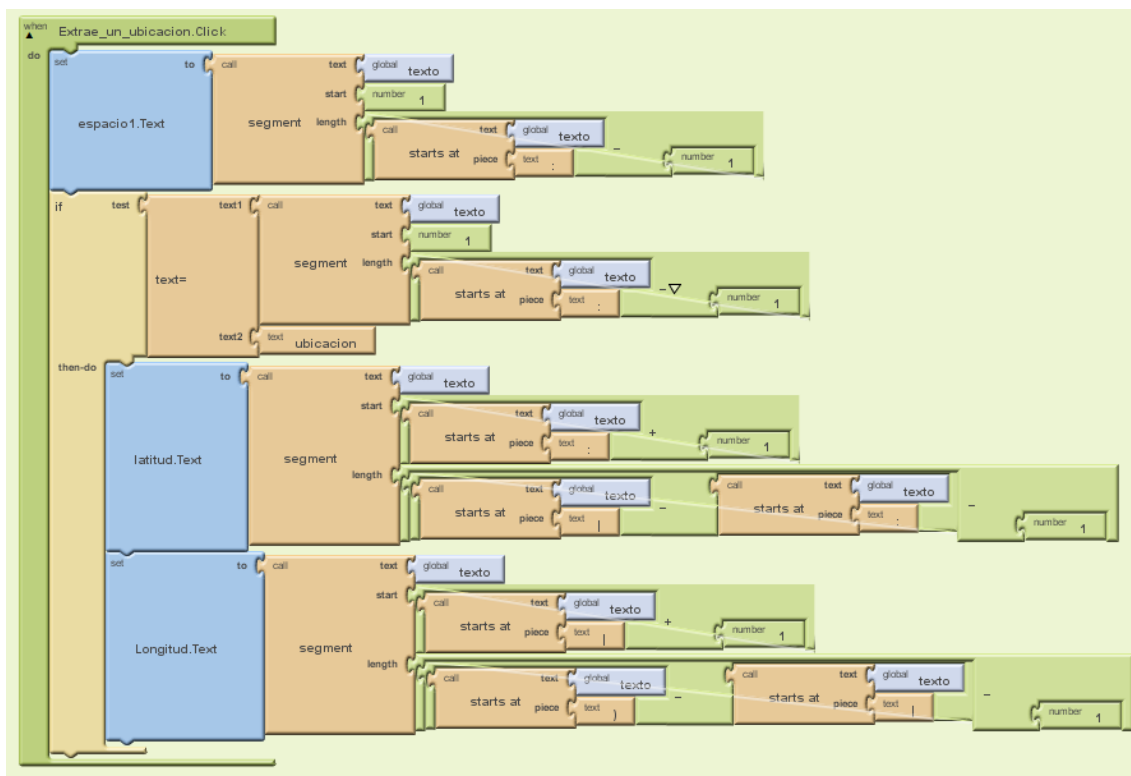


Figura 4.46 Lógica del componente *Extraer_Ubicacion*

Fuente: El Investigador

11. Configuramos la lógica del componente *Buscar_en_google_maps.Click*, en esta parte le indicamos al componente los pasos a seguir para mostrar los datos de latitud y longitud en el mapa de google creando un acceso directo desde la aplicación *Control_Vehicular* hacia la App de *Google Maps* cuando el usuario realiza la acción *Click*, primero seleccionamos el componente *Buscar_en_googe_maps* de la barra MyBlocks. De dicho componente elegimos el bloque *Click*, a continuación definimos el comportamiento específico dentro del evento *Click*. Primero, atravez de función *ActivityStarter1.DataUri* incluimos la dirección web contenida con el dato latitud y longitud provenientes del GPS esto lo conseguimos uniando subcadenas en una cadena atravez de la función *make.text*, luego llamamos a la aplicación atravez de la función *ActivityStarter1.StartActivity* contenida en la barra MyBlocks.

A continuación se muestra la lógica en la figura 4.47.

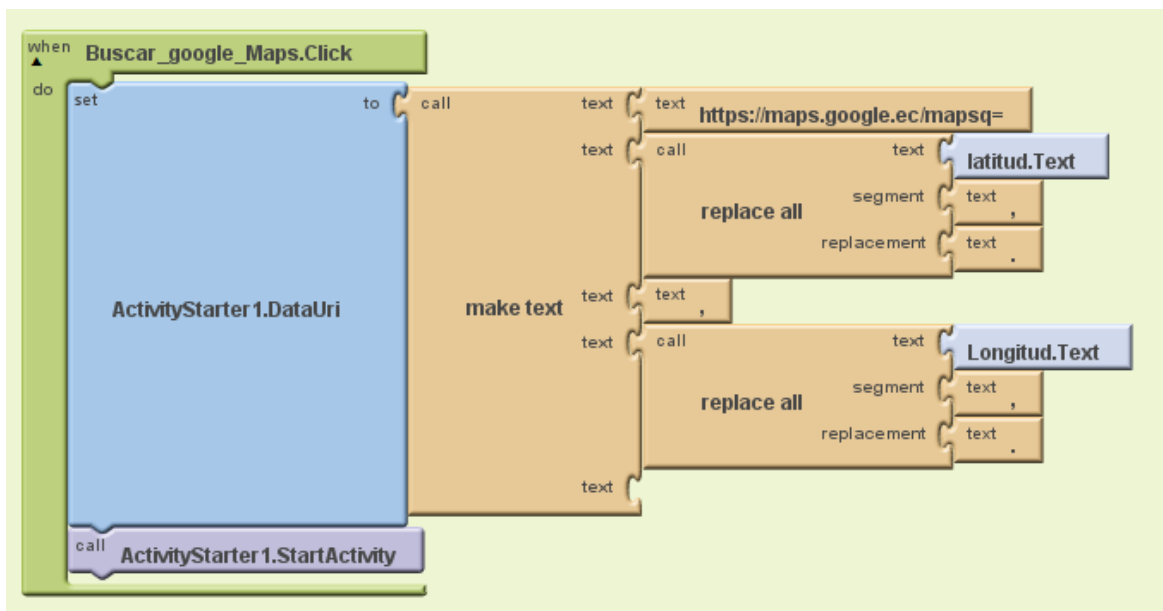


Figura 4.47 Lógica del componente *Buscar_en_Google_Maps.Click*

Fuente: El Investigador

4.11 Instalación del sistema

4.11.1 Bloqueo Central del Vehículo

Localizamos y cortamos el cable del motor de arranque. Un lugar recomendado es entre los cables que salen de la llave como se muestra en la figura 4.48.

El cable del Motor de Arranque muestra 12 voltios positivos solamente cuando el motor de arranque está funcionando.

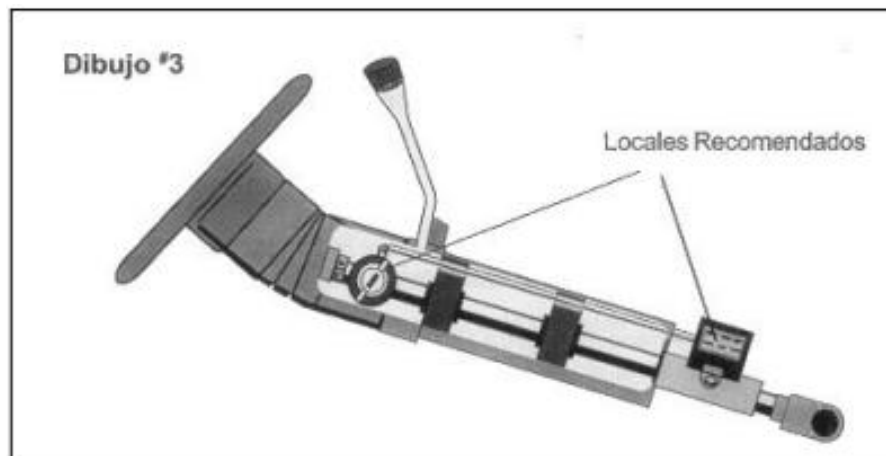


Figura 4.48 Ubicación del bloqueo Central

Fuente: <http://diagramasde.com/diagramas/otros2/Montaje%20alarma%20carro.pdf>

4.11.2 Sensores y actuadores

Los sensores pulsadores se los ubica en la carrocería del vehículo y son activados cuando la puerta se cierra o se abre, el diagrama de conexión se lo muestra en la figura 4.49.

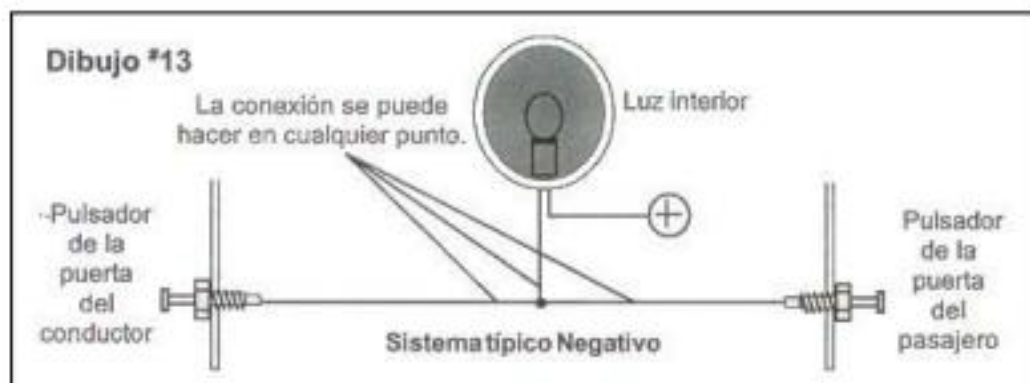


Figura 4.49 Montaje de los pulsadores de puertas

Fuente: <http://diagramasde.com/diagramas/otros2/Montaje%20alarma%20carro.pdf>

La instalación de los cierres centralizados se los ubica dentro de la carrocería de las puertas como se muestra en la figura 4.50.

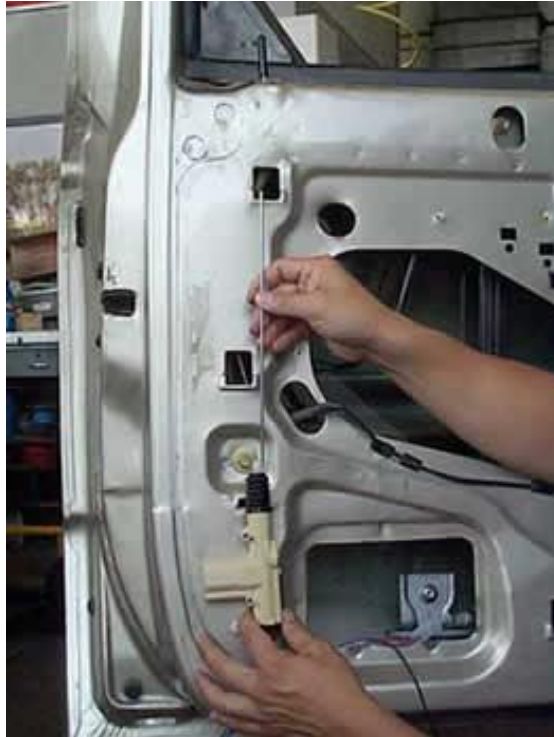


Figura 4.50 Conexión de los Seguros centralizados

Fuente: <http://www.aficionadosalamecanica.net/cierre-centralizado.htm>

4.11.3 Ubicación del Ordenador Electrónico Vehicular con pantalla touch

La ubicación del ordenador vehicular con pantalla touch 7" va a depender del conductor y del vehículo, a continuación en la figura 4.51. Se muestra los posibles lugares de ubicación:

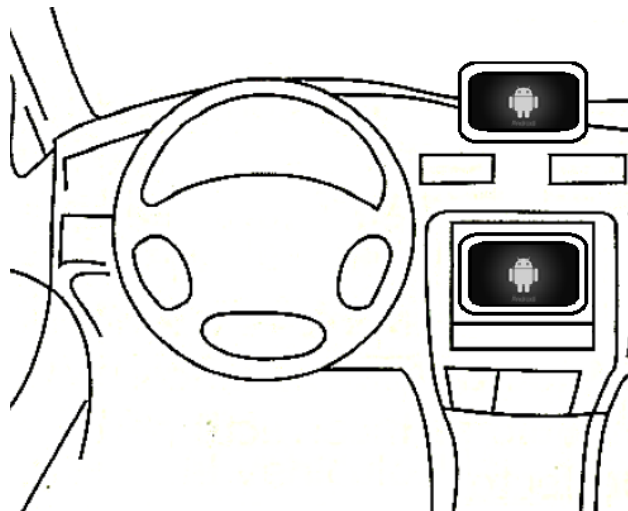


Figura 4.51 Ubicación de la pantalla touch 7"

Fuente: [http://alarmasgenius.com/documentos/Manuales Genius Web/Sensores Genius de Retroceso4.pdf](http://alarmasgenius.com/documentos/Manuales%20Genius%20Web/Sensores%20Genius%20de%20Retroseso4.pdf)

4.11.4 Instalación de la Aplicación Móvil en un sistema operativo Android

Para la instalación de la aplicación móvil seguimos los siguientes pasos.

1. Descargamos el archivo ejecutable .apk desde la pestaña *generar*. *App(guardar archivo .apk en mi ordenador)* como se muestra en la figura 4.52.

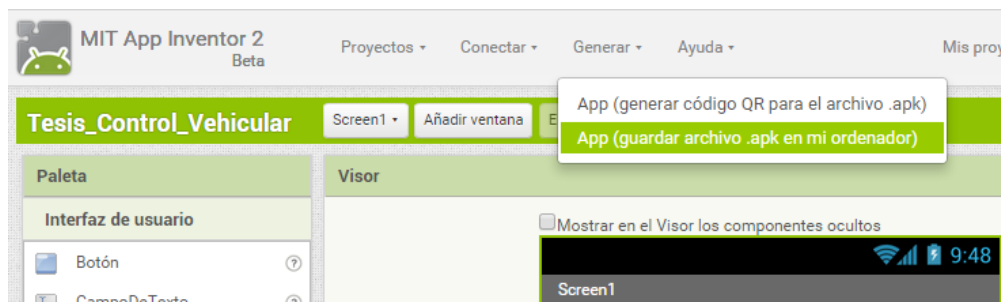


Figura 4.52 Generación del archivo ejecutable .apk de la aplicación Móvil.
Fuente: El Investigador.

2. Copiamos el archivo *Control Vehicular .apk* en nuestro Smartphone como se muestra en la figura 4.53.

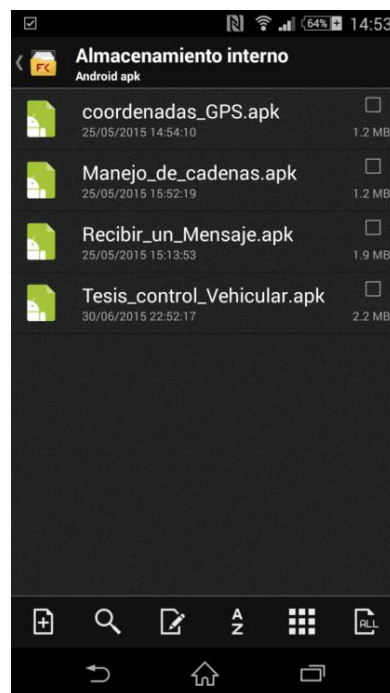


Figura 4.53 Archivo *Tesis_Ordenadorl_Vehicular.apk* copiado en el Smartphone
Fuente: El Investigador.

3. Instalamos la aplicación en nuestro Smartphone y se ejecutara en modo grafico como se muestra en la figura 4.54.



Figura 4.54 Aplicación Tesis_Ordenador_Vehicular en el Smartphone

Fuente: El Investigador

4.12 Costo del proyecto

El costo de los materiales del Ordenador vehicular en un automotor liviano se detalla en la tabla 4.12.

Tabla 4.11. Costo de los materiales para el proyecto

Descripción	Cantidad	Valor Unit.	Valor Total
Servomotores	2	\$22,00	\$44,00
Pulsadores de puertas	4	\$5,00	\$20,00
Placa Raspberry Pi	1	\$ 65,00	\$ 65,00
Placa Arduino Uno	1	\$ 30,00	\$ 30,00
Kit Pantalla de 7 pulgadas	1	\$ 110,00	\$ 110,00
Kit Phone Shield Sim 908 con GPS	1	\$ 120,00	\$ 120,00
Fibra de vidrio para acople del sistema	1	\$ 40,00	\$ 40,00
Elementos electrónicos	1	\$ 48,00	\$ 48,00

Placa de baquelita + cloruro férrico	1	\$10,00	\$10,00
Fusible	1	\$ 0,30	\$ 0,30
Interruptor	1	\$ 0,50	\$ 0,50
Cable para los sensores y actuadores (metros)	10	\$ 0,80	\$ 8,00
Cable HDMI	1	\$ 8,00	\$ 8,00
Cable USB	1	\$ 3,00	\$ 3,00
Conectores	5	\$ 0,40	\$ 2,00
TOTAL			\$ 508,80

Fuente: El Investigador

El costo de instalación del Ordenador Electrónico Vehicular se obtuvo de acuerdo a los precios de los equipos y módulos que forman parte del sistema.

La mano de obra es de \$100 considerando que el tiempo promedio de instalación es 8 horas.

Debido a esto el costo total del Ordenador Electrónico vehicular instalado en un automóvil es: \$ 608,80.

Cabe mencionar que se ha empleado Software y Hardware Libre el cual no tiene ningún costo de ejecución y funcionamiento.

Las pruebas de funcionamiento se detallan en el Anexo F.

4.13 Implementación del ordenador vehicular en un automóvil Chevrolet Forza

En la figura 4.55 Se muestra como quedaron instalados los sensores en las puertas laterales y en los compartimientos delantero y posterior.



Figura 4.55 Instalación de los Pulsadores en las puertas y compartimientos
Fuente: El Investigador

En la figura 4.56. Se muestra la vista frontal del ordenador Electrónico Vehicular, en la cual se visualiza la pantalla.

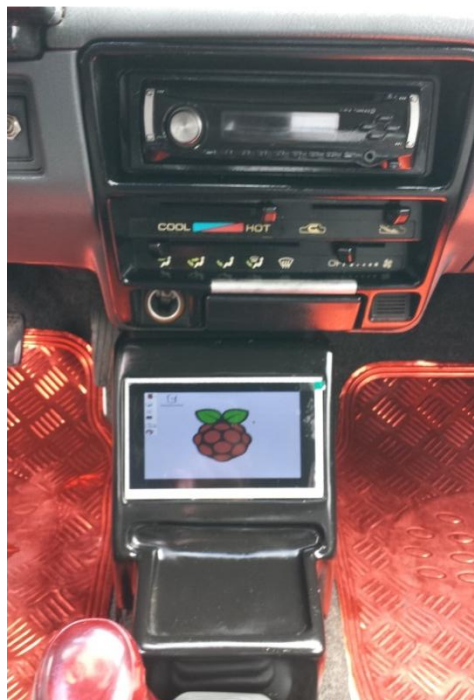


Figura 4.56 Vista Frontal del Ordenador Vehicular
Fuente: El Investigador

En la figura 4.57. Se muestra la vista posterior del ordenador electrónico vehicular en la cual se ubicaron los equipos que forman parte del sistema.



Figura 4.57 Vista posterior del Ordenador Electrónico Vehicular
Fuente: El Investigador

El proyecto de investigación denominado Ordenador Electrónico Vehicular, con Sistema de Seguridad y GPS utilizando Raspberry Pi y Hardware Libre, supone un cambio importante en la forma de elevar un nivel de seguridad, monitoreo y control de un vehículo de manera personal, el proyecto contempla un diseño acoplado a la estética del vehículo y la tecnología que hoy en día las personas tenemos acceso.

CAPÍTULO IV

Conclusiones y Recomendaciones

Conclusiones

- Se elaboró un ordenador electrónico vehicular a través de un sistema de hardware y software libre para controlar el sistema de alarmas contra atracos beneficiando a los propietarios.
- Se aplicó comando AT de la tecnología inalámbrica para establecer la comunicación del usuario hacia el ordenador consiguiendo su manejo a cualquier distancia dentro de la red GSM.
- Se observó mediante un análisis profundo como están conformados los sistemas de alarmas de empresas integradas a la seguridad vehicular logrando a través de los diferentes tipos de encendidos un adecuado bloqueo.
- Se desarrolló un método operativo ligado a una aplicación Android para el control y rastreo del vehículo acoplando su uso a la era tecnológica actual.

Recomendaciones

- Ocultar los equipos que forman parte del ordenador Electrónico Vehicular, de manera que este se de difícil acceso, sin embargo la LDC touch debe ser colocada en un lugar visible al conductor salvaguardando la estética del vehículo.
- El sistema se encuentra limitado a la área de cobertura de la Red GSM por lo que es recomendable realizar un análisis sobre la zona donde funcionara el sistema para seleccionar la operadora adecuada.
- El Ordenador electrónico Vehicular entrega las coordenadas GPS de la ubicación del Vehículo hacia la aplicación Móvil, debido a esto el dispositivo debe mantener una conexión a internet para que la aplicación funcione correctamente.
- El Ordenador Electrónico Vehicular utiliza la misma energía de la batería del automóvil, es recomendable instalar una batería de 12 V la cual se encargue de proveer energía al sistema en caso de que la batería principal sea desconectada.

Bibliografía y referencias

- [1] L. E. Cando Tite, “Bloqueo Electrónico en el Encendido del Vehículo”, Tesis Ingeniero Electrónico y Comunicaciones, Facultad de Ingeniería en Sistemas Electrónica y Comunicaciones, Universidad Técnica de Ambato, Ambato, Ecuador, 2011.
- [2] M. M. Freddy, Ch. Q. Richard, “Diseño e Implementación de un prototipo de un Sistema de monitoreo y bloqueo del motores de vehículos vía mensajes de texto usando las redes de telefonía celular GSM existentes en el país”, Tesis de Electrónica y Telecomunicaciones, Facultad de Electrónica y Telecomunicaciones, Escuela Politécnica Nacional, Quito, Ecuador, 2014.
- [3] G. C. Quevedo, “Diseño e Implementación de un Sistema de localización, Rastreo y monitoreo satelital de camiones de entrega de encomiendas; Mediante el Uso de GPS y un dispositivo móvil”, Tesis de Ingeniería en Telecomunicaciones, Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogota, Colombia, 2010.
- [4] J. M. Alonso, “Técnicas del automóvil”, Equipo Eléctrico, Séptima Edición, Editorial Paraninfo, España, 1998.
- [5] Batería y Acumuladores, Tecnología, [Online]. disponible en: <http://www.area-tecnologia.com/baterias-y-acumuladores.htm>
- [6] Tecnoficio, Electricidad el automóvil. [Online]. Disponible en: http://www.tecnoficio.com/electricidad/electricidad_del_automotor5.php
- [7] Motor Giga, Motor de Arranque, [Online]. Disponible en: <http://diccionario.motor-giga.com/diccionario/motor-de-arranque-definicion-significado/gmx-niv15-con194883.htm>
- [8] Wikipedia, Encendido electrónico, Septiembre 2015, [Online]. Disponible en: http://es.wikipedia.org/wiki/Encendido_electr%C3%B3nico
- [9] Unidad Didáctica, El circuito eléctrico. [Online]. Disponible en: http://www.quimicaweb.net/grupo_trabajo_fyq3/tema8/index8.htm
- [10] Técnicas del automóvil – equipo eléctrico J. M. Alonso, Sétima Edición (1998), (2008), Editorial Paraninfo, Madrid. España.
- [11] Sistema de Posicionamiento Global, Características técnicas y prestaciones, [Online], Disponible en: <http://samubebe88.blogspot.com/2010/08/gps.html>
- [12] J. L. SAPIA, Manual técnico Inmovilizadores, Instituto tecnológico Superior del Automotor, Centro de capacitación Automotriz, Santa fe, Argentina, 2002.

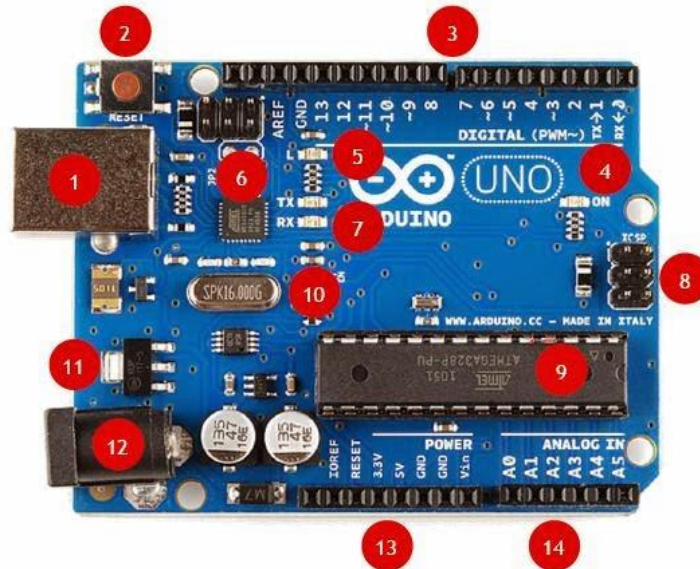
- [13] Sistema de Seguridad, Mecánica y automoción, Marzo 2009. [Online]. Disponible en: <http://me-canicayautomocion.blogspot.com/2009/03/sistemas-de-seguridad-en-el-automovil.html>
- [14] Sistemas de alarmas vehiculares, Alarmas Tausend, [Online]. Disponible en: <http://www.alarmatausend.com/AlarmasParaAuto2.html>
- [15] Nueva App Chevystar, Pág oficial Chevrolet, [Online]. Disponible en: <http://www.chevrolet.com.ve/chevystar/app.html>.
- [16] GPS trackers Ecuador, Allbiz. [Online]. Disponible en: <http://www.ec.all.biz/gps-trackers-bgg1087053>
- [17] Hunter SOS pág. Oficial, Soluciones Tecnológicas en Seguridad, Tecnología, 2011, [Online]. Disponible en: <http://www.hunter.com.ec/tecnologia.aspx>
- [18] Software libre, Wikipedia, Diciembre 2004. [Online], Disponible en: http://es.wikipedia.org/wiki/Software_libre
- [19] Todo sobre GNU y Linux, Ciberaula, 2014, [Online]. Disponible en: http://linux.ciberaula.com/articulo/ventajas_inconvenientes_linux/
- [20] A. Álvaro, Manual de Linux. [Online], Disponible en: <http://www.asturlinux.org.PDF>.
- [21] Hardware Libre, EcuRed, Octubre 2015, [Online]. Disponible en: http://www.ecured.cu/index.php/Hardware_libre
- [22] Pantalla Táctil. Wikipedia, Diciembre 2011, [Online], Disponible en: http://es.wikipedia.org/wiki/Pantalla_t%C3%A1ctil
- [23] "TFT LCD, Wikipedia, Septiembre 2015, Disponible en: " http://es.wikipedia.org/wiki/TFT_LCD
- [24] 3.5 " TFT LCD de pantalla táctil Shield Kit + estuche + disipadores para Raspberry Pi. AliExpress, [Online], Disponible en: <http://es.aliexpress.com/item/3-5-TFT-LCD-Shield-Touch-Screen-Kit-Display-Case-Heatsinks-for-Raspberry-Pi/32325005734.html>
- [25] 7 " LCD Pi frambuesa pantalla táctil TFT del Monitor AT070TN90 con pantalla táctil Kit HDMI de entrada VGA del tablero de conductor, AliExpress, [Online]. Disponible en: <http://es.aliexpress.com/item/7-Raspberry-Pi-LCD-Touch-Screen-Display-TFT-Monitor-AT070TN90-with-Touchscreen-Kit-HDMI-VGA-Input/1782782511.html>
- [26] "Digital pantalla del IPS (1280 x 800) + tablero de la impulsión para Pi / Pduino / Cubieboard - negro # N101ICG-L21, AliExpress, [Online]. Disponible en: <http://es.aliexpress.com/item/10-1-Digital-IPS-Screen-1280-x-800-Drive-Board-for-Raspberry-Pi-Pduino-Cubieboard-Black/1782794578.html>

- [27] UDOO. [Online], Disponible en: <http://www.udoo.org/docs/>
- [28] Raspberry Pi, Wikipedia,. Disponible en: http://es.wikipedia.org/wi-ki/Raspberry_Pi#Hardware
- [29] Raspberry Pi Modelo A+, Wikipedia. Disponible en: <https://www.raspberrypi.org/products/model-a-plus/>
- [30] Raspberry Pi Modelo B+, Wikipedia, Disponible en: <https://www.raspberrypi.org/products/model-b-plus/>
- [31] Raspberry Pi 2 Modelo B+, Wikipedia, [Online]. Disponible en: <https://www.raspberrypi.org/products/raspberrypi-2-model-b/>
- [32] Que es Arduino, Arduino. 2012, [Online], Disponible en: <https://www.arduino.cc/es/pmwiki.php?n=>
- [33] Series de Arduino, [Online]. Disponible en: <http://www.electronicaembajadores.com/Admin/Content/eyontzqw.pdf>
- [34] C. D. Alberto. Sistema de control de temperatura a través de Arduino y la tecnología GPRS/GSM, E.U.I.T. Telecomunicación, Comunicaciones Móviles, 2012.
- [35] Phone shield SIM 908, [Online], Disponible en: <http://www.wvshare.com/product/Phone-Shield-SIM908.htm>
- [36] Comunicación Inalámbrica, Wikipedia, Disponible en: http://es.wikipedia.org/wiki/Comunicaci%C3%B3n_inal%C3%A1mbrica
- [37] Que es Android, Xalaka Android, Febrero 2011, Disponible en: <http://www.xatak-android.com/sistema-operativo/que-es-android>
- [38] Que es app Inventor, TuAppinventor. Disponible en: <http://www.tuappinventorandroid.com/aprender/>

ANEXOS

ANEXO A.- Características técnicas De Arduino Uno

Vamos a conocer cómo es el hardware de la placa de ARDUINO UNO con la que vamos a trabajar en estas prácticas:



1. Cable USB para conectar a PC.
2. Botón de RESET.
3. Pines de Entrada/Salida digital y PWM.
4. Mini LED verde de ON (placa alimentada o encendida).
5. Mini LED naranja conectado al PIN 13.
6. ATmega 16U2, responsable de las comunicaciones con el PC.
7. LED TX (transmisor) y LED RX (receptor) de la comunicación serial.
8. Puerto ICSP para programación serial.
9. Microcontrolador ATmega 328 (el cerebro de ARDUINO)
10. Cristal de cuarzo de 16 MHz.
11. Regulador de voltaje.
12. Conector para alimentación.

13. Pines de voltaje y tierra.

14. Entradas analógicas.

ESPECIFICACIONES TÉCNICAS

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current for I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Vamos a explicar con un mayor detalle los tres bloques de pines más importantes de la placa:

13. Pines de alimentación



Bien alimentemos al Arduino mediante la conexión USB o mediante una fuente externa (recomendada de 7-12V), vamos a tener unas salidas de tensión continua debido a unos reguladores de tensión y condensadores de estabilización.

Estos pines son:

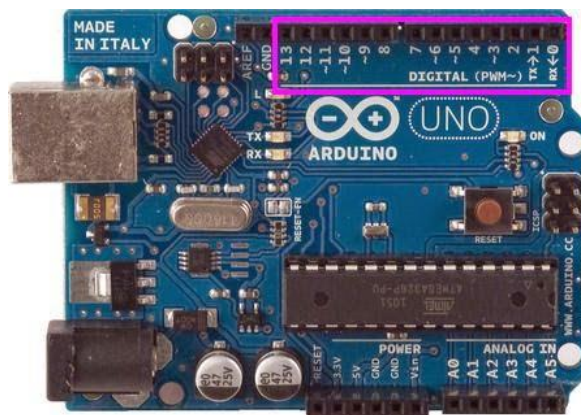
VIN: se trata de la fuente tensión de entrada que contendrá la tensión a la que estamos alimentando al Arduino mediante la fuente externa.

5V: fuente de tensión regulada de 5V, esta tensión puede venir ya sea de pin VIN a través de un regulador interno, o se suministra a través de USB o de otra fuente de 5V regulada.

3.3V: fuente de 3.3 voltios generados por el regulador interno con un consumo máximo de corriente de 50mA.

GND: pines de tierra.

3. Entradas/salidas digitales



Cada uno de los 14 pines digitales se puede utilizar como una entrada o salida. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas como:

Pin 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y la transmisión (TX) de datos serie TTL.

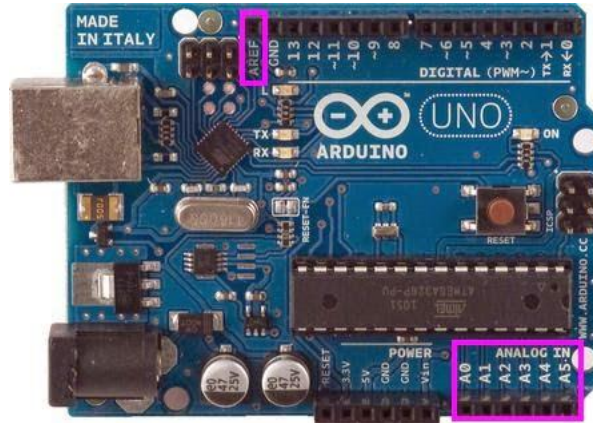
Pin 2 y 3. Interrupciones externas. Se trata de pines encargados de interrumpir el programa secuencial establecido por el usuario.

Pin 3, 5, 6, 9, 10 y 11. PWM (modulación por ancho de pulso). Constituyen 8 bits de salida PWM con la función `analogWrite()`.

Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines son de apoyo a la comunicación SPI.

Pin 13. LED. Hay un LED conectado al pin digital 13. Cuando el pin es de alto valor, el LED está encendido, cuando el valor está bajo, es apagado.

14. Entradas analógicas.



El Arduino posee 6 entradas analógicas, etiquetadas desde la A0 a A5, cada una de las cuales ofrecen 10 bits de resolución (es decir, 1024 estados). Por defecto, tenemos una tensión de 5V, pero podemos cambiar este rango utilizando el pin de AREF y utilizando la función `analogReference()`, donde le introducimos una señal externa de continua que la utilizara como referencia.

ANEXO B.- Características técnicas de Raspberry Pi Model B+



Raspberry Pi



MODEL B+

Product Name	Raspberry Pi Model B+
Product Description	The Raspberry Pi Model B+ incorporates a number of enhancements and new features. Improved power consumption, increased connectivity and greater IO are among the improvements to this powerful, small and lightweight ARM based computer.

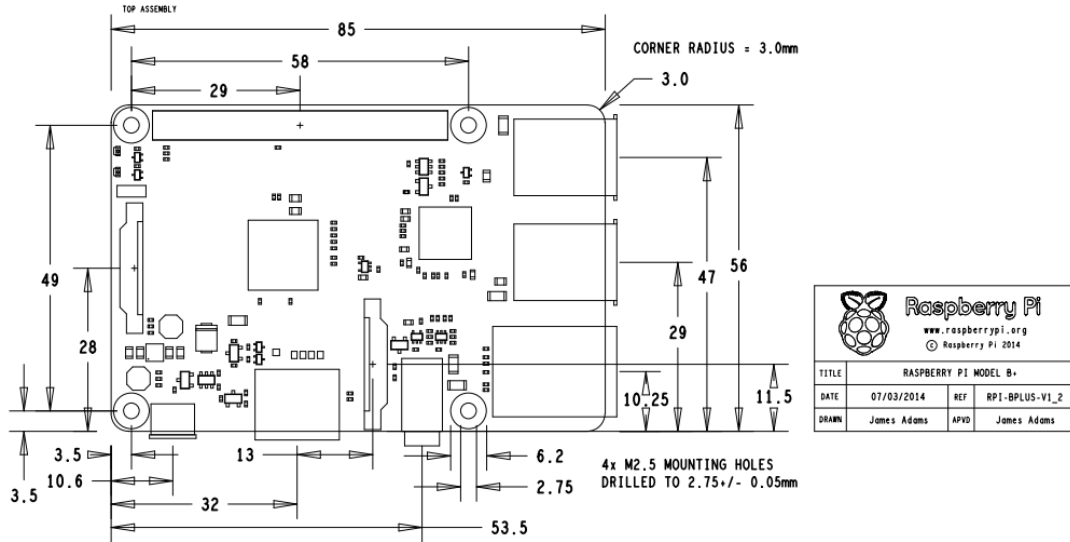
Specifications

Chip	Broadcom BCM2835 SoC
Core architecture	ARM11
CPU	700 MHz Low Power ARM1176JZFS Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	512MB SDRAM
Operating System	Boots from Micro SD card, running a version of the Linux operating system
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V, 2A

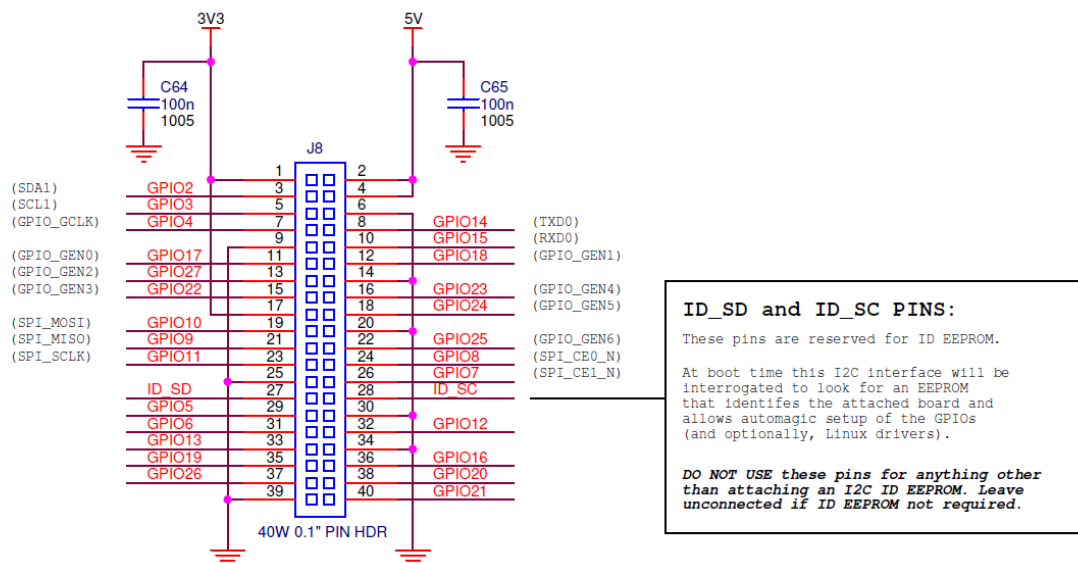
Connectors:

Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	3.5mm jack, HDMI
USB	4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
JTAG	Not populated
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card Slot	SDIO

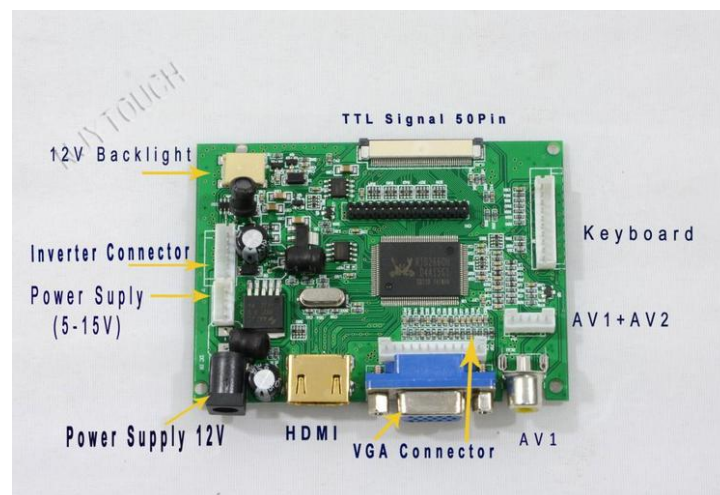
Dimensiones



Circuiteria



ANEXO C.- Características técnicas de TONTEC 7”



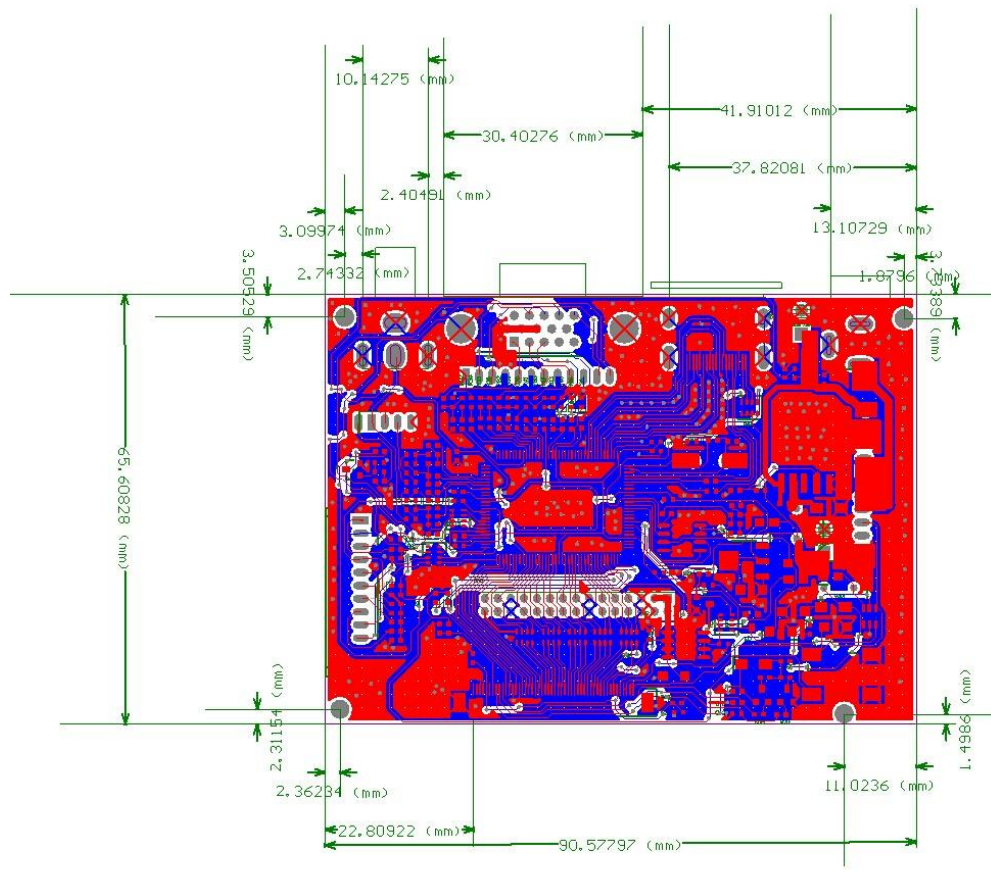
1. Product Features

Supports the following:

- 1, 1 * VGA input
- 2, 2* AV input
- 3, 1* HDMI input , and IC supports HDMI1.1
- 4, 1* Reversing signal input

- 5, Wide voltage input, and normal operation between 5V-24V
- 6, Standard backlight(6PIN interface); external inverter connected
- 7, Integrated LCD LED backlight driver board
- 8, Standard LVDS signal output, supports 1 ch 6 bit; 1 ch 8 bit; 2 ch 6 bit; 2 ch 8 bit LVDS signal LCD panels, but only for LCD supplied by 3.3V power.
- 9, Standard keypad interface, and support double color LED Indicator

PCB Dimension:



ANEXO D.-Características del Phone Shield Sim908 GSM/GPRS/GPS

General

- Conectividad Arduino, se conecta a la Junta Arduino directamente
- NOKIA altavoz y micrófono, apoya manos libres de teléfono charla
- A bordo CP2102 para depurar fácilmente a través del USB interferir
- Onboard 74HC125, por tiempo compartido de control de GSM y GPS puerto a través del puerto de serie único
- Inserción de detección auricular y llame a la vibración
- Quad-banda de 850/900/1800 / 1900MHz
- GPRS clase 10 multi-slot
- GPRS clase estación móvil B
- Cumple con GSM fase 2/2 +
 - Clase 4 (2 W @ 850/900 MHz)
 - Clase 1 (1 W @ 1800/1900 MHz)
- Control a través de comandos AT (GSM 07.07, 07.05 y SIMCom mejorado Comandos AT)
- SIM Application Toolkit
- Suministro rango de tensión:
 - GPRS: 3.2 ~ 4.8 V
 - GPS: 3.0 ~ 4.5V
- Temperatura de funcionamiento: -40 ° C a 85 ° C

SMS

- Punto a punto de MO y MT
- Difusión celular SMS
- Modo Texto y PDU

Audio

- Tricodec
 - Tasa media (HR)
 - Tasa completa (FR)

- Tasa completa mejorada (EFR)
- Manos libres
- Cancelación del eco

GPRS

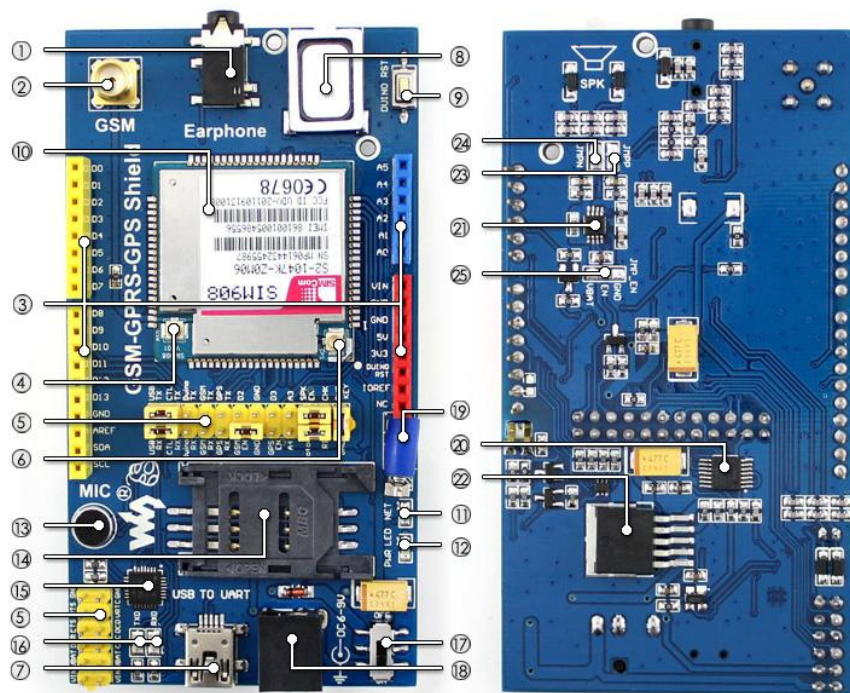
- GPRS clase 8/10: max. 85,6 kbps (enlace descendente)
- Apoyo PBCCH
- Codificación esquemas de CS 1, 2, 3, 4
- PPP-pila

CSD

- CDS hasta 14.4 kbps
- USSD
- El modo no transparente

GPS

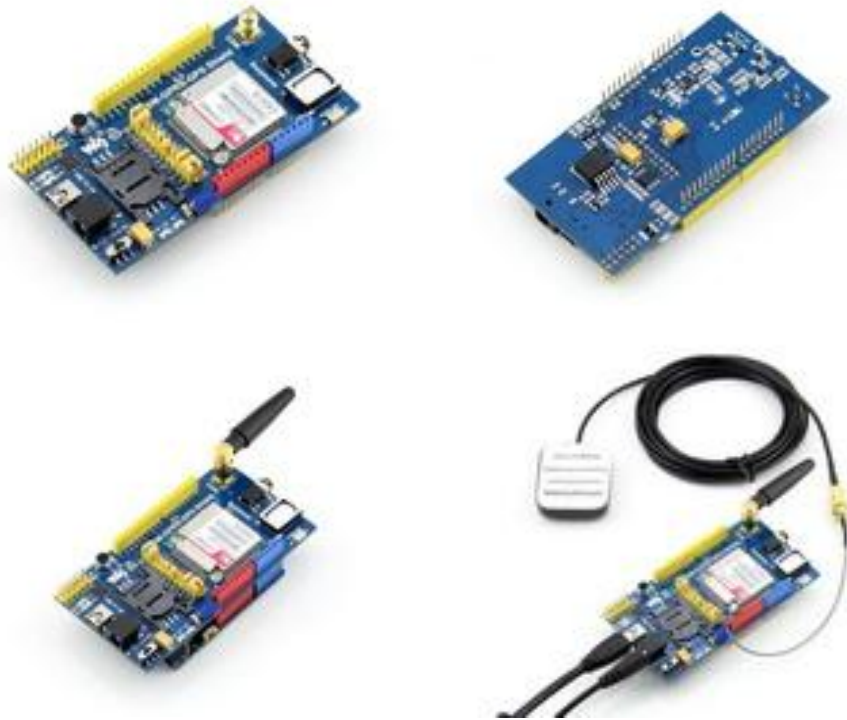
- Tipo Receptor
 - 42 canales
 - GPS L1 código C / A
 - STE motor de alto rendimiento
- Sensibilidad
 - Seguimiento: -160 dBm
 - Arranques en frío: -143 dBm
- Time-To-First-Fix
 - Arranques en frío: 30s (typ.)
 - Arranques en caliente: 1s (typ.)
- Exactitud
 - Posición horizontal: <2,5 m CEP
- Consumo de energía (motor de GSM en el modo inactivo)
 - Adquisición: 77mA
 - Seguimiento: 76mA



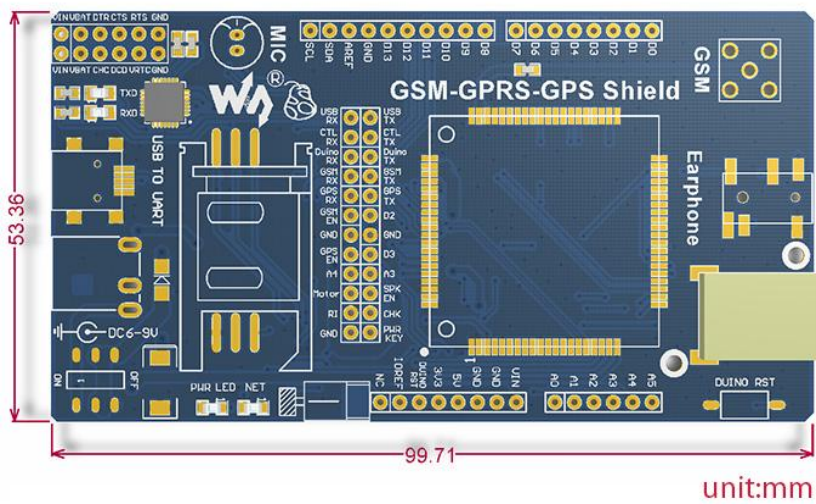
1. Toma de auriculares: SIM908 segunda salida del canal de voz analógica
2. Conector de antena GSM: Interfaz de SMA
3. Conector de expansión de Arduino: para conectar con una placa Arduino
4. Conector de antena GSM
5. Interfaz de control SIM908
6. Conector de antena GPS
7. USB TO interfaz UART
8. NOKIA altavoces originales: SIM908 primera salida de los canales de voz analógica
9. Botón de reinicio Arduino
10. SIM908
11. Indicador de red SIM908: parpadea lentamente cuando el registro de la red es un éxito
12. Indicador de encendido
13. Mic: SIM908 primera entrada del canal de voz analógica
14. Ranura para tarjetas SIM
15. CP2102

- 16.** UART Tx indicador / Rx
- 17.** Interruptor de alimentación
- 18.** Conector de alimentación de 6 V ~ 9V DC
- 19.** Motor de vibración
- 20.** 74HC125
- 21.** Amplificador de energía NCP2890: amplifica el canal de voz SIM908 primera analógico
- 22.** Chip de potencia MIC29302
- 23.** SIM908 salida analógica puente positivo:
 - abrirla cuando NCP2890 está habilitado
 - corta cuando NCP2890 se desactiva
- 24.** SIM908 salida analógica puente negativo:
 - abrirla cuando NCP2890 está habilitado
 - corta cuando NCP2890 se desactiva
- 25.** NCP2890 habilitar puente:
 - corta el EN y VBAT para permitir
 - corta la ES y GND para desactivar

Fotos

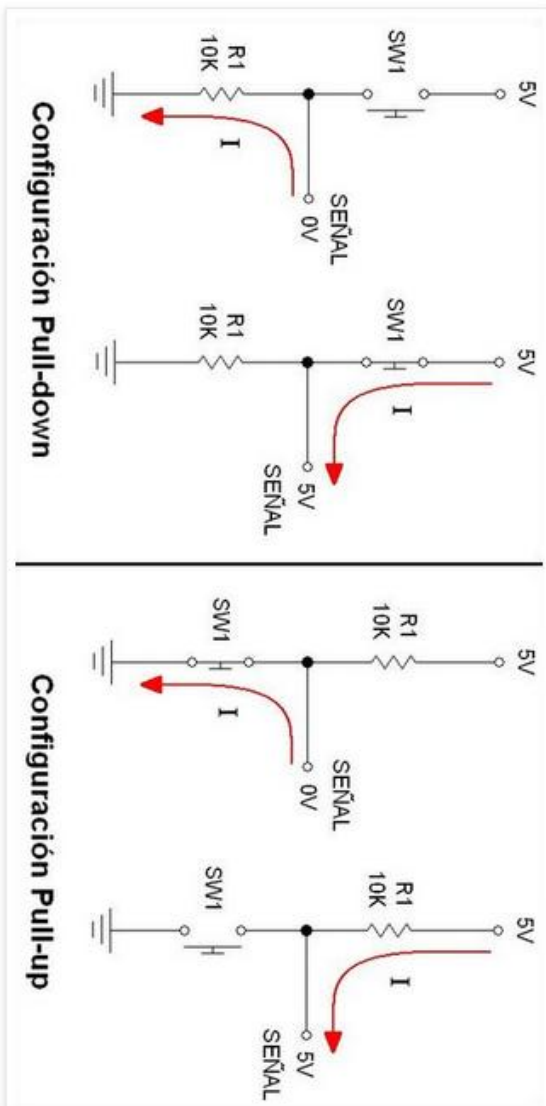


Dimensiones



ANEXO E.- Características técnicas de resistencia Pull-Up y Pull-Down

Son resistencias normales, solo llevan el nombre pull up por la función que cumplen. Las resistencias pull-up puede ser cualquier resistencia. Lo importante es que está conectada entre la señal y la fuente de alimentación lógica. Los rangos comunes de resistencias de pull-ups pueden ser de 1k ohm a 100k ohms, pero depende de la familia lógica de tu microcontrolador



Se podría decir que normalmente se llaman resistencias pull up a resistencias que en general se conectan entre una señal lógica y el positivo y su función es asegurar que esa señal no quede en un estado "flotante".

ANEXO F.- Código de programación para manejo de datos GPS a través de comandos AT.

```
#include "SIM900.h"
#include <SoftwareSerial.h>
#include "sms.h"
#include "gps.h"
#include <stdlib.h>

SMSGSM sms;
GPSGSM gps;

char charLat[10];
char charLon[10];

char lon[15];
float lon_db;
char lat[15];
float lat_db;
char alt[10];
char time[15];
char vel[10];
char msg1[5];
char msg2[5];
char number[]="+593995900360";
char message[180];
char message1[40];
char pos;
char *p;
char stat;
boolean started=false;
void setup()
{
  //Serial connection.
  Serial.begin(9600);
  Serial.println("GSM Shield testing.");
  //Start configuration of shield with baudrate.
  //For http uses is raccomanded to use 4800 or slower.
  if (gsm.begin(2400)){
    Serial.println("\nstatus=READY");
    gsm.forceON();    //To ensure that SIM908 is not only in charge mode
    started=true;
  }
  else Serial.println("\nstatus=IDLE");

  if(started){
```

```

//GPS attach
if (gps.attachGPS())
  Serial.println("status=GPSREADY");
else Serial.println("status=ERROR");

delay(20000);    //Time for fixing
stat=gps.getStat();
  if(stat==1)
    Serial.println("NOT FIXED");
  else if(stat==0)
    Serial.println("GPS OFF");
  else if(stat==2)
    Serial.println("2D FIXED");
  else if(stat==3)
    Serial.println("3D FIXED");
  delay(5000);
  //Get data from GPS
  gps.getPar(lon,lat,alt,time,vel);
  Serial.println(lon);
  Serial.println(lat);
  Serial.println(alt);
  Serial.println(time);
  Serial.println(vel);
}
};
void loop()
{
  pos=sms.IsSMSPresent(SMS_UNREAD);
  Serial.println((int)pos);
  if((int)pos>0&&(int)pos<=20){
    Serial.print("Nuevo Mensaje, POS=");
    Serial.println((int)pos);
    message[0]='\0';
    sms.GetSMS((int)pos,number,message,180);
    p=strstr(message,"gps");
    if(p){
      Serial.println("SMS de GPS OK");
    //Get data from GPS
    gps.getPar(lon,lat,alt,time,vel);
    lon_db=convert(lon);
    dtostrf(lon_db, 4, 4, charLon); //4 is minimum width, 4 is precision; float value is
copied onto buff
    lat_db=convert(lat);
    dtostrf(lat_db, 4, 4, charLat); //4 is minimum width, 4 is precision; float value is
copied onto buff
    Serial.println("dd.dddddd data:");
    Serial.print("LON: ");
    Serial.println(lon_db,4);

```

```

    Serial.print("LAT: ");
    Serial.println(lat_db,4);
    message1[0]='\0';
    strcat(message1,"Ubicacion:");
    strcat(message1,charLat);
    strcat(message1,"");
    strcat(message1,charLon);
    if (sms.SendSMS(number, message1))
        Serial.println("\nSMS sent OK");
    }
    sms.DeleteSMS((int)pos);
}
delay(5000);
};
int strpos(char *str, char *target) {
    char *res=0;
    res = strstr(str, target);
    if (res == NULL) return false;
    else return res - str;
}
float convert(char* str){
    float mmmmmm;
    float dd;
    int pos;
    pos=strpos(str,".");
    char dd_str[pos-1];
    dd_str[0]='\0';
    char mmmmmm_str[6];
    mmmmmm_str[0]='\0';
    for (int i=0; i<pos-2; i++){
        dd_str[i]=str[i];
    }
    dd_str[pos-2]='\0';
    dd=atof(dd_str);
    mmmmmm_str[0]=str[pos-2];
    mmmmmm_str[1]=str[pos-1];
    mmmmmm_str[2]=str[pos];
    mmmmmm_str[3]=str[pos+1];
    mmmmmm_str[4]=str[pos+2];
    mmmmmm_str[5]=str[pos+3];
    mmmmmm_str[6]='\0';
    mmmmmm=atof(mmmmmm_str);
    float result;

    result=dd+mmmmm/1000/60;
    return result;
}

```

ANEXO G.- Código de programación para control del sistema de bloqueo a través de comandos AT.

```
int puertas = 4;
int capot = 5;
int sensONOFFvehiculo = 6;
int blokcentral = 8;
int monitortouch = 9;
int blokdoorON = 10;
int blokdoorOFF = 11;

int pin;
int pin2;
int INdatoCelular;
int INProcessing;
int sp;
int sc;
int sv;
int aux1;
int aux2;
int aux3;
int aux4;

int aux5;
int aux6;
int aux7;
int aux8;
void setup()
{
  pinMode(puertas,INPUT); //define el puerto como entrada
  pinMode(capot,INPUT); //define el puerto como entrada
  pinMode(sensONOFFvehiculo,INPUT); //define el puerto como entrada
  pinMode(blokcentral,OUTPUT); //define el puerto como salida
  pinMode(blokdoorON,OUTPUT); //define el puerto como salida
  pinMode(blokdoorOFF,OUTPUT); //define el puerto como salida
  pinMode(monitortouch,OUTPUT); //define el puerto como salida
  Serial.begin(9600); // define la velocidad de transferencia a 9600 bps (baudios)
  aux5 = 1;
}
void loop()
{
  sp = digitalRead(puertas);
  sc = digitalRead(capot);
  sv = digitalRead(vibracion);
  pin2 = digitalRead(sensONOFFvehiculo);
  if(pin2 == 0)
```

```

{
  digitalWrite(blokcentral,HIGH);
  digitalWrite(bloklinefuel,HIGH);
}

if (Serial.available()>0)//Si el Arduino recibe datos a través del puerto serie
{
  INProcessing = Serial.read(); //Los almacena en la variable INProcessing
  if(INProcessing == 113)
  {
    digitalWrite(blokcentral,LOW);
    digitalWrite(bloklinefuel,LOW);
  }
}
//mensaje recibido se almacena en INdatoCelular
{
  if (pin2 == LOW) //si el vehiculo esta apagado
  {
    if (INdatoCelular == 105 && aux1 == 0)//Condicion para activar el Sistema de
Seguridad
    {
      digitalWrite(blokcentral,HIGH); //Bloquea el sistema centralizado
      digitalWrite(bloklinefuel,HIGH);//Bloquea la linea de combustible
    }
    if (INdatoCelular == 105 && aux1 == 0)
    {
      digitalWrite(monitortouch,HIGH); //Apaga el monitor touch
      digitalWrite(blokdoorON,HIGH); //Activa el seguro de Puertas
      delay(500);
      digitalWrite(monitortouch,LOW);
      digitalWrite(blokdoorON,LOW);
      aux1 = 1;
      aux2 = aux3 = aux4 = aux1;
    }
  }
  else
  {
    if (INdatoCelular == 105)
    //enviar por mensaje ¡Vehiculo en movimiento no se puede activar el sistema¡
    Serial.println("¡No se puede activar el sistema, vehiculo esta en movimiento!");
  }
  if (pin2 == LOW)
  {
    if (INdatoCelular == 111 && aux1 == 1) //Condicion para desbloquear el Sistema
de Seguridad
    {
      //digitalWrite(blokcentral,LOW); //Desbloquea el sistema centralizado
      //digitalWrite(bloklinefuel,LOW); //Desbloquea la linea de combustible
    }
  }
}

```

```

digitalWrite(monitortouch,HIGH); // enciende la pantalla Touch
digitalWrite(blokdoorOFF,HIGH); //Desbloquea el Seguro de las Puertas
delay(500);
digitalWrite(blokdoorOFF,LOW);
digitalWrite(monitortouch,LOW);
aux1 = 0;
aux2 = aux3 = aux4 = 0;
}
}
}
// Configuracion de las condicion alerta de robo
if (aux2 == 1)
{
if (sp == HIGH)
{
message1[0]='\0';
strcat(message1,"Alerta: Puerta Lateral Abierta");
Serial.println("A");
aux2 = 2;
}
}
if (aux3 == 1)
{
if (sc == HIGH)
{
message1[0]='\0';
strcat(message1,"Alerta: Compartimiento Abierto");
Serial.println("B");
aux3 = 2;
}
}
// Condiciones para mostrar aperturas de puertas en pantalla touch
{
if (sp == LOW && sc == LOW && aux5 == 1)
{
Serial.println(3); //Envia el numero 3 para indicar "puertasabiertas"
aux5 = 2;
aux8 = 1;

}
if (sp == LOW && sc == HIGH && aux6 == 1)
{
Serial.println(3); //Envia el numero 3 para indicar "puertasabiertas"
aux6 = 2;
aux8 = 1;
}
if (sp == HIGH && sc == LOW && aux7 == 1)
{

```

```
Serial.println(3); //Envia el numero 3 para indicar "puertas abiertas"
aux7 = 2;
aux8 = 1;
}
if (sp == HIGH && sc == HIGH && aux8 == 1)
{
Serial.println(2); //Envia el numero 2 para indicar "puertas Cerradas"
aux8 = 2;
aux5 = 1;
aux6 = 1;
aux7 = 1;
}
}
}
```

ANEXO H.- Código de programación para establecer la comunicación entre Raspberry pi y Arduino mediante Processing.

```
import processing.serial.*; //libreria del Puerto serial
Serial puerto;
PrintWriter output;
int valor;
int aux;
int salto;
int dato;
PImage autoabierto;
PImage fondo;
PImage nivelfuel;
PImage temp;
PImage aceite;
PImage bat;
PImage engine;
PImage close;
PImage dooropen;
PImage capoopen;

int circleX, circleY; // Position of circle button
int circleX1, circleY1; // Position of circle button
int circleX2, circleY2; // Position of circle button
int circleX3, circleY3; // Position of circle button
int circleX4, circleY4; // Position of circle button
int circleX5, circleY5; // Position of circle button
int circleSize = 50; // Diameter of circle
int circleSize1 = 50; // Diameter of circle
color circleColor;
color circleHighlight;
boolean circleOver = false;
boolean circleOver1 = false;
boolean circleOver2 = false;
boolean circleOver3 = false;
boolean circleOver4 = false;
boolean circleOver5 = false;
PImage mapImage;

int clave0= 0, clave1= 0, clave2= 0, clave3 = 0, clave4 = 0, clave5 = 0;
int aux2;
int aux3;

void setup()
{
  noStroke();
```



```

size(640, 460);
puerto = new Serial(this, Serial.list()[0], 9600);
//println(Serial.list());
//output = createWriter("señales.txt");

autoabierto = loadImage("carroabierto.jpg");
fondo = loadImage("carro2.jpg");
nivelfuel = loadImage("iconofuel.jpg");
temp = loadImage("temp.jpg");
aceite = loadImage("aceite2.jpg");
bat = loadImage("bat2.jpg");
engine = loadImage("engine.jpg");
dooropen = loadImage("puertas.JPG");
close = loadImage("cerrado.JPG");
capoopen = loadImage("capotpuertas.jpg");

circleColor = color(0);
circleHighlight = color(255);
//circleColor1 = color(255);
//circleHighlight1 = color(204);
circleX = 270;
circleY = 350;
circleX1 = 330;
circleY1 = 350;
circleX2 = 390;
circleY2 = 350;
circleX3 = 270;
circleY3 = 410;
circleX4 = 330;
circleY4 = 410;
circleX5 = 390;
circleY5 = 410;
}

void draw()
{
  update(mouseX, mouseY);
  background(255);
  if (puerto.available() > 0);
  {
    dato = puerto.read();
  }
  if(dato>1)
  {
    println(dato);
    aux2 = dato;
  }
  if(aux2 == 50)

```

```

aux3 = 50;
if(aux2 == 51)
aux3 = 51;
if (aux3 == 50)
{
  image(fondo, 90, 10);
}
if (aux3 == 51)
{
  image(autoabierto, 90, 0);
}

if(clave3 == 1)
{
  image(capoopen,550,60);
  print(clave3);
  puerto.write(113);
  puerto.write("q");
  clave3=0;
}
else
{
  //background(255);
}
if (circleOver)
{
  fill(circleHighlight);
} else
{
  fill(circleColor);
}
stroke(0);
ellipse(circleX, circleY, circleSize, circleSize);

if (circleOver1)
{
  fill(circleHighlight);
} else
{
  fill(circleColor);
}
stroke(0);
ellipse(circleX1, circleY1, circleSize1, circleSize1);

if (circleOver2)
{
  fill(circleHighlight);
} else

```

```

    {
        fill(circleColor);
    }
    stroke(0);
    ellipse(circleX2, circleY2, circleSize, circleSize);

    if (circleOver3)
    {
        fill(circleHighlight);
    } else
    {
        fill(circleColor);
    }
    stroke(0);
    ellipse(circleX3, circleY3, circleSize, circleSize);

    if (circleOver4)
    {
        fill(circleHighlight);
    } else
    {
        fill(circleColor);
    }
    stroke(0);
    ellipse(circleX4, circleY4, circleSize, circleSize);

    if (circleOver5)
    {
        fill(circleHighlight);
    } else
    {
        fill(circleColor);
    }
    stroke(0);
    ellipse(circleX5, circleY5, circleSize, circleSize);
}

void update(int x, int y)
{
    if ( overCircle(circleX, circleY, circleSize) )
    {
        circleOver = true;
    }
    else
    {
        circleOver = false;
    }
    if ( overCircle1(circleX1, circleY1, circleSize) )

```

```

{
  circleOver1 = true;
}
else
{
  circleOver1 = false;
}
if ( overCircle(circleX2, circleY2, circleSize) )
{
  circleOver2 = true;
}
else
{
  circleOver2 = false;
}
if ( overCircle1(circleX3, circleY3, circleSize) )
{
  circleOver3 = true;
}
else
{
  circleOver3 = false;
}
if ( overCircle1(circleX4, circleY4, circleSize) )
{
  circleOver4 = true;
}
else
{
  circleOver4 = false;
}
if ( overCircle(circleX5, circleY5, circleSize) )
{
  circleOver5 = true;
}
else
{
  circleOver5 = false;
}
}

void mousePressed()
{

  if (circleOver)
  {
    //image(autoabierto, 90, 80);
    clave1 = 0;
  }
}

```

```

    clave2 = 0;
    clave3 = 0;
}
//image(fondo, 90, 100);
if (circleOver1)
{
    //image(fondo, 90, 0);
    clave1 = 1;
}
if (circleOver2)
{
    //image(fondo, 90, 0);
    if(clave1==1)
    {
        clave2 = 1;
    }
}
if (circleOver3)
{
    //image(fondo, 90, 0);
    clave1 = 0;
    clave2 = 0;
    clave3 = 0;
}
    if (circleOver4)
    {
        //image(fondo, 90, 0);
        clave1 = 0;
        clave2 = 0;
        clave3 = 0;
    }
    if (circleOver5)
    {
        //image(fondo, 90, 0);
        if(clave2 ==1)
        {
            clave3 = 1;
        }
        else
        {
            clave1 = 0;
        }
    }
}
boolean overCircle(int x, int y, int diameter)
{
    float disX = x - mouseX;

```

```

float disY = y - mouseY;
if (sqrt(sq(disX) + sq(disY)) < diameter/2 )
{
    return true;
} else
{
    return false;
}
}
boolean overCircle1(int x1, int y1, int diameter1)
{
    float disX1 = x1 - mouseX;
    float disY1 = y1 - mouseY;
    if (sqrt(sq(disX1) + sq(disY1)) < diameter1/2 )
    {
        return true;
    } else
    {
        return false;
    }
}
boolean overCircle2(int x2, int y2, int diameter2)
{
    float disX2 = x2 - mouseX;
    float disY2 = y2 - mouseY;
    if (sqrt(sq(disX2) + sq(disY2)) < diameter2/2 )
    {
        return true;
    } else
    {
        return false;
    }
}
boolean overCircle3(int x3, int y3, int diameter3)
{
    float disX3 = x3 - mouseX;
    float disY3 = y3 - mouseY;
    if (sqrt(sq(disX3) + sq(disY3)) < diameter3/2 )
    {
        return true;
    } else
    {
        return false;
    }
}
boolean overCircle4(int x4, int y4, int diameter4)
{
    float disX4 = x4 - mouseX;

```

```

float disY4 = y4 - mouseY;
if (sqrt(sq(disX4) + sq(disY4)) < diameter4/2 )
{
    return true;
} else
{
    return false;
}
}
boolean overCircle5(int x5, int y5, int diameter5)
{
    float disX5 = x5 - mouseX;
    float disY5 = y5 - mouseY;
    if (sqrt(sq(disX5) + sq(disY5)) < diameter5/2 )
    {
        return true;
    } else
    {
        return false;
    }
}
}

```

ANEXO I.- Pruebas del funcionamiento del Ordenador Vehicular

Pruebas de funcionamiento del bloqueo centralizado y control de puertas

Para desactivar el Bloqueo centralizado ingresamos la clave que fue configurada en el sistema, al ingresarla correctamente el ordenador Electrónico activa la energía al vehículo como se muestra en la figura Siguiete:



Figura 4.58 Desactivación del Bloqueo Centralizado mediante la pantalla Touch

Fuente: El Investigador

La Siguiete figura muestra la información de alguna puerta o compartimiento abierto en la pantalla del Ordenador vehicular cuando el sistema de seguridad no ha sido activado.



Figura 4.59 Reacción del Ordenador Vehicular ante una puerta o compartimiento abierto del automóvil

Fuente: El Investigador

En la Siguiete figura se muestra la información del vehículo con las puertas y compartimientos cerrados cuando el sistema de seguridad no ha sido activado



Figura 4.60 Reacción del Ordenador Vehicular ante una puerta o compartimiento cerrado del automóvil

Fuente: El Investigador

Prueba de funcionamiento del sistema de bloqueo

En la Siguiete figura Se muestra el sistema de bloqueo de acceso al vehículo activado mediante la aplicación móvil.

Nota: El sistema cierra los seguros de las puertas y activa los sensores.



Figura 4.61 Activación del Sistema de bloqueo de acceso al vehículo

Fuente: El Investigador

En la Siguiete figura se muestra el sistema de bloqueo cuando este es desactivado

Nota: El sistema abre los seguros y desactiva los sensores.

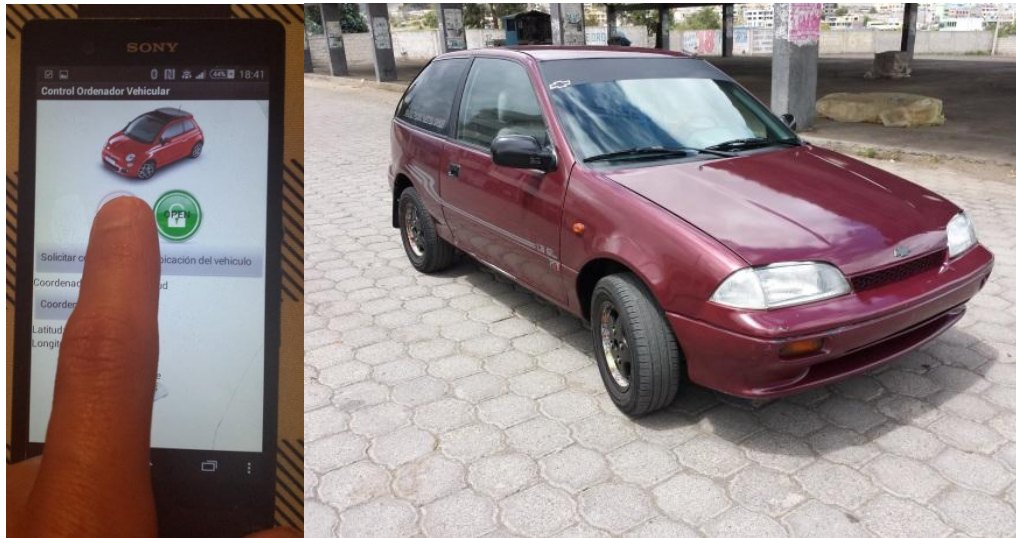


Figura 4.62 Desactivación del Sistema de bloqueo de acceso al vehículo

Fuente: El Investigador

En la siguiente figura se muestra la reacción del Ordenador Vehicular cuando alguna puerta lateral es abierta.

Nota: el ordenador reacciona cuando el sistema está activado y sufre intento de atraco.

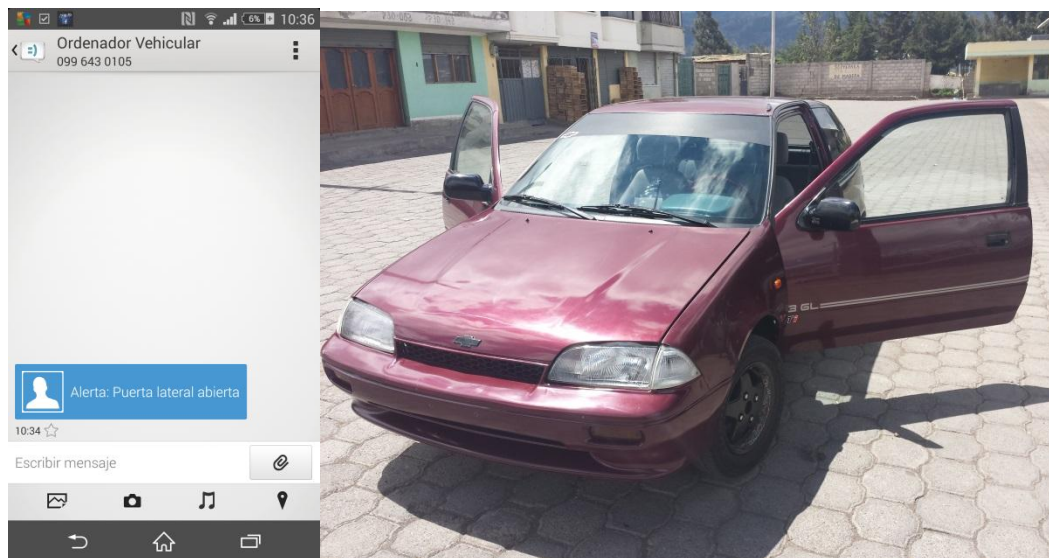


Figura 4.63 Reacción del Ordenador cuando alguna puerta lateral es abierta

Fuente: El Investigador

En la siguiente figura se muestra la reacción del Ordenador Vehicular cuando algún compartimiento es abierto.

Nota: El ordenador reacciona cuando el sistema está activado y sufre intento de atraco.

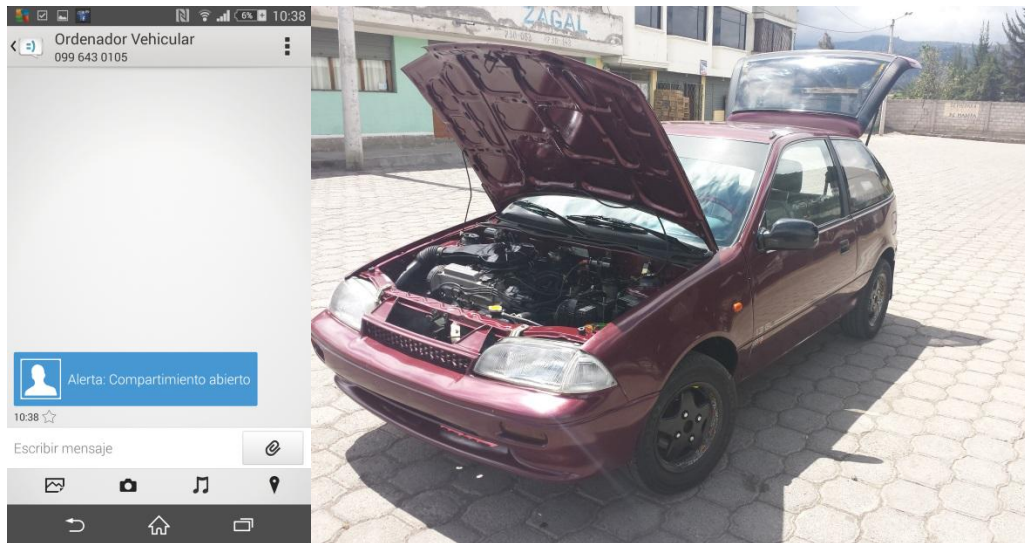


Figura 4.64 Reacción del Ordenador cuando alguna puerta lateral es abierta

Fuente: El Investigador

Prueba de funcionamiento del sistema GPS

En la siguiente figura se muestra el funcionamiento del sistema cuando se solicita la ubicación GPS del Vehículo mediante la aplicación móvil.

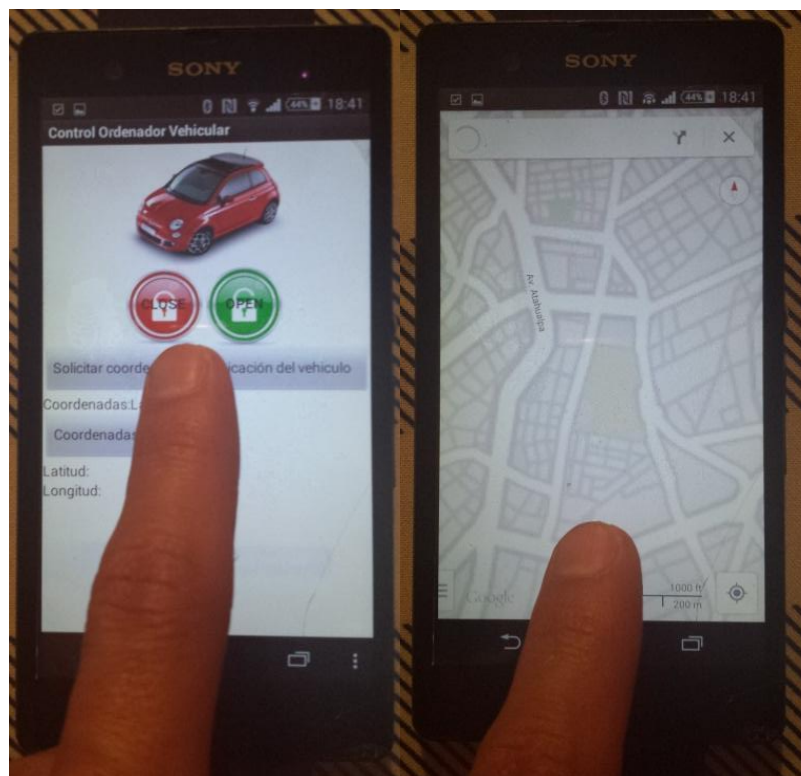


Figura 4.65 Localización del vehículo con datos del GPS

Fuente: El Investigador