



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE TECLADO
INVISIBLE PARA SEGURIDAD DE VEHÍCULOS USANDO UNA PANTALLA
TÁCTIL (TECLADO INVISIBLE), RFID Y MENSAJES DE TEXTO.**

AUTOR: EDIN FERNANDO LIMA CASTILLO

TUTOR: Ing. David Cando, Mg.

AÑO: 2017

INFORME FINAL DE RESULTADOS DEL PIC

CARRERA:	ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES
AUTOR/A:	LIMA CASTILLO EDIN FERNANDO
TEMA DEL TT:	DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE TECLADO INVISIBLE PARA SEGURIDAD DE VEHÍCULOS USANDO UNA PANTALLA TÁCTIL (TECLADO INVISIBLE), RFID Y MENSAJES DE TEXTO
ARTICULACIÓN CON LA LÍNEA DE INVESTIGACIÓN INSTITUCIONAL:	TECNOLOGÍA APLICADA A LA PRODUCCIÓN Y SOCIEDAD.
SUBLÍNEA DE INVESTIGACIÓN INSTITUCIONAL:	PANTALLA RESISTIVA, TECNOLOGÍA RFID, SMS
ARTICULACIÓN CON EL PROYECTO DE INVESTIGACIÓN INSTITUCIONAL DEL ÁREA	SEGURIDAD VEHICULAR CON NUEVAS TECNOLOGÍAS
FECHA DE PRESENTACIÓN DEL INFORME FINAL:	ENERO 2017

DEDICATORIA

En cada una de las letras de este proyecto, van dirigidas primeramente a Dios y su hijo Jesucristo mi Salvador.

A mí querida esposa que ha sido mi apoyo incondicional, su amor hizo posible la culminación de esta etapa de mi vida.

A mi hija Danna, a mi hijo Santiago quienes son la alegría de mi corazón y mi razón de cada día.

Edin Lima

AGRADECIMIENTO

Agradezco a Dios por darme la oportunidad de cumplir una meta más de mi vida.

A mi familia por todo el apoyo, comprensión y sacrificio que conjuntamente hemos realizado para cumplir este logro.

A todos: maestros, compañeros, amigos, y familiares cuyo recuerdo llevaré por siempre en mi mente y corazón.

A la Universidad Tecnológica Israel y a todo su personal docente que contribuyó con el desarrollo y profesionalismo de sus alumnos.

Gracias

Edin Lima

RESUMEN

El constante desarrollo de la tecnología electrónica permite tener sistemas de seguridad cada vez más complejos, sin embargo, muchos de estos resultan demasiado costosos. Por este motivo, surge la necesidad de encontrar un sistema que use tecnología nueva, sea accesible, y sobretodo garantice la seguridad del automotor.

Plataformas como Arduino UNO, que junto con su hardware (microcontrolador Atmega, entradas y salidas digitales, entradas analógicas, conexión USB, botón de reinicio, etc), su software de código abierto, de costo relativamente bajo, y una serie de productos compatibles en el mercado entre otras características, hicieron posible el desarrollo del proyecto.

El presente prototipo de sistema de seguridad para vehículos integró tecnología RFID como su mecanismo de autenticación, una pantalla resistiva donde ingresa un código único y junto con el envío de mensajes de texto vía SMS al usuario final, permitieron un producto fiable, seguro y de costo promedio, una alternativa novedosa frente los sistemas tradicionales.

DESCRIPTORES:

Plataforma Arduino

Sistemas de seguridad vehicular

Pantallas táctiles

Tecnología RFID

Shield Sim900

ABSTRACT

The constant development of the electronic technology lets us have more complex security systems, however, many of them turns out to be too expensive, for this reason, there is a need of finding a system that uses new accessible technology, and overall that guarantees the safety of the car.

Platforms Like Arduino Uno, with its hardware (Atmega microcontroller, digital output and input, analogic inputs, USB connections, reset buttons, etc.) It's relatively low cost open code software, and a series of compatible products in the market, and other characteristics, made possible the development of the project.

This car lock systems prototype, integrated RFID technology as its authentication system, a resistive touchscreen to enter a unique code, and together with text messages sent via SMS to the final user, allowed a reliable, safe and an average cost product as a novel alternative to the traditional systems.

DESCRIPTORS:

Arduino

Car lock systems

Touch screen

RFID technology

Shield Sim900

ÍNDICE GENERAL

INFORME FINAL DE RESULTADOS DEL PIC	I
DEDICATORIA.....	II
AGRADECIMIENTO.....	III
RESUMEN.....	IV
ABSTRACT.....	V
1. INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
2. FUNDAMENTACIÓN TEÓRICA.....	3
2.2 ARDUINO	3
2.3 MICROCONTROLADORES ATMEL	3
2.4 ARDUINO UNO.....	3
2.4.1 Entradas y Salidas de un Arduino Uno	4
2.4.2 Alimentación de un Arduino Uno	5
2.5 PANTALLAS TÁCTILES	5
2.5.1 Pantallas resistivas y capacitivas.....	5
2.6 MÓDULO SHIELD GSM SIM900	8
2.6.1 GSM (Global System for Mobile Communications)	9
2.6.2 SMS (Short Message Service).....	9
2.7 TECNOLOGÍA RFID.....	10
2.7.1 Módulo RFID RC522 y sus componentes.....	10
2.8 BLOQUEO INTELIGENTE EN AUTOMÓVILES	11
2.8.1 Bloqueo inteligente en el Sistema de encendido.....	11
2.8.2 Bloqueo inteligente en el Sistema de arranque.....	12

2.8.3	Bloqueo inteligente en el Sistema de combustible	13
2.9	ANÁLISIS DE SISTEMAS DE ALARMAS E INMOVILIZADORES	15
2.9.1	Alarmas Eléctricas.....	15
2.9.2	Alarmas Electrónicas.....	15
2.9.3	Sistema de Inmovilizadores.....	16
2.9.4	Localizadores	18
3.	BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO	19
3.1	PROBLEMA PRINCIPAL.....	19
3.2	PROBLEMAS SECUNDARIOS.....	19
3.2.1	Por qué y para qué de los objetivos.....	19
3.2.2	Hipótesis o idea a defender	19
3.2.3	Métodos utilizados para el desarrollo del proyecto	20
4.	PRESENTACIÓN DE LOS RESULTADOS.....	21
4.1	DISEÑO DEL SISTEMA	24
4.2	DIAGRAMA DE FLUJO	25
4.2.1	Modo normal de funcionamiento.....	25
4.2.2	Modo Mantenimiento	27
4.2.3	Modo Antiatraco	29
4.2.4	Modo de Reinicio.....	30
4.3	ESQUEMA DEL CIRCUITO	31
4.4	IMPLEMENTACIÓN	32
4.4.1	Diseño de placa base	33
4.4.2	Fabricación y ensamblaje de elementos.....	34
4.4.3	Interconexión Placa base con módulo Arduino Uno.....	35
4.4.4	Interconexión módulo Arduino Uno con módulo Sim900.....	35
4.5	PRUEBAS DE COMPROBACIÓN DE DISPOSITIVOS:	38

4.6	PRUEBAS DE OPERATIVIDAD.....	40
4.7	ANÁLISIS DE COSTOS.....	47
	CONCLUSIONES.....	49
	RECOMENDACIONES.....	50
	BIBLIOGRAFÍA.....	51
	REFERENCIAS BIBLIOGRAFICAS.....	52
	ANEXOS.....	54

ÍNDICE DE FIGURAS

Figura 1 Arduino Uno con microcontrolador en formato DIP y SMD	4
Figura 2. Pantalla Resistiva	6
Figura 3. Estructura de pantalla resistiva	7
Figura 4. Pantalla Capacitiva	8
Figura 5 Módulo GSM Sim900. a) Vista frontal. b) Vista posterior	9
Figura 6 Módulo RFID RC522 y componentes.....	11
Figura 7 Diagrama básico del sistema de encendido de un automotor.	12
Figura 8 Diagrama básico del sistema de arranque	13
Figura 9 Sistema de combustible.....	14
Figura 10 Estructura en bloque del sistema	21
Figura 11 Comparativo de placas Arduino	22
Figura 12 Módulo SIM900.....	22
Figura 13 Módulo RFID-RC522	23
Figura 14 Interruptor tipo pin.....	23
Figura 15 Pantalla Resistiva de 4 hilos / 5,7 pulgadas.	24
Figura 16 Diagrama de bloques del Hardware del sistema	25
Figura 17 Diagrama de flujo de Software	26
Figura 18 Diagrama de flujo Ingreso/Salida del modo Mantenimiento.....	28
Figura 19 Diagrama de flujo Modo Antiatraco/Secuestro	29
Figura 20 Diagrama de flujo Modos de reseteo	30
Figura 21 Diagrama del circuito	32
Figura 22 Diseño de placa base y vista 3D.....	34
Figura 23 Fabricación y ensamblaje de elementos de placa base	35
Figura 24 Interconexión Placa base con módulo Arduino Uno.....	35

Figura 25 Interconexión Placa base con módulo Arduino Uno	36
Figura 26 Interconexión Arduino Uno - Sim900 y placa de control.....	37
Figura 27 Diagrama de Pareto.....	39
Figura 28 Conexión de dispositivos al módulo central	40
Figura 29 Arranque del sistema y sus componentes.....	41
Figura 30 Autenticación de tarjeta o tag RFID.....	41
Figura 31 Sensor de apertura y cierre de puerta del automóvil	43
Figura 32 Sistema de encendido activado	43
Figura 33 Captura de pantalla de mensajes de texto enviados al usuario.....	45
Figura 34 Prototipo final.....	46

ÍNDICE DE TABLAS

Tabla 1 Pines de entrada/salida en el módulo Arduino Uno.....	31
Tabla 2 Conexión de pines del entre el módulo RFID y Arduino Uno.....	37
Tabla 3 Pruebas de encendido	38
Tabla 4 Lectura hexadecimal RFID.....	42
Tabla 5 Tabla de mensajes de texto enviados al usuario.....	45
Tabla 6 Costos del prototipo	47

1. INTRODUCCIÓN

De acuerdo a los datos presentados por el Ministerio del Interior (Minterior, 2015), en su página web, en Quito son más de 10.000 casos entre motocicletas y automóviles recuperados o retenidos por la Policía Judicial en los 10 primeros meses del año, resultado de los operativos de control para intentar frenar el alto índice de robo, alteración en las series de motor y chasis vehicular. Afectando sobretodo en jurisdicciones con mayor densidad poblacional, Quito como el eje capitalino y Guayaquil el de mayor actividad comercial y económica, luego continúan las provincias de Manabí, Los Ríos, Santo Domingo de los Tsáchilas y Azuay respectivamente, los días de mayor incidencia son los miércoles, viernes, sábados y domingos, sobre todo en horario de 16:00 a 23:59 horas.

El robo de vehículos afecta a todas las regiones del mundo, esta actividad delincuenciales tiene un alto nivel de organización, según el análisis de la Policía Nacional, en el norte de Quito, los robos se producen en su mayoría en parqueaderos no autorizados o establecidos por el Municipio Metropolitano, en los exteriores de entidades comerciales y bancarias. Por el contrario, en el sur se suscitan en su mayoría en la madrugada, cuando sus propietarios descansan en sus hogares o mientras se recrean en lugares públicos.

La delincuencia hoy en día es un problema muy común, especialmente el robo de automóviles, partes y accesorios de los mismos, muchos vehículos no cuentan con un sistema básico de alarma, el bloqueo vehicular mediante cualquier dispositivo constituye un éxito a la hora de neutralizar cualquier intento de robo, la protección del vehículo debe ser una prioridad.

En el mercado existen diversos mecanismos de seguridad, no todos aptos para cualquier auto. Los sistemas antirrobo clásicos, tales como: bloqueo de dirección, barra de volante o alarmas acústicas, son eficientes para persuadir a ladrones poco experimentados. Otros dispositivos más sofisticados incluyen tecnología de punta, como los inmovilizadores tipo electrónicos que evitan el robo al desconectar un circuito vital del auto, eficaces para desamistar a ladrones "profesionales".

Según estudios realizados por OCU (Organización de Consumidores y Usuarios), los sistemas antirrobo más destacados para el auto son los denominados inmovilizadores electrónicos. Estos dispositivos se clasifican en tres tipos según la zona donde se sitúe

el equipo encargado de dar la orden de activar/desactivar los circuitos, y son los siguientes:

1. En la llave de contacto, el dispositivo identifica la llave y permite arrancar el auto. La llave y el sistema receptor no se detectan desde el exterior.
2. En el mando a distancia el acceso es a través de señales infrarrojas o por ondas de radiofrecuencia. Su desventaja es que el código puede llegar a retransmitirse, se recomienda uno que use un código modificable.
3. En llaves inteligentes similares a una tarjeta o llavero, cuando el vehículo detecta su aproximación, cancela el inmovilizador, habilita la apertura de puertas y el arranque del vehículo.

1.1 Objetivo General

Elaborar un prototipo de sistema de bloqueo antirrobo para vehículos mediante el uso de pantalla táctil resistiva, tecnología RFID y mensajes de texto.

1.2 Objetivos específicos

- Analizar los distintos tipos de sistemas de bloqueo electrónico existentes en el mercado en función de sus vulnerabilidades.
- Diseñar un sistema de bloqueo electrónico de seguridad con nuevas tecnologías tales como una pantalla táctil, etiquetas RFID y mensajes de texto.
- Implementar el prototipo del sistema de seguridad de bloqueo antirrobo
- Validar el prototipo del sistema de bloqueo antirrobo mediante el uso de pantalla táctil resistiva, tecnología RFID y mensajes de texto.

2. FUNDAMENTACIÓN TEÓRICA

2.2 Arduino

Es una plataforma para desarrollo electrónico de estudiantes, profesionales o simplemente aficionados que quieren incursionar en la electrónica y control, es fácil y flexible para la creación de proyectos (código abierto), está basada en una sencilla placa de circuito impreso que contiene un microcontrolador de la marca "ATMEL". (ATMEGA168, ATMEGA328, ATMEGA1280), cuenta con entradas y salidas analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing. (Regalado, 2011).

Arduino puede ser programado para tomar información del mundo que lo rodea a través de sus pines de entrada mediante una gran variedad de sensores, o controlar su entorno a través de sus pines de salida. 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) o con otros programadores como BASCOM. Los proyectos con Arduino pueden ejecutarse sin la necesidad de permanecer conectado a un computador". (Regalado, 2011)

2.3 Microcontroladores ATMEL

Las tarjetas Arduino son placas sencillas que incluye un microcontrolador de la marca Atmel llamada AVR 8-Bit RISC (Reduced Instruction Set Computer), esta línea de microcontroladores la conforman varios grupos, entre los cuales está el Atmega. Cada familia se diferencia por los periféricos y la cantidad de memoria que pueden manejar. El microcontrolador de este prototipo es el Atmega 328P, que contiene una CPU, memoria RAM para datos, memoria ROM, PROM, EPROM para escritura del programa, pines de entrada y salida y algunos periféricos. (Tapia Carlos, 2013)

2.4 Arduino UNO

Es una placa que contiene un microcontrolador de la marca Atmel y una serie de circuitos de soporte dentro del mismo que incluye: reguladores de tensión, puerto USB desde el cual se programa el microcontrolador a través de un computador de manera fácil, cómoda y también realizar pruebas de comunicación con el mismo, puertos de

entradas análogas, botón reset, leds indicadores de encendido, recepción y transmisión, puertos de entradas o salidas digitales, entre otras características de su hardware.

Arduino Uno dispone de 14 pines configurables como entrada o salida, acepta cualquier dispositivo apto para transmitir o recibir señales digitales de 0 y 5 V. Dispone de pines de entradas y salidas analógicas, las entradas permiten recibir datos de los sensores traducidos en variaciones continuas de voltaje. Las salidas analógicas por lo general se utilizan para el envío de señales de control en forma de PWM (Modulación por ancho de pulsos).

Se desarrollaron dos tipos de variantes de acuerdo al microcontrolador que poseen, el convencional con un microcontrolador en formato DIP (Dual in-line package o DIL), y el segundo con un microcontrolador en formato SMD (Surface-Mounted Device), tal como observa en la figura 1. En este proyecto se optó por el Arduino Uno con microcontrolador en formato SMD disponible en cualquier tienda tecnológica del país. (Regalado, 2011)



Figura 1 Arduino Uno con microcontrolador en formato DIP y SMD

Fuente: Arduino.cc (2013)

2.4.1 Entradas y Salidas de un Arduino Uno

Como se mencionó anteriormente, Arduino Uno posee 14 pines configurables como entrada o salida que funcionan a 5 V, cada pin permite entregar hasta 40 mA. La corriente máxima de entrada igualmente es de 40 mA. Los pines digitales tienen una resistencia de pull-up interna de entre 20 k Ω y 50 k Ω desconectada, salvo que el usuario indique lo contrario. Arduino Uno además dispone de 6 pines de entrada analógicos que envían las señales a un conversor análogo/digital de 10 bits. En el Anexo B y C se

encuentran cuadros con los resúmenes de las características del Arduino Uno y Atmega328.

2.4.2 Alimentación de un Arduino Uno

Puede alimentarse de dos maneras, la primera directamente mediante un cable USB; o mediante una fuente de alimentación externa (por conector 2,1mm o por diseño de placa a los pines). El rango límite de voltaje está entre los 6 a 12 V, si se alimenta con un voltaje menor a 7 V, posiblemente la salida del regulador de tensión de 5 V dará menos que este voltaje y si sobrepasa los 12 V, posiblemente dañe la placa. (Regalado, 2011)

2.5 Pantallas táctiles

Las pantallas táctiles existen desde finales de los años 70, pero no fue hasta el año 2007 que Apple revoluciona el mercado con el lanzamiento de su Iphone, desde ese momento nació una nueva manera de entender a los dispositivos portátiles, actualmente en el mercado se encuentran diferentes tipos de pantallas con retículas tipo touch o táctiles, esto depende del tipo de tecnología usada, permiten leer exactamente la posición o coordenada por decirlo así entre X y Y de donde se presionó, las más comunes son las pantallas táctiles resistivas y las capacitivas. (K-twin, 2011)

2.5.1 Pantallas resistivas y capacitivas

Las pantallas táctiles más usadas y comunes en el mercado son las pantallas táctiles resistivas y capacitivas. Las pantallas resistivas fueron de las primeras en desarrollarse y son las más extendidas, pueden usarse con objetos como punteros, lápices, o con la uña. Se componen de dos finas capas entre las cuales hay un espacio. Al presionar la pantalla como observa en la figura 2, la capa superior toca a la capa inferior en algún punto, el dispositivo ubica ese lugar y responde al toque. Este sistema tiene una desventaja puesto que la respuesta no es muy efectiva al tacto de los dedos, esto implica el uso de punteros especiales. Además, las pantallas resistivas son vulnerables a rayaduras y perforaciones. El costo de fabricación es más barato, aunque de inferior calidad a las capacitivas. (K-twin, 2011)

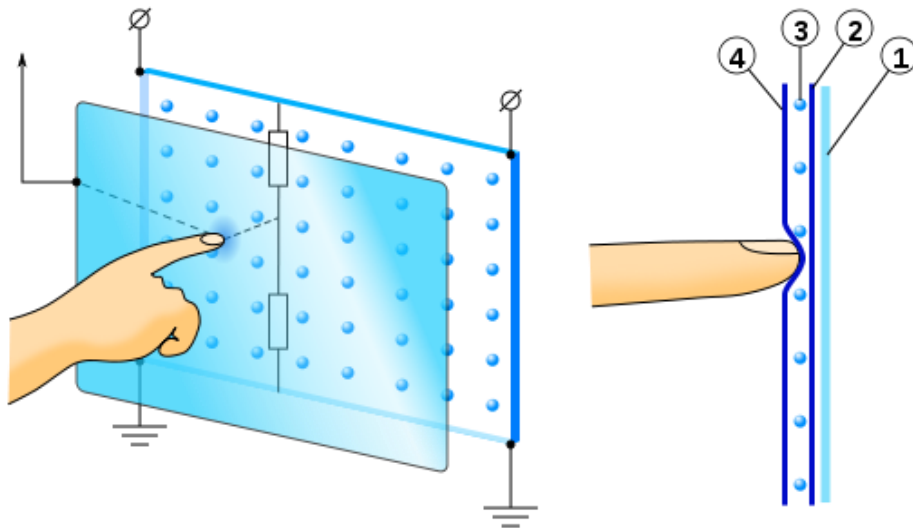


Figura 2. Pantalla Resistiva

Fuente: Ecojovent.com (2015)

Las pantallas resistivas las hay en el mercado de 4 hasta 8 hilos así también de diferentes medidas, en este proyecto se usó una de 4 hilos o pines, como ya se mencionó las pantallas touch tienen varias capas y un aislante, son tratadas normalmente con óxido de indio y estaño, además de una barra conductora (tinta de plata) en dos lados opuestos como se observa en la figura 3. Al presionar la primera lámina provoca contacto entre ambas capas, esto genera un nivel resistivo en X y Y. El sistema lo detecta y mide la resistencia para calcular el punto de presión o contacto.

Mediante un programa en el módulo Arduino Uno primero se identifica cada uno de los 4 pines de la pantalla resistiva, básicamente cada capa funciona como un potenciómetro, una vez identificados los valores (Xalto, Xbajo, Yalto y Ybajo), se conectan al módulo Arduino Uno a través de su convertidor analógico-digital, quien digitaliza la tensión analógica generada al presionar sobre la pantalla. Arduino mide esta tensión y calcula la coordenada "X" y "Y" del punto de contacto. (K-twin, 2011)

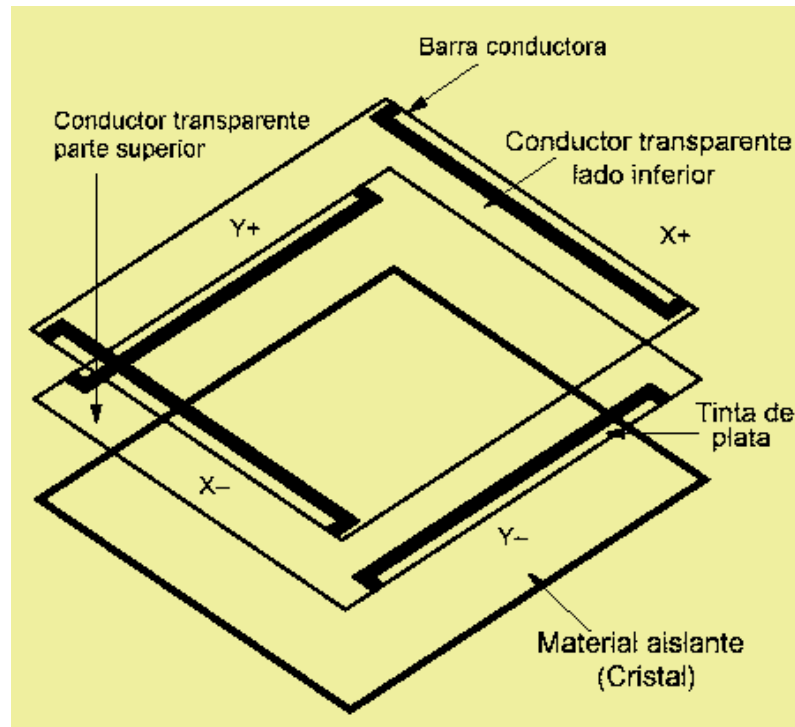


Figura 3. Estructura de pantalla resistiva

Fuente: Ecojovent.com (2015)

La pantalla capacitiva es más avanzada, posee una capa que detecta las variaciones eléctricas en su superficie, en el momento que un objeto con diferente carga eléctrica toca la pantalla ésta localiza dónde se produjo el cambio y responde a tal efecto como observa en la figura 4. El ser humano es un conductor de la electricidad, por lo que estas pantallas son adecuadas para los dedos,

Las pantallas resistivas son más antiguas respecto a las capacitivas, una de las ventajas de las pantallas táctiles capacitivas es que pueden reconocer 2 o más puntos a la misma vez, a esto se le llama multitouch, las pantallas táctiles resistivas no funcionan así, se deben leer una coordenada a la vez al sentir la presión entre las capas o retículas. (K-twin, 2011)

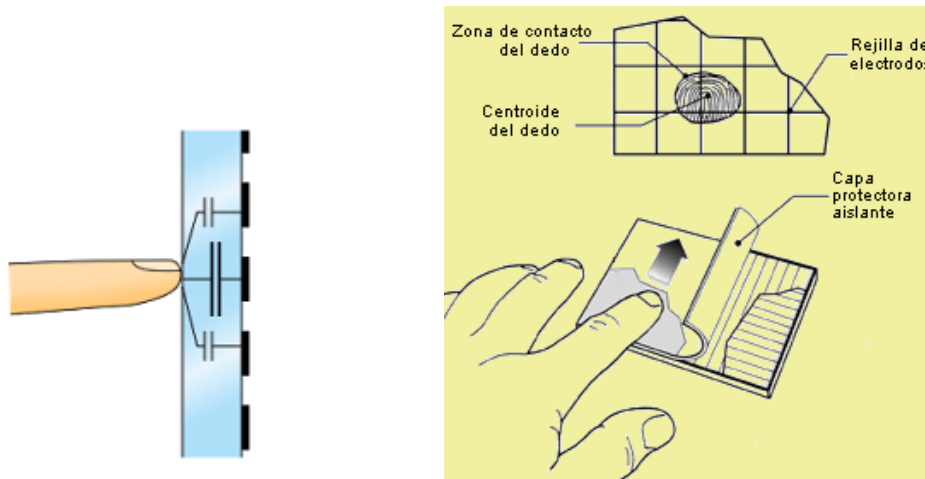


Figura 4. Pantalla Capacitiva

Fuente: Ecojuven.com (2015)

Algunas ventajas de las pantallas táctiles resistivas son: pueden ser usadas con cualquier objeto, un lápiz, puntero, un dedo, o con guantes, etc. Además, son más económicas, fiables y versátiles. Su desventaja es que, al usar varias capas de material transparente sobre la pantalla, pierde luminosidad, es sensible a los rayos UV (ultravioleta), con el paso del tiempo se degrada, pierde transparencia y flexibilidad. (K-twin, 2011)

2.6 Módulo Shield GSM SIM900

Es una tarjeta GSM/GPRS (Global System for Mobile communications / General Packet Radio Service) cuadribanda (850/900/1800/1900 MHz) basada en el módulo SIM900 muy compacta de comunicación inalámbrica que permite enviar y recibir llamadas y SMS (Short Message Service), es compatible con todos los modelos de Arduino con el formato UNO y también con otros microcontroladores.

Este módulo cuenta con un selector vía UART (Universal Asynchronous Receiver-Transmitter - Transmisor-Receptor Asíncrono Universal) a través de comandos AT (ATención = comandos para configurar y parametrizar módems), o por Serial Software (pin 7-8). Es compatible con módulos Arduino o microcontrolador, Raspberry Pi o cualquier PC. Es ideal para telemetría móvil, envío de mensajes de texto a celulares, sistemas remotos, puntos de control, comunicación recursiva, etc.

Soporta reset y encendido por hardware, se puede montar directamente sobre el Arduino, al ser cuadribanda garantiza compatibilidad con la mayoría de las operadoras móviles a nivel mundial, posee un zócalo donde se inserta la SIMCARD que es una tarjeta de identificación del suscriptor (compatible con SIMCARD de 2, 3 y 4G), como se observa en la figura 5.

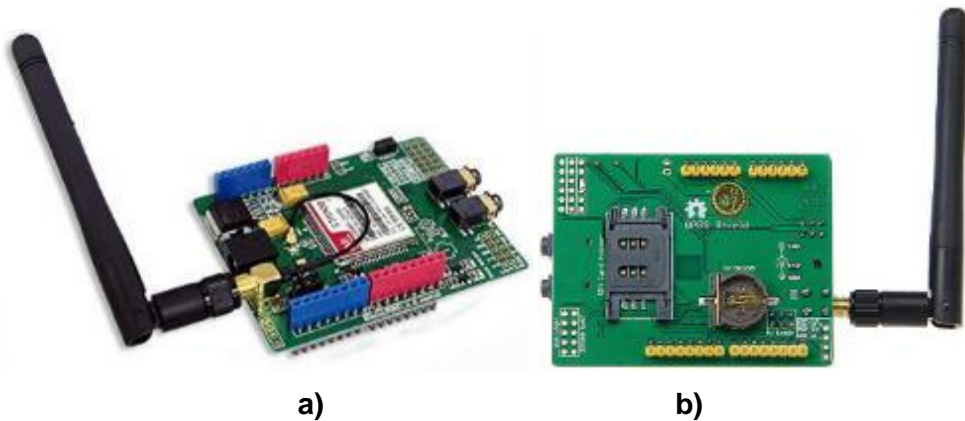


Figura 5 Módulo GSM Sim900. a) Vista frontal. b) Vista posterior

Fuente: Geeetech.com (2014)

2.6.1 GSM (Global System for Mobile Communications)

GSM es una tecnología de red celular de segunda generación (2G) desarrollado en Europa, es el estándar más extendido en telefonía móvil con más 3.000 millones de usuarios, permite transmisiones de voz y de datos digitales como mensajes de texto SMS o mensajes multimedia MMS a velocidad baja (9,6kbps). Tiene una modulación GMSK (Gaussian Minimum Shift Keying) y una técnica de acceso múltiple TDMA (Time división Multiple Access) y FDMA (Frequency Division Multiple Access) combinada. La separación o ancho de banda entre canales es de 200 KHz, los mismos que se denominan ARFCN (Absolute Radio Frequency Channel Number). (Martinez, 2011).

2.6.2 SMS (Short Message Service)

El servicio se desarrolló junto al sistema GSM, es parte del estándar de telefonía móvil digital disponible en todas las redes móviles existentes, este servicio permite enviar o recibir breves mensajes de hasta 160 caracteres con protocolos sin conexión, es decir que no se produce conexión entre el terminal que envía y el que la recibe, el estándar permite enviar mensajes desde un teléfono a otro (PP point to point) o enviar uno o más mensajes a todos los teléfonos dentro de una determinada cobertura (CB cell broadcast) (Garcia, 2006)

2.7 Tecnología RFID

RFID (Radio Frequency IDentification) es un sistema de almacenamiento, identificación y recuperación de información de forma remota de corto/mediano alcance a través de tarjetas o tags mediante ondas de radio, las mismas que poseen una pequeña antena que responde por RF desde un dispositivo emisor/receptor RFID, de manera similar a los lectores infrarrojos usados para leer código de barras.

RFID transmite la identidad de un objeto a través de ondas de radio, esta identidad es única, ventaja que es aprovechada por un sin número de nuevas aplicaciones tales como: identificación de animales, peajes electrónicos, dinero electrónico, tarjetas de transporte público, llaves inteligentes en automóviles, en implantes humanos para solucionar problemas de falsificación de identidad, pasaportes y carnets de conducción entre otras. (Rosinelys, 2013)

2.7.1 Módulo RFID RC522 y sus componentes

El Módulo Lector RFID - RC522 utiliza 3,3 V de alimentación y se controla por medio del protocolo SPI (Serial Peripheral interface) o por el protocolo UART (Universal Asynchronous Receiver-Transmitter), por lo que es compatible con casi cualquier microcontrolador, Arduino, Raspberry Pi o tarjeta de desarrollo. Este módulo utiliza un sistema de comunicación Simplex, además de un sistema de modulación y demodulación avanzado para todo tipo de dispositivos pasivos de 13.56 MHz. La tarjeta o tag que viene con el módulo RFID tiene 64 bloques de memoria (0 - 63) donde se realiza la lectura y/o escritura. Cada bloque de memoria almacena hasta 16 bytes. El número de serie único de cada tarjeta o tag consta de 5 valores hexadecimales.

Los tags o tarjetas RFID son dispositivos que reemplazan a los sistemas de tarjetas con banda magnética, como se observa en la figura 6, son tarjetas de plástico también conocidas como RFID cards o tags, pueden ser activas, semiactivas o pasivas, estas últimas no requieren de una fuente de alimentación interna, se activan cuando el lector está cerca e induce una corriente eléctrica suficiente para suministrar la energía requerida, las activas y semiactivas requieren de alimentación, normalmente una pila pequeña. (Arduino, 2016)

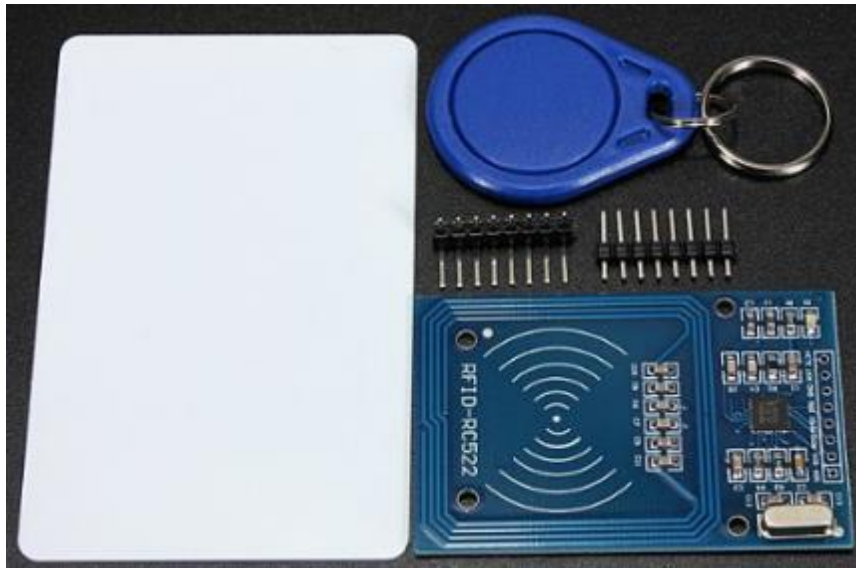


Figura 6 Módulo RFID RC522 y componentes

Fuente: SODIAL(R) MFRC-522 RC522 RFID Module (2016)

2.8 Bloqueo inteligente en automóviles

Se realizan mediante dispositivos inmovilizadores que son los encargados del envío de la señal para activar/desactivar los circuitos vitales del automóvil, se pueden realizar cortes de corriente donde la necesidad amerite, este tipo de corte de corriente se lo llama Bloqueo Inteligente. Los sistemas que normalmente se bloquean son: sistema de combustible (bomba de gasolina), al sistema de encendido/arranque (bobina en autos antiguos), o en cualquier sistema que permita que el auto se apague en un momento determinado.

2.8.1 Bloqueo inteligente en el Sistema de encendido

En la actualidad la gran mayoría de vehículos funcionan con sistemas de inyección de combustible, de tal forma que muchos de sus componentes trabajan con electricidad, este bloqueo es aplicable para cortar el paso de entrada de corriente a la ECM (Modulo de Control Electrónico), o en vehículos antiguos que no son de inyección electrónica, encendido del tipo con platino o transistorizado, se lo instala en el secundario de la bobina para que corte la corriente de entrada a la bobina. (Tomas, 2007)

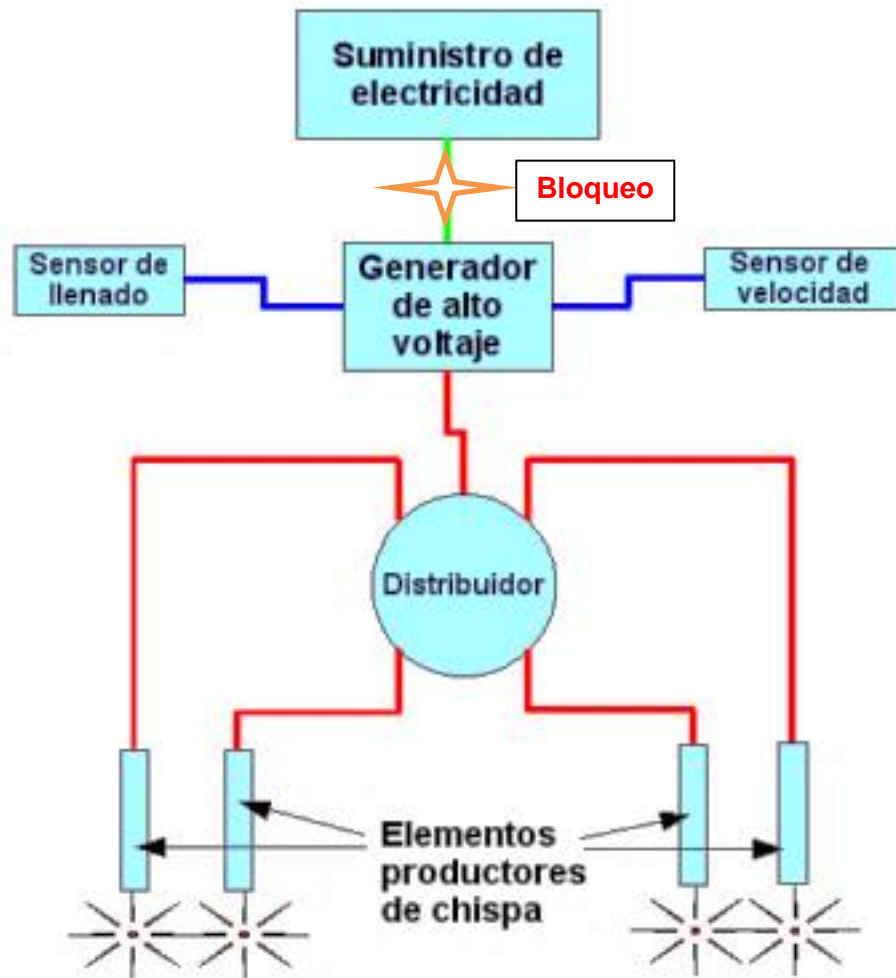


Figura 7 Diagrama básico del sistema de encendido de un automotor.

Fuente: Sabelotodo.org - Tomás Bruzos (2007)

En el mercado actual existen varios sistemas de encendido desarrollados por los fabricantes de vehículos, detallar cada uno de estos resulta extenso, pero básicamente todos se pueden resumir tal como se muestra en la figura 7, el bloqueo inteligente se lo realiza entre la etapa de suministro de electricidad y la etapa de generación de alto voltaje.

2.8.2 Bloqueo inteligente en el Sistema de arranque

El sistema de arranque usa un motor eléctrico de corriente continua, se alimenta desde la batería del auto a través de un relé. Este relé a su vez se acciona desde el interruptor de encendido del automóvil, tal como se observa en la figura 8. Al accionar

el interruptor de encendido, el relé de la bobina cierra los contactos en su interior que alimentan el motor de arranque directamente desde la batería.

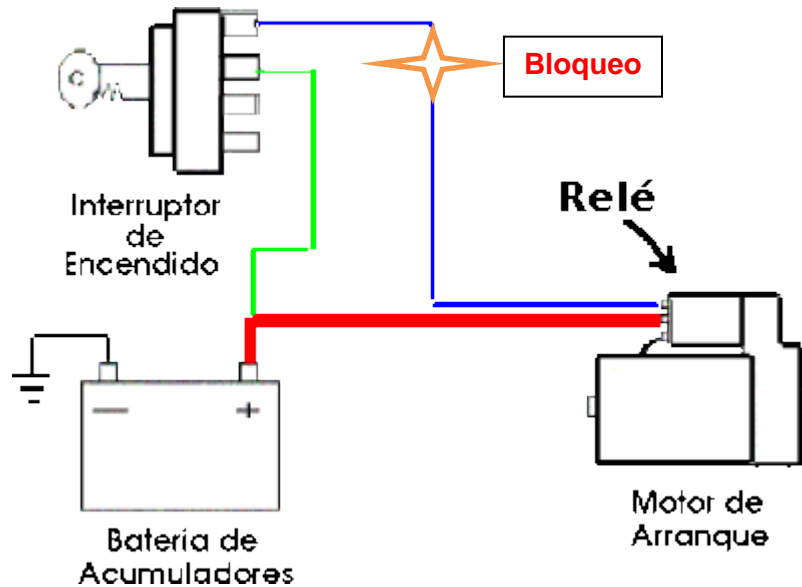


Figura 8 Diagrama básico del sistema de arranque

Fuente: Sabelotodo.org - Tomás Bruzos (2007)

El bloqueo de este sistema consiste en evitar la puesta en marcha del vehículo al abrir el circuito entre el interruptor de encendido y el motor de arranque a través de un actuador, siempre y cuando el sistema de bloqueo no haya sido desactivado. Este tipo de bloqueo es el más usado en inmovilizadores o alarmas existentes en el mercado actual. (Tomas, 2007)

2.8.3 Bloqueo inteligente en el Sistema de combustible

El sistema de combustible es quien alimenta el motor del automóvil a gasolina o a diésel para que se realice la mezcla de aire y combustible en él, las partes más importantes de este sistema son: el tanque de combustible, que es un recipiente de almacenamiento La bomba de combustible instalada dentro del tanque de combustible en los vehículos nuevos como observa en la figura 9, mientras que en los más antiguos se instalan cerca del motor o en el recorrido o vía del combustible. Si la bomba está instalada en el tanque o en el chasis funcionará a electricidad con la batería del auto,

las que están próximas al motor utilizan el movimiento del mismo para el bombeo del combustible, se consigue por acople mecánico.

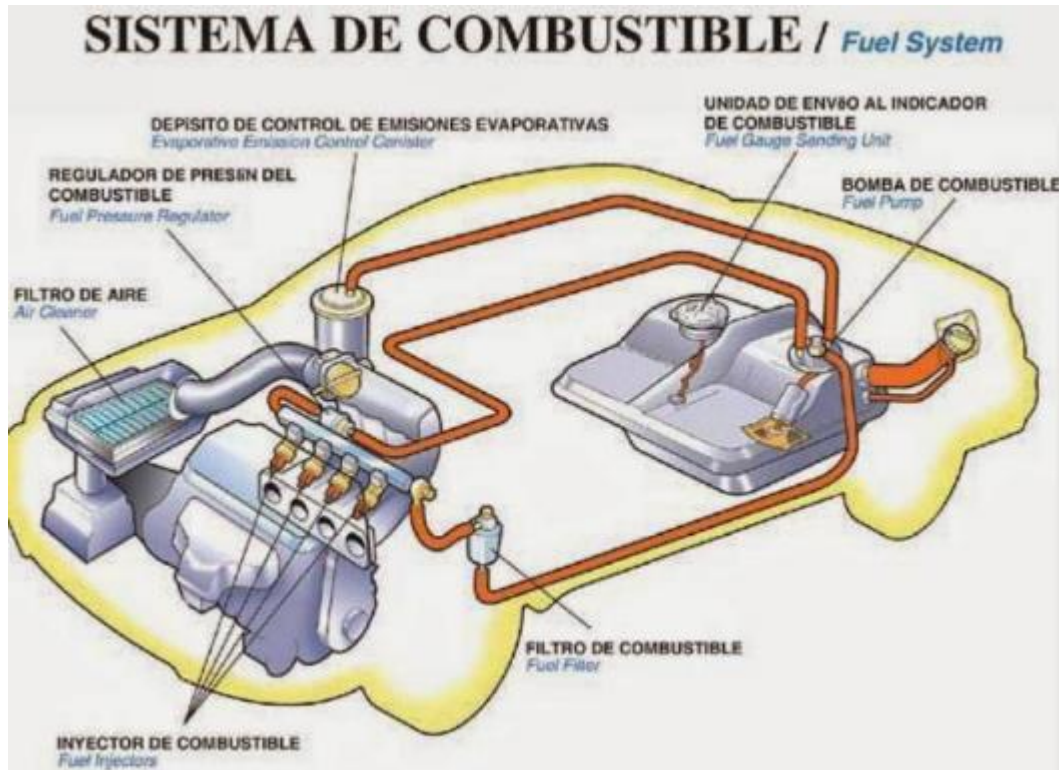


Figura 9 Sistema de combustible

Fuente: Easymecanica.blogspot.com (2013)

El bloqueo inteligente en este sistema se realiza al cortar la alimentación eléctrica de la bomba de combustible, el motor del vehículo no funcionará e irremediamente se detendrá, este bloqueo se usa comúnmente para las funciones denominadas antiatraco o antiasalto, es decir cuando el vehículo es esta en movimiento y se lo debe detener. (Bunker Electronic, 2016)

En el estudio realizado no se toman en cuenta los sistemas correspondientes a vehículos con tecnología híbrida o eléctrica, debido a la gran cantidad de sensores y componentes electrónicos sensibles que estos poseen. Además, al ser tecnologías de vanguardia incorporan sofisticados sistemas de seguridad tanto personal como integral del vehículo.

2.9 Análisis de sistemas de alarmas e inmovilizadores

La seguridad es un elemento básico al hablar de automóviles, en el mercado actual adquirir un vehículo representa un gran costo, por consiguiente, crea la necesidad de protegerlos, existen varias alternativas tanto mecánicas como electrónicas que permiten de alguna forma impedir que éstos sean robados, La mayoría de los vehículos actuales vienen equipados de fábrica con sistemas de seguridad antirrobo, que pueden ser alarmas o inmovilizadores de acuerdo a su costo y tecnología.

Los sistemas antirrobo clásicos, tales como: bloqueo de dirección, barra de volante o alarmas acústicas, resultan ineficientes para persuadir a los delincuentes, pueden ser desactivados con demasiada facilidad, son sistemas de bajo costo que no impiden el accionar delictivo, pero, dispositivos electrónicos con mayor tecnología aunque representan un incremento de costo y ser menos accesibles, aumentan la seguridad al usuario, vehículos de alta gama suelen disponer de sistemas sofisticados que muchas veces son inaccesibles para el público en general. A continuación, se realiza un estudio de cada uno de estos sistemas clasificándolos en: alarmas eléctricas, alarmas electrónicas, sistemas inmovilizadores, y localizadores.

2.9.1 Alarmas Eléctricas

Es el más simple de los sistemas de alarmas, su función básica es la activación de la bocina del auto al detectar la apertura de la puerta por una persona no autorizada, no impide que el vehículo pueda arrancarse, su costo es relativamente barato ya que no necesita de un módulo electrónico, sino únicamente de un circuito de relés o interruptores.

Este tipo de alarmas no son muy comunes de encontrar debido al desarrollo de las alarmas electrónicas que poseen mayores características, bloquean el arranque del motor y además evitan que el auto descargue su batería al activarse por prolongados periodos, el costo promedio de una alarma eléctrica es de 10 a 20 dólares, sin instalación.

2.9.2 Alarmas Electrónicas

Las alarmas electrónicas poseen componentes electrónicos que permiten el cierre de forma automática de los seguros de las puertas, cortan el arranque del motor y su activación provoca salida sonora de una sirena en el auto. Poseen un módulo que

controla las funciones de la alarma, además incluyen controles remotos, relés automotrices, sensor de puerta, sensores de golpes, anti atraco, botón valet entre otras características.

El módulo electrónico controla a los sensores instalados en el auto, al igual que la sirena, luces y sensores se conectan a la batería del vehículo, cuando detecta alguna variante vulnerada activa las señales visuales y sonoras e impide el arranque del vehículo, también incluyen receptores de radio para control inalámbrico desde el mando. Dependiendo de la cantidad de sensores que maneje y equipo adicional como bloqueo central para apertura y cierre remoto de puertas, calidad de transmisores o mandos de radio (una vía, dos vías, anti escáner, con función de vibración etc.), batería de respaldo en caso de cortar la alimentación del auto, hace que el costo oscile desde los 40 a 180 dólares sin instalación. (Blog A. d., 2013)

Es el sistema de alarma más difundido del mercado, lo que permite que existan copias de mala calidad que se activan cuando no hay intención de robo y producen el efecto de molestar en vez de persuadir, al ser popular da cabida a muchos instaladores, y por consiguiente este tipo de sistemas son muy conocidos por los asaltantes que pueden vulnerarlos con facilidad en pocos minutos.

2.9.3 Sistema de Inmovilizadores

Son dispositivos electrónicos que habilitan o no la puesta en marcha del motor por medio de una computadora que corta los pulsos de la bobina de encendido o pulsos de inyección de combustible de la bomba, se desactiva cuando la llave, mando remoto u otro dispositivo es reconocido por el sistema inmovilizador. Brinda un alto grado de inviolabilidad, generalmente un inmovilizador no ofrece ninguna protección antirrobo visual o audible y requiere la presencia del conductor.

Estos inmovilizadores están presentes en vehículos de gama media a alta, en diversos tipos pero que básicamente su principio de funcionamiento es similar, entre estas versiones de inmovilizadores se encuentran los que funcionan por: comando remoto infrarrojo, teclado numérico, llave transponder (el más utilizado) y por tarjeta codificada.

Los inmovilizadores de comando remoto infrarrojo funciona al emitir una señal que habilita el arranque del motor, también abre y cierra los seguros de las puertas, vienen incorporados dentro del mango de la llave del vehículo o puede ser un control separado, no poseen antena y su unidad receptora de código infrarrojo está ubicada en el habitáculo del auto, el sistema se complementa con la unidad electrónica inmovilizadora que realiza el bloqueo electrónico de la unidad de control o al relé de arranque del motor. (Blog, 2016).

Los inmovilizadores con teclado numérico funcionan al ingresar un código de 4 dígitos cada vez que se intente arrancar el vehículo, se ubica muy cercano al alcance del conductor, al igual que los otros inmovilizadores, éste también bloquea la unidad de control, es más simple que el sistema con comando remoto porque no necesita receptores ni emisores de señales de radiofrecuencia, su desventaja es que se debe ingresar la clave cada vez que se requiera encender el auto. (Blog A. d., 2013)

Los inmovilizadores con tarjeta codificada elimina las llaves de contacto, se insertan en un lector de tarjetas que desbloquean los relés de auto retención para eliminar los distintos sistemas o circuitos del auto, equivale al interruptor de encendido y permite el arranque del vehículo, posee un código variable aleatorio para reducir su clonación, esta tarjeta debe permanecer insertada en el lector, si es retirada se interrumpe el circuito de retención al relé principal y el vehículo activa todos sistemas y circuitos para inmovilizar el auto. En algunos modelos de autos la tarjeta funciona con una pila y el lector la lee sin necesidad de que esta sea introducida al lector (Blog A. d., 2013)

El inmovilizador con llave transponder es el sistema más usado principalmente por su grado de inviolabilidad, este dispositivo que generalmente está en la misma llave del vehículo transmite y recibe las señales de radiofrecuencia RFDI, este chip emite un código RF al momento de accionar el contacto, y es captado por una unidad lectora o antena que se suele ubicar en el conmutador de arranque, este código es enviado a la unidad o módulo inmovilizador que compara con el que tiene memorizado, al ser correctos la unidad autoriza el arranque del motor o caso contrario bloquea el vehículo.

Los transponder operan en diferentes rangos de frecuencias a una distancia de 1 a 15 cm, la tensión requerida para su funcionamiento se crea por inducción de campo magnético variable que circula por la unidad lectora, esta corriente se envía por la unidad del inmovilizador al momento de accionar la llave de contacto. En caso de que este transponder se averíe el motor del vehículo no arrancará.

El módulo o unidad del inmovilizador debe reconocer que se colocó la llave en el switch de arranque, emitir un campo magnético para activar el transponder (emisor de código) de la llave, luego debe recibir el código secreto emitido por la llave, evaluar y dirigir la comunicación serial bi-direccional y finalmente autorizar el encendido del motor del vehículo. (Blog, 2016)

2.9.4 Localizadores

Son dispositivos modernos que permiten localizar un objeto por satélite (GPS: sistema de posicionamiento global), además, pueden brindar funciones de alarma y de inmovilizadores, son equipos de última generación, muy compactos y con varias prestaciones que lo posicionan como el más completo sistema antirrobo del mercado. Estos localizadores avisan de inmediato a través del celular ya sea por mensajes de texto (SMS), llamadas o aplicaciones móviles, de esta forma le permite al usuario tomar acción inmediata.

Entre otras prestaciones y de acuerdo al servicio que se contrate puede incluir servicios de control inteligente de velocidad, control de consumo de combustible, comandos remotos de bloqueo, alertas de incidencias, apertura remota, recorrido y mantenimiento del vehículo, monitoreo y control de ruta en tiempo real, reportes, historial y estadísticas entre otras.

Un sistema localizador básico sin renta mensual tiene un costo de 70 dólares no incluida su instalación, mientras que para sistemas más completos sobrepasan los 200 dólares y son recomendados para flotas de vehículos que necesitan mayor control de sus unidades. Además, necesita de una SIM CARD con plan de datos económicos de cualquier operadora móvil (Smartech, 2016)

3. BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO

3.1 Problema principal

De acuerdo a las estadísticas que maneja la Policía Nacional en su publicación del 11 de abril del 2016 (<http://www.policiaecuador.gob.ec/evite-robo-de-su-vehiculo/>) y el Observatorio de Seguridad Ciudadana del DMQ manifiestan que cerca del 80% de incidencia de robos de vehículos se da en la vía pública sobre todo en los exteriores de los domicilios, en sitios de gran afluencia pública como conciertos, estadios, balnearios, etc. Según datos policiales, bandas actuales operan con ayuda de personas con conocimiento de sistemas electrónicos para lograr inhabilitar los sistemas de seguridad tradicionales tales como la alarma electrónica.

3.2 Problemas secundarios

- Sistemas tradicionales como la alarma no garantizan la función de proteger los bienes de la delincuencia común, por esta razón, se plantea un prototipo sistema de bloqueo que combine varios dispositivos con mayor tecnología.
- Los sistemas de bloqueo con nueva tecnología son limitados a los autos de media a alta gama.

3.2.1 Por qué y para qué de los objetivos

Elaborar un prototipo de sistema de bloqueo y ser una alternativa a los sistemas planteados anteriormente, que combina características de alarma, de inmovilizador y de localizador empleando tecnología innovadora que brinde la seguridad necesaria al vehículo por medio de tecnología RFID, pantalla táctil resistiva y envío de SMS a un precio promedio.

3.2.2 Hipótesis o idea a defender

La elaboración del prototipo de bloqueo para vehículos mediante el uso de tecnología RFID, pantalla táctil resistiva, y envío de mensajes de texto SMS, ofrecerá un mejor control al acceso del encendido del automotor, adaptable a cualquier tipo de automóvil (a excepción de vehículos híbridos o eléctricos), a un costo accesible, incrementando el nivel de tranquilidad y seguridad del usuario.

3.2.3 Métodos utilizados para el desarrollo del proyecto

El presente proyecto de investigación se centra en el estudio de los dispositivos existentes en el mercado para el bloqueo de vehículos, sus principales elementos, características y vulnerabilidades mediante el método analítico que permitirá comparar los diferentes sistemas que contribuyan a mejorar el desarrollo en el sector automovilístico incluyendo nuevas tecnologías para adaptarlas al prototipo.

El prototipo presentado se basó en la investigación Bibliográfica-documental por la necesidad de recurrir a diferentes fuentes de información como manuales, páginas webs, blogs, catálogos y libros a fin de recolectar información profunda respecto a los sistemas de bloqueo y seguridad vehicular existentes que ayudaron al desarrollo de proyecto.

La investigación realizada fue a nivel exploratorio al permitir conocer las características actuales de los sistemas de seguridad automotriz, también a nivel descriptivo donde se especifica los hechos y particularidades reales de los mismos para luego comparar con los resultados obtenidos, finalmente a nivel explicativo donde se obtendrá el producto final.

4. PRESENTACIÓN DE LOS RESULTADOS

Este proyecto usa una placa Arduino Uno como módulo o sistema central, un lector de tecnología RFID (Shield RC522) para primera autenticación del sistema, una pantalla táctil resistiva para introducción de clave, sensor pulsador de puerta de vehículo como condición de seguridad y envío de SMS mediante un módulo sim900 para información del usuario. En el diagrama de bloques de la figura 10 se observa la estructura de los elementos que conforman cada etapa y componente del sistema.

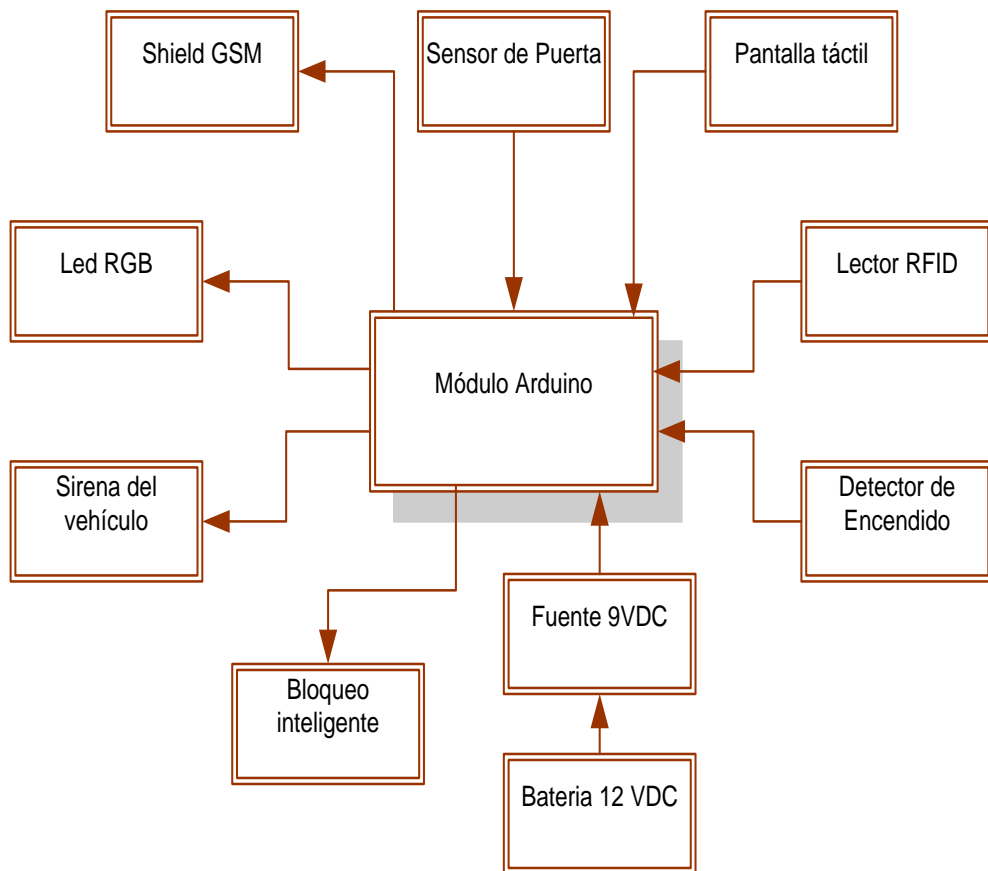


Figura 10 Estructura en bloque del sistema

Fuente: Autor - 2016

Se utiliza el módulo Arduino Uno porque los shield o módulos utilizados son compatibles eléctricamente y a nivel de hardware para ser empleados sin inconveniente, además, como se observa en la figura 11 dispone de memoria necesaria para que el sistema no este sobredimensionado. Adicional son los Arduino de mayor demanda, muy comerciales y con muchos proyectos desarrollados en la web.





	Arduino Uno	Arduino Mega2560	Arduino Leonardo	Arduino Due	Arduino ADK	Arduino Nano	Arduino Pro Mini	Arduino Esplora
								
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32 K (0,5 K usado pelo bootloader)	256 K (8 K usados pelo bootloader)	32 K (4 K usados pelo bootloader)	512 K disponível para aplicações	256 K (8 K usados pelo bootloader)	16 K (ATmega168) ou 32K (ATmega328), 2 K usados pelo bootloader	16 K (2k usados pelo bootloader)	32 K (4 K usados pelo bootloader)
Clock	16 Mhz	16 Mhz	16 Mhz	84 Mhz	16 Mhz	16 Mhz	8 Mhz (modelo 3.3v) ou 16 Mhz (modelo 5v)	16 Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial / Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5v	5v	5v	3.3v	5v	5v	3.3v ou 5v, dependendo do modelo	5v
Corrente máxima portas E/S	40 mA	40 mA	40 mA	130 mA	40 mA	40 mA	40 mA	-
Alimentação	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	3.35 - 12 V (modelo 3.3v), ou 5 - 12 V (modelo 5v)	5v

Figura 11 Comparativo de placas Arduino

Fuente: learn.adafruit.com - 2015

El envío de mensajes de texto SMS se realiza por medio del módulo GSM SIM900 (QUAD-BAND- bandas 850/900/1800/1900 MHz) permite trabajar con cualquier operadora de celular, en este caso se usó un chip de la operadora Movistar, permite enviar 160 caracteres de los cuales solo se utilizan un máximo de 64 caracteres. no es necesario altas velocidades de transmisión. Este módulo se observa en la figura 12.



Figura 12 Módulo SIM900

Fuente: Autor - 2016

Para el caso de la autenticación del usuario se utiliza la tecnología RFID con tags o tarjetas pasivas (de pocos centímetros de alcance), mediante la emisión de ondas electromagnéticas desde el módulo lector que alimenta un circuito de memoria interno, por medio de RF(radiofrecuencia) valida o autentifica al usuario del sistema. Es un sistema de difícil clonación, los tags no necesitan de batería por lo que pueden durar muchos años, son robustas y no se afectan por otros campos magnéticos, y su tecnología está en desarrollo para muchas aplicaciones. El módulo elegido es el RFDI-RC522 como el que se observa en la figura 13, que es compatible con la tecnología Arduino.



Figura 13 Módulo RFID-RC522

Fuente: Autor - 2016

Para el sensor de puerta se escogió un interruptor tipo pin que interrumpe/permite el paso de la corriente eléctrica al abrir o cerrar la puerta del prototipo, para este caso el interruptor normalmente está abierto (puerta cerrada), cerrando el circuito al abrir la puerta, esta señal llega al módulo central donde decide la operación a efectuar. En la figura 14 se puede observar el modelo de pin interruptor modelo P-7R usado en el proyecto.



Figura 14 Interruptor tipo pin

Fuente: Autor - 2016

La pantalla táctil escogida para el proyecto es una malla resistiva de 5,7 pulgadas (113x72mm) modelo AH-2365 con bus de 4 hilos, estos touch están probados en varios proyectos de ingeniería a nivel local y además por ser accesibles en las electrónicas a nivel nacional. Esta pantalla fue calibrada para reconocer 5 puntos de presión que serán reconocidas como “teclas”, distribuidas como se observa en la figura 15, una tecla en cada esquina y una tecla central conforman el teclado de este touch resistivo.



Figura 15 Pantalla Resistiva de 4 hilos / 5,7 pulgadas.

Fuente: Autor - 2016

4.1 Diseño del Sistema

Una vez descrito los componentes principales del sistema de bloqueo, se procede a la etapa de diseño del prototipo, para ello se describen 3 etapas y los elementos que intervienen, estas son: etapa de censado, etapa de control y la etapa de actuación, como observa en la figura 16:

- Etapa de censado: consta del lector RFID, el sensor de estado de la puerta, clave o patrón ingresado a la pantalla táctil resistiva y el sensor de intento de encendido.
- Etapa de control: mediante el Módulo Arduino Uno con el microcontrolador Atmega 328p.
- Etapa de actuación: constituido por un LED RGB indicador, módulo Sim900, una salida en relé para alarma sonora, una salida en relé para bloqueo inteligente.

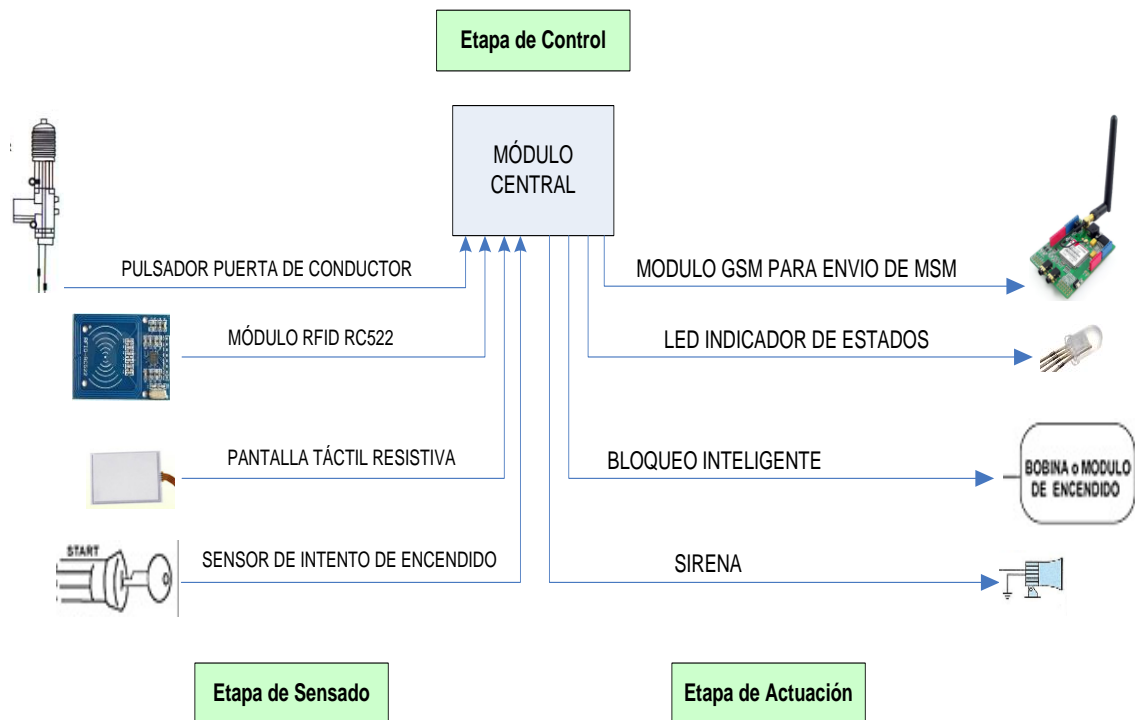


Figura 16 Diagrama de bloques del Hardware del sistema

Fuente: Autor - 2016

4.2 Diagrama de flujo

Los siguientes diagramas ilustran y explican los diferentes modos de funcionamiento del sistema de acuerdo al evento o circunstancia en las que el usuario se encuentre. Para facilitar su comprensión se dividen en 4 subsistemas: modo normal, modo Mantenimiento, modo Antiatraco y los modos de reinicio. A continuación, se explica cada uno de lo señalado.

4.2.1 Modo normal de funcionamiento

Se llama modo normal de funcionamiento cuando el usuario sigue los pasos en el orden correcto, es decir, el sistema no detecta ninguna anomalía en el procedimiento desde el momento en que el usuario acerca su tag RFID hasta que enciende el vehículo de forma normal, tal como lo describe la figura 17. El proceso a seguir en el modo normal de operación es el siguiente:

- El usuario debe acercarse al prototipo de forma que este sea detectado por el lector (distancia de lectura aproximadamente 5 cm).

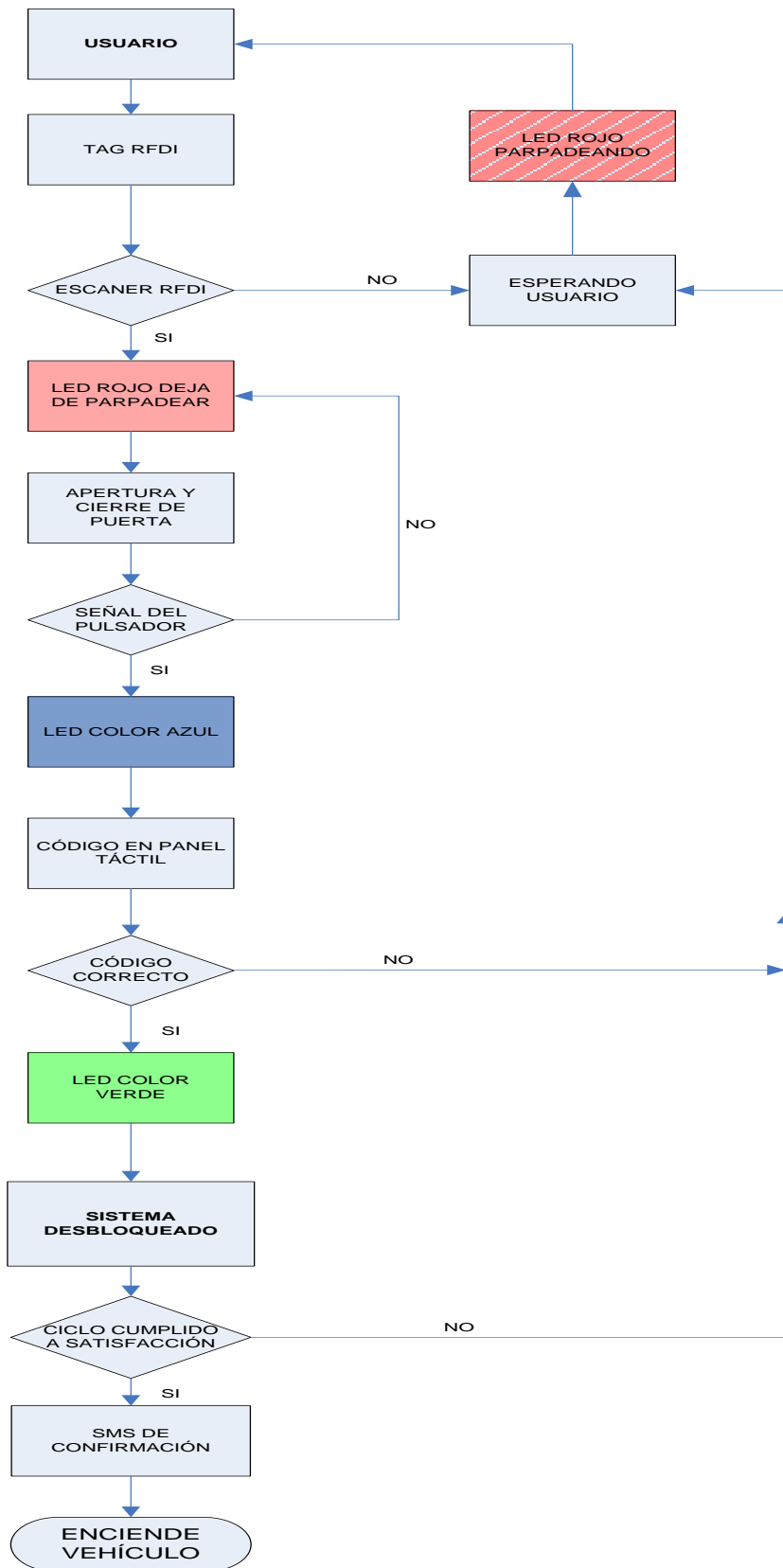


Figura 17 Diagrama de flujo de Software

Fuente: Autor – 2016

- El usuario abre la puerta del vehículo, con la llave o con su control de radiofrecuencia, el sistema detecta la apertura y cierre de la puerta mediante el pulsador.
- Al ingresar al vehículo, debe colocar un patrón o clave en la pantalla táctil resistiva, la pantalla táctil tiene 4 cuadrantes para configurar los códigos, cada cuadrante equivale a un dígito, en este caso el sistema se desbloquea con un patrón de 4 dígitos (patrón de prueba para desbloqueo de sistema: 1-2-3-4). Adicional, en el centro de la pantalla táctil se diseñó y programó un botón de activación/reinicio del sistema.
- Al cumplir con estos 3 requisitos el bloqueo inteligente (sistema de encendido y arranque) del vehículo puede encender.

Estos 3 requisitos son visualizados mediante un LED tricolor o RGB, el color rojo parpadeante indica que el sistema está activo, al cumplir el primer requisito (RFID) este deja de parpadear, luego al abrir la puerta del vehículo este cambia a color azul en espera del código en el panel táctil, el LED RGB se pondrá verde una vez cumplido los 3 requisitos.

Adicional, mediante el módulo SIM900 se envía los mensajes de texto al usuario de acuerdo a cada situación que se presente, para el nodo normal de funcionamiento el mensaje enviado al final del ciclo es: "Auto desbloqueado, puede iniciar su viaje". Al detectar alguna violación en el ciclo normal el sistema envía los mensajes siguientes: "Intento de encendido del vehículo" sin haber desbloqueado el sistema, "sensor de puerta detectado" cuando no se cierra la puerta del conductor, "clave incorrecta o no ingresada" por ausencia o mal ingreso de la clave en el touch resistivo.

4.2.2 Modo Mantenimiento

Para la opción del modo de servicio o mantenimiento, primero se debe realizar la lectura del tag RFID, luego abrir y cerrar la puerta, a continuación, abrir nuevamente la puerta e ingresar un código o patrón adicional (es el mismo código de usuario digitado al revés) cumplido este ciclo el sistema entra a modo de mantenimiento, en este caso el LED RGB se apaga por completo, el sistema se anula y el vehículo podrá encenderse sin ninguna restricción, es ideal cuando necesite servicio técnico, de mantenimiento o lavado del automotor. En la figura 18 se observa este proceso. Adicional el sistema enviara un mensaje de texto al usuario que le indica que su vehículo ingreso al "modo Mantenimiento".

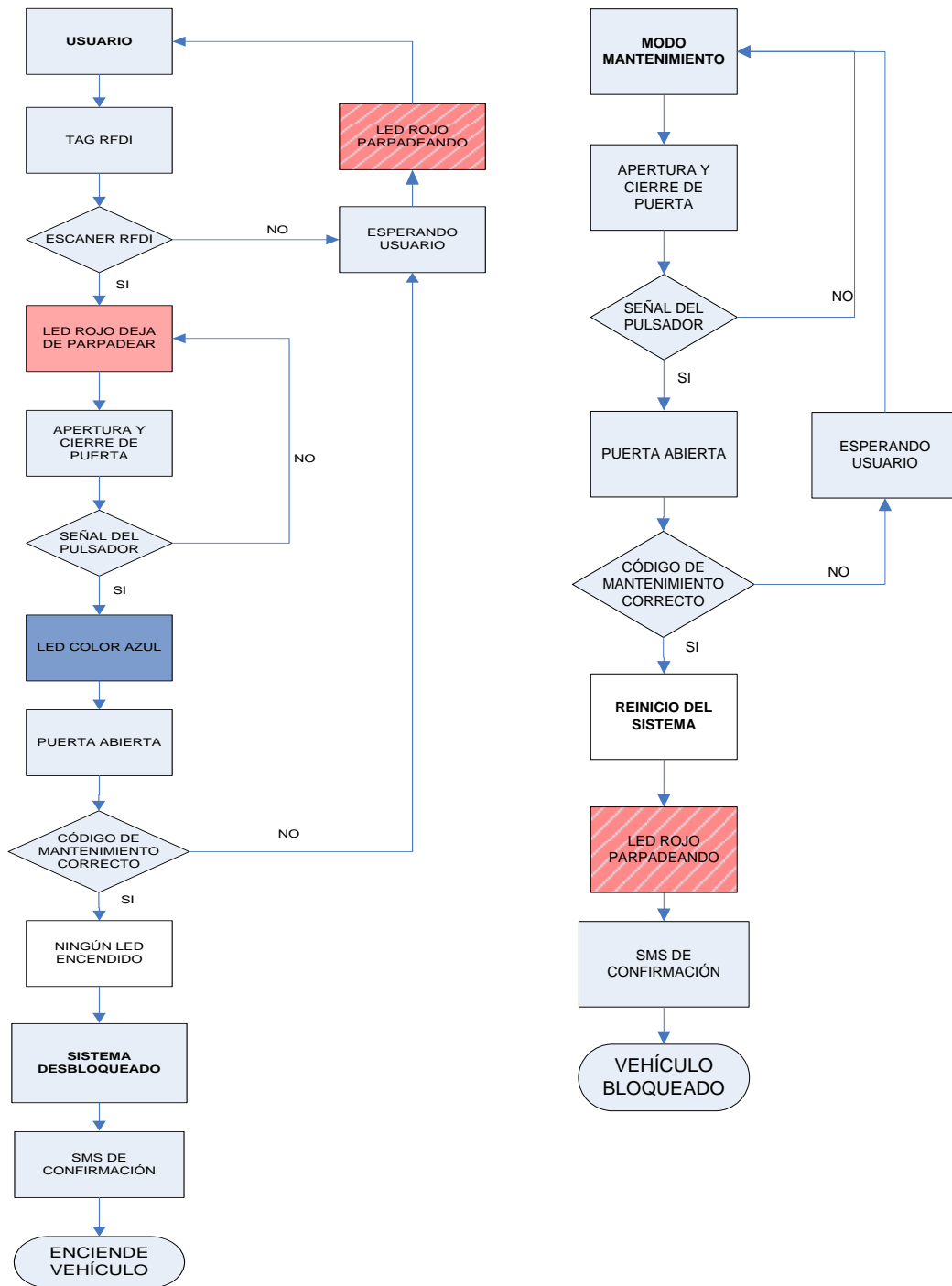


Figura 18 Diagrama de flujo Ingreso/Salida del modo Mantenimiento

Fuente: Autor - 2016

4.2.3 Modo Antiatraco

El modo antiatraco o secuestro se activa cuando el vehículo está encendido y se abre la puerta del conductor, en este caso, se debe ingresar la clave de desbloqueo en el lapso de 60 segundos, el LED RGB parpadea en color verde, transcurrido el tiempo de espera el sistema corta el paso de gasolina y enciende la alarma del vehículo. De igual forma, el sistema envía un SMS con la respectiva advertencia “Se ha detectado secuestro, auto bloqueado”. En la figura 19 puede observar este proceso.

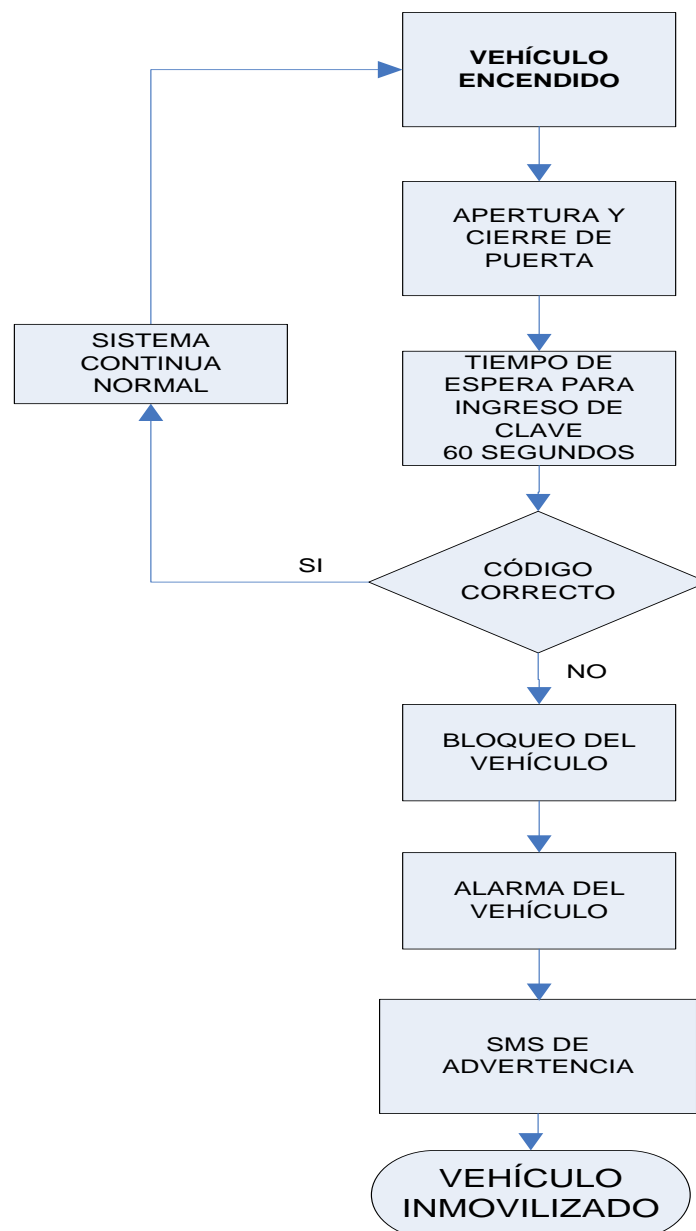


Figura 19 Diagrama de flujo Modo Antiatraco/Secuestro

Fuente: Autor – 2016

4.2.4 Modo de Reinicio

El modo de Reinicio del sistema se aplica cuando se detecta un error en el proceso, el módulo bloquea el sistema y la alarma del vehículo se activa, esto sucede cuando se intenta encender el vehículo sin cumplir las verificaciones del sistema, o cuando se necesita desactivar el modo antiatraco, o para salir del modo de mantenimiento al modo normal, para estos casos existen las 3 alternativas de reinicio, mismas que se observan en la figura 20 y son:

1. Con la puerta abierta, presionar el botón central del panel táctil hasta que el LED RGB se encienda por 2 ocasiones en color blanco.
2. A través de un código celular que deberá ser enviado por el usuario al chip del vehículo, en este caso es el código “e10”.
3. Mediante un pulsador que se encuentra en el módulo principal.

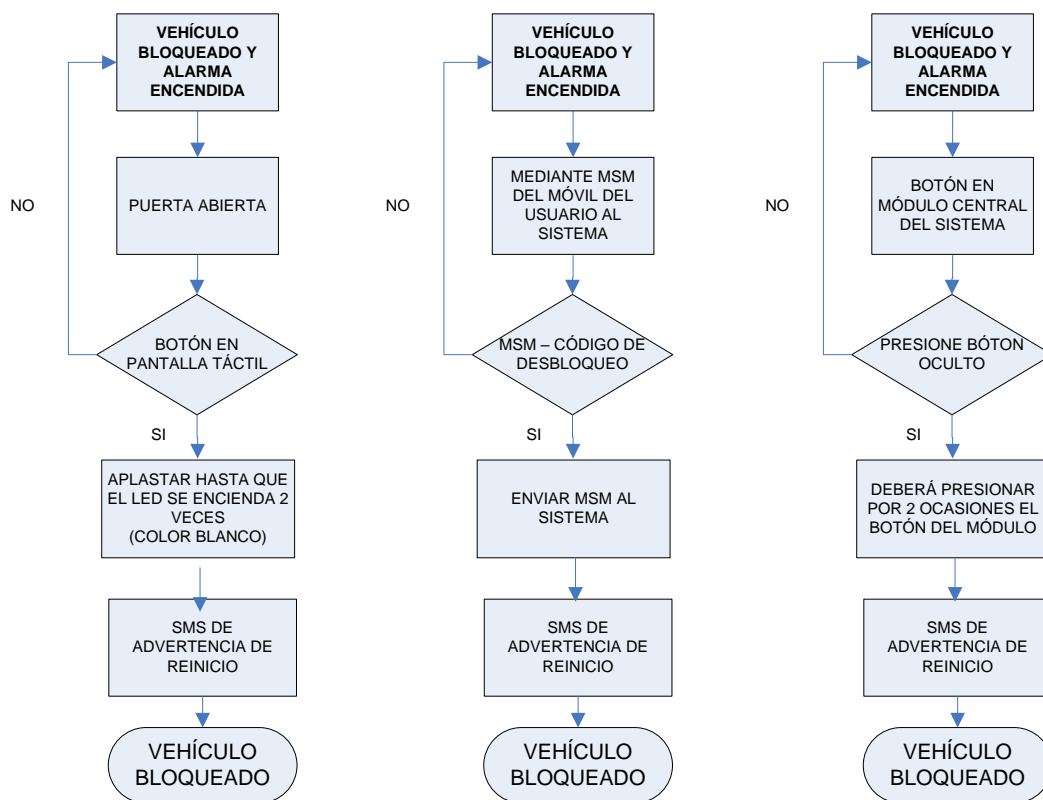


Figura 20 Diagrama de flujo Modos de reinicio

Fuente: Autor – 2016

4.3 Esquema del circuito

A continuación, en la tabla 1, se observa la distribución de pines del módulo Arduino Uno y su función en la placa base, el voltaje de alimentación que brinda la batería del vehículo es de 12 VDC, a través de un circuito regulador LM7809 se establece la alimentación del Arduino Uno al pin VIN con 9 VDC, la pantalla táctil resistiva usa 4 pines análogos, el módulo RFID usa 8 pines (7 con función) , los circuitos de activación de alarma, detección de intento de encendido, estado de puerta, bloqueo inteligente usan un pin cada uno, el control del LED RGB usa 3 pines.

Tabla 1 Pines de entrada/salida en el módulo Arduino Uno

PIN	FUNCIÓN	DESCRIPCIÓN
0	Señal para buzzer	Alerta sonora para la pantalla táctil
1	Señal de salida para activación del sistema de arranque	Se activa cuando todas las condiciones se ingresan correctamente
2 - 3 - 4	Señal para activación del LED RGB	Se activa cada color de acuerdo a las condiciones cumplidas
5	Señal de salida para activación de la sirena	Se activa cuando todas las condiciones no se ingresan correctamente
6	Señal de estado de puerta del vehículo	Señal de entrada de estado de puerta del vehículo ABIERTO-CERRADO
7- 8 - 9	NO CONECTADOS	SIN USO
10 - 11 - 12 - 13	Señal de entrada lector RFID	Señal de entrada del lector RFID mediante los tags
GND - AREF - NC - NC	NO CONECTADOS	SIN USO
A0 - A1 - A2 - A3	Señal de entrada panel táctil	Señal de entrada de la pantalla táctil mediante código
A4	RESET	RST para módulo RFID
A5	Sensor de encendido	Señal de entrada en caso de intento de encender el vehículo sin cumplir con las condiciones
VIN	VCC	9 VDC
3.3 V	Alimentación lector RFID	Voltaje para funcionamiento del lector RFID
RESET- OIREF - NC	NO CONECTADOS	SIN USO

Fuente: Autor - 2016

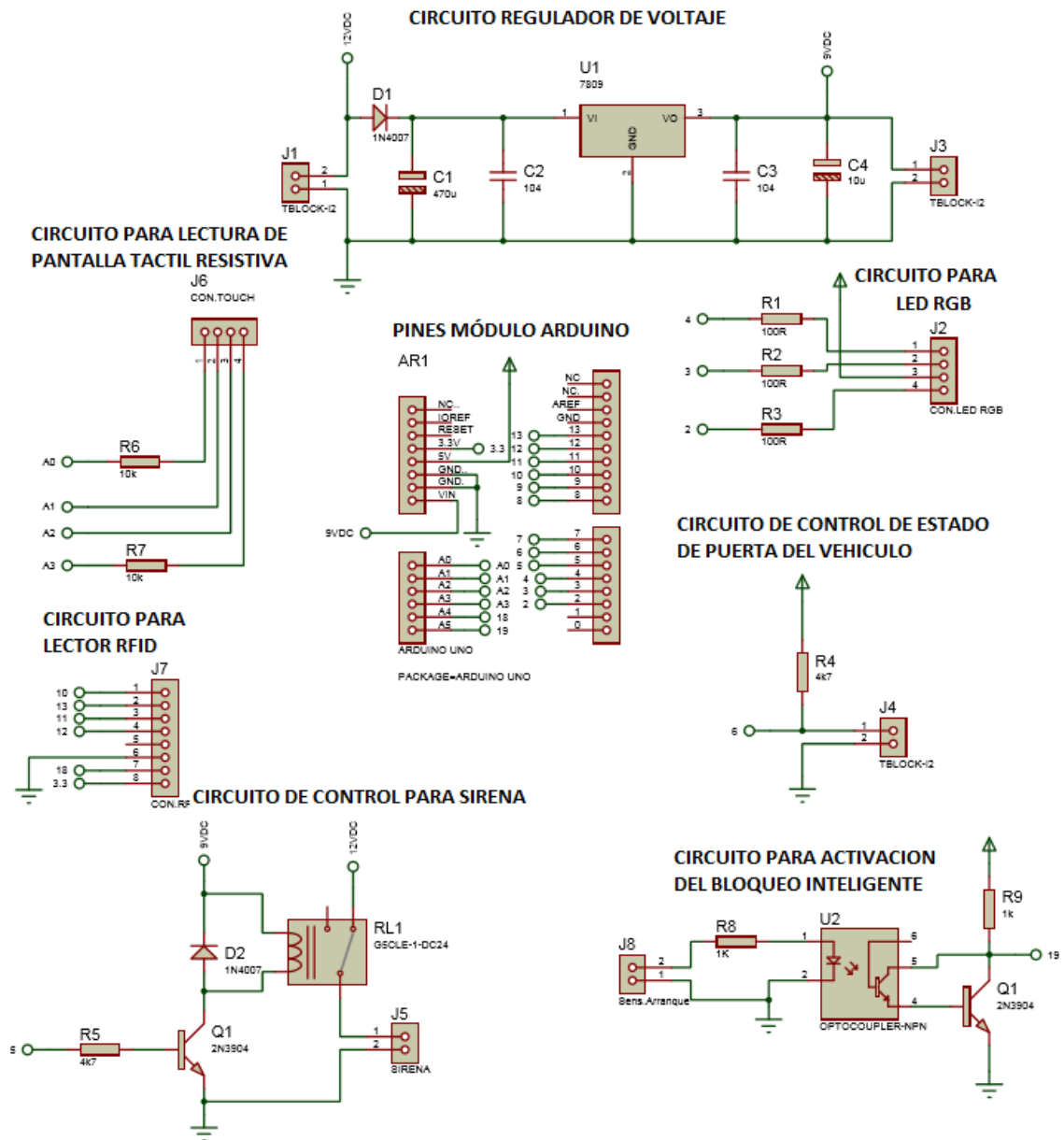


Figura 21 Diagrama del circuito

Fuente: Autor - 2016

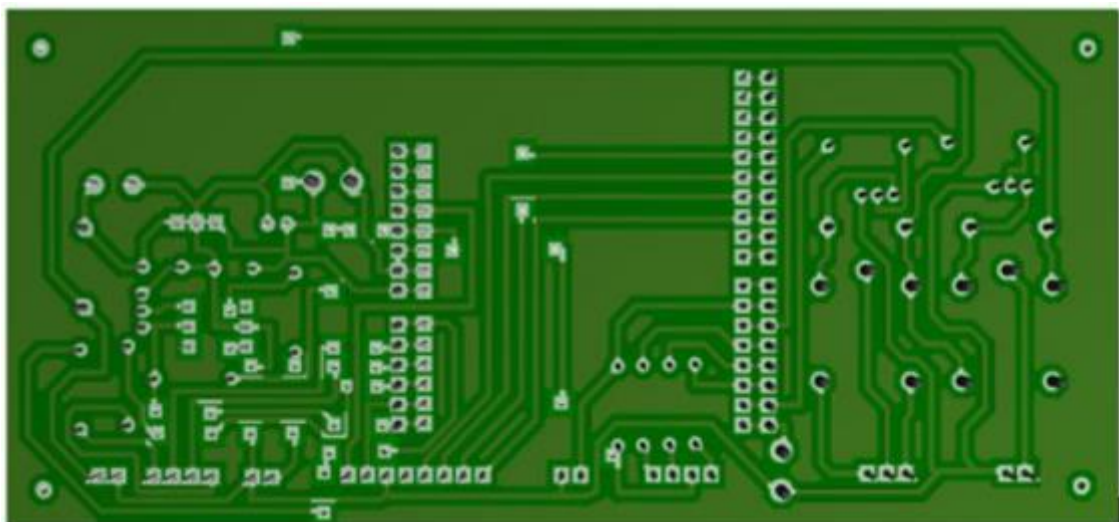
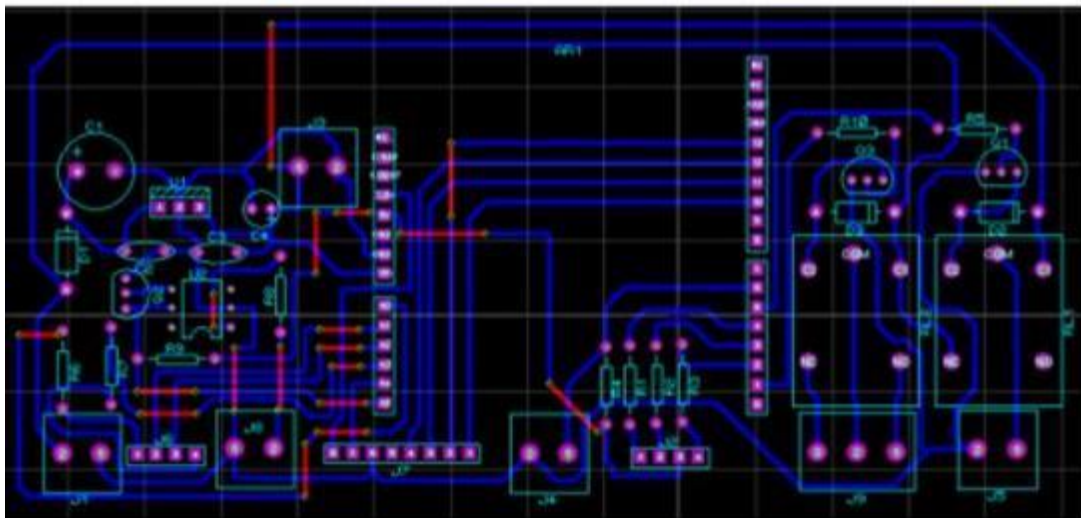
4.4 Implementación

Una vez analizado todas las especificaciones y necesidades del sistema, se procede a la implementación e interconexión de los elementos que conforman el módulo central, para esto se usa el programa Proteus 7.7 a través de ISIS (Sistema de Enrutado de Esquemas Inteligente) para el plano eléctrico del circuito, en la figura 21 se observa el diagrama general de los dispositivos y elementos necesarios para el correcto funcionamiento del módulo. Para una mejor comprensión se observan los diferentes circuitos que conforman el sistema, estos son: circuito regulador de voltaje, circuito para

lectura de la pantalla resistiva, circuito para lector RFID, circuito para activación del led RGB, circuito para control de sirena, circuito sensor de estado de la puerta, circuito para activación por relé del bloqueo inteligente y el circuito del módulo Arduino.

4.4.1 Diseño de placa base

A continuación, se presenta el diseño y fabricación de la placa base donde se realiza el enrutamiento de pistas de todos los dispositivos y elementos que conforman el módulo del sistema, para esto se usa el programa Proteus 7,7 a través de ARES (Software de Edición y Ruteo Avanzado), así mismo se presenta la simulación 3D disponible en el software, esto se observa en la figura 22.



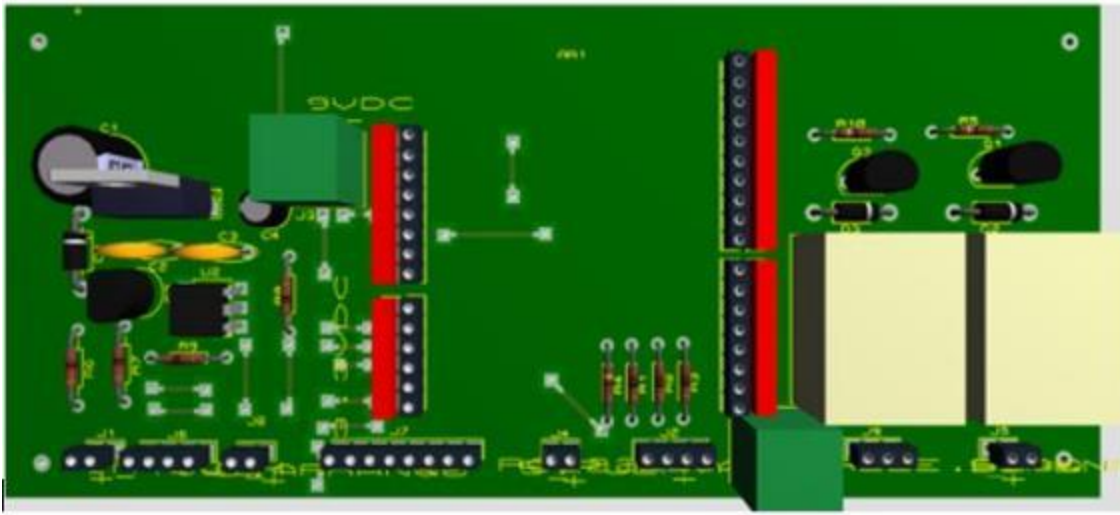
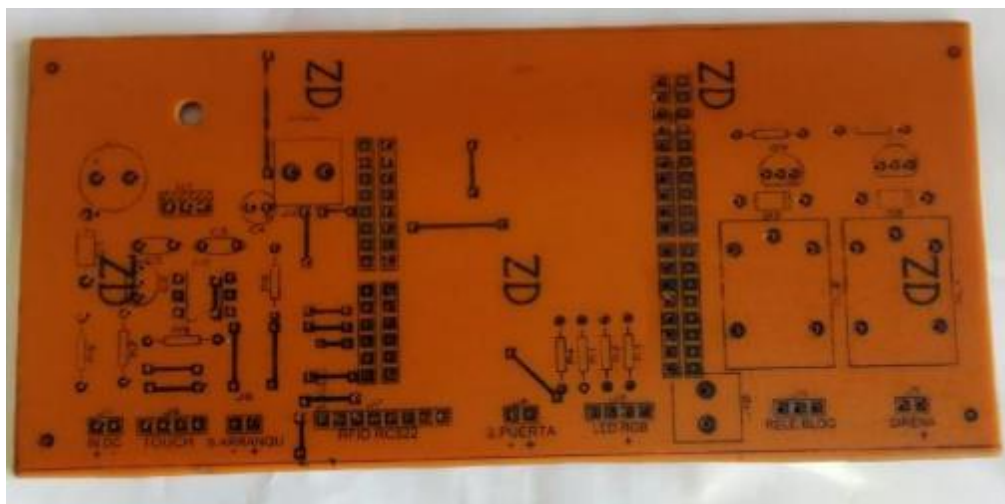


Figura 22 Diseño de placa base y vista 3D

Fuente: Autor – 2016

4.4.2 Fabricación y ensamblaje de elementos

A continuación, se crea la placa de circuito impreso PCB, esto se realiza mediante la impresión láser del circuito en hojas de transferencia, calor mediante una plancha y el uso de cloruro férrico como principales elementos en su elaboración, luego se procede a la implementación de todos los elementos electrónicos de la placa base, como observa en la figura 23.



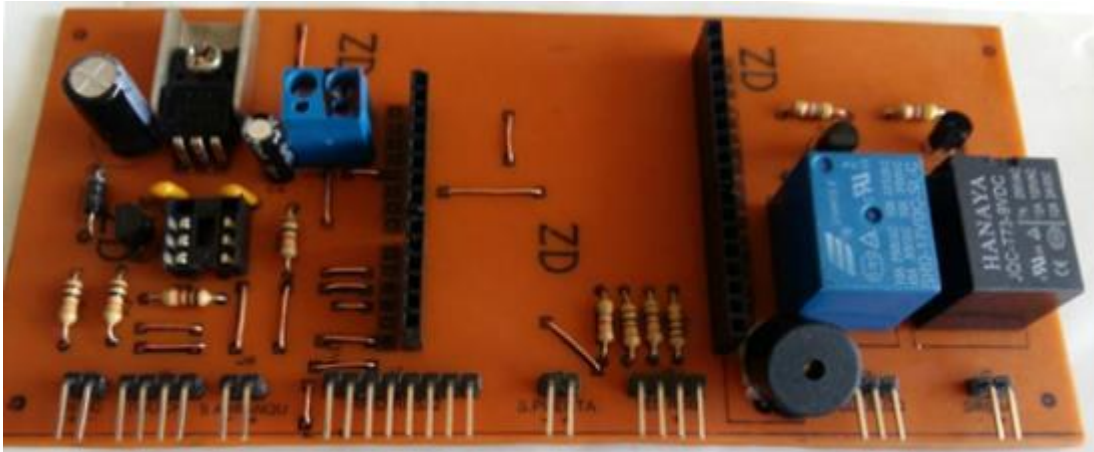


Figura 23 Fabricación y ensamblaje de elementos de placa base

Fuente: Autor - 2016

4.4.3 Interconexión Placa base con módulo Arduino Uno

Después de colocar los elementos de la placa base, se conecta el módulo Arduino Uno sobre el zócalo diseñado para esto, tal como se observa en la figura 24, los pines del Arduino y la placa base se deben alinear con mucho cuidado para evitar doblar o conectar de forma incorrecta, una inadecuada conexión puede dañar el módulo Arduino Uno.



Figura 24 Interconexión Placa base con módulo Arduino Uno

Fuente: Autor - 2016

4.4.4 Interconexión módulo Arduino Uno con módulo Sim900

Sobre el módulo Arduino se acopla el shield SIM900 junto con la SimCard, así mismo se debe evitar que los pines del módulo se doblen, verifique la correcta alineación de los pines de comunicación del módulo SIM900 al módulo Arduino como se observa en

la figura 25. Es importante revisar la información del fabricante del shield Sim900 adquirido, existen pequeñas configuraciones que varían entre ellos, especialmente en los pines TX y RX, en este caso están asignados a los pines 7 y 8 respectivamente, adicional el autoarranque necesita unirse por suelda en el sector R13 o JP, según cada fabricante.

Respecto al chip instalado en el módulo SIM900 se escogió a la operadora Movistar, sencillamente por efectos prácticos, el sector norte de Quito (San José de Morán) donde se realizaron las pruebas no tuvo la respuesta adecuada de la operadora Claro, por otro lado, la operadora CNT presenta caídas de red constantes en el sector, a pesar de lo mencionado, se recomienda el uso de un chip Claro, por su principal ventaja en cobertura a nivel nacional.



Figura 25 Interconexión Placa base con módulo Arduino Uno

Fuente: Autor - 2016

Para realizar las pruebas se construyó una caja de control provisional como la observada en la figura 26, con la que se ejecutan todas las pruebas del dispositivo. Una vez cumplidas a satisfacción se procede a la implementación del sistema dentro del prototipo a fin de cerciorar el correcto funcionamiento



Figura 26 Interconexión Arduino Uno - Sim900 y placa de control

Fuente: Autor - 2016

Por último, se presenta en la tabla 2, la interacción entre los pines del módulo lector RFID al módulo Arduino Uno, en este caso se usa el protocolo SPI para la comunicación entre ambos dispositivos. El Módulo RFID-RC522 utiliza 3.3 voltios como alimentación. El RC522 maneja un sistema avanzado de modulación y demodulación para todo tipo de dispositivos pasivos de 13.56Mhz. Las tarjetas o tags que vienen junto al módulo RFID tienen 64 bloques de memoria (0-63) donde se hace lectura y/o escritura. Cada bloque de memoria tiene la capacidad de almacenamiento hasta 16 Bytes. El número de serie consta de 5 valores hexadecimales únicos de cada tag.

Tabla 2 Conexión de pines del entre el módulo RFID y Arduino Uno

Conexión del entre el módulo RFID y Arduino	
Módulo RC522	Arduino Uno
SDA (SS)	10
SCK	13
MOSI	11
MISO	12
IRQ	No conectado
GND	GND
RST	18
3.3V	3.3V

Fuente: Autor – 2016

4.5 Pruebas de comprobación de dispositivos:

Se realizan las pruebas básicas del prototipo para comprobar y verificar que todos los elementos o dispositivos estén correctamente conectados y respondan de manera correcta a las funciones que fueron diseñados, esta comprobación se realiza para descartar errores de ensamblaje de los elementos que conforman el dispositivo y se detalla en la tabla 3.

Tabla 3 Pruebas de encendido

PROCEDIMIENTO	Número de Pruebas		50	
	PORCENTAJES	Correcto	Incorrecto	
Encendido del arduino uno	100%	0%	50	0
LED RGB intermitente Rojo-Azul-Verde	100%	0%	50	0
Sensor de apertura de puerta	100%	0%	50	0
Activación del panel táctil	92%	8%	46	4
Activación del Sim900	100%	0%	50	0
Envío de mensajes de texto (SMS)	94%	6%	47	3
Salida de relé a sirena	100%	0%	50	0
Salida en relé para activación de encendido	100%	0%	50	0
Sensor de intento de encendido	100%	0%	50	0
Desbloques (SMS, pulsador, reseteo)	94%	6%	47	3
Ingreso/salida de mantenimiento	100%	0%	50	0
Modo antisequestro	100%	0%	50	0

Fuente: Autor – 2016

Del total de 50 pruebas efectuadas, la mayoría de las pruebas fueron al 100% fiables como se muestra en la tabla 3, a excepción de las pruebas que comprometen al envío de SMS cuyo resultado fue del 94% (3 errores), este resultado se produjo error debido a la cobertura de la operadora. Otro resultado de fiabilidad del 92% (4 errores) fue producido por el mal ingreso de la clave en la pantalla resistiva, esta pantalla debe ser presionada en los lugares que fueron calibrados para reconocer cada dígito. El diagrama de Pareto de la figura 27 nos indica la tendencia de los errores de mayor frecuencia en el prototipo, esta información servirá para trabajar en mejoras futuras al dispositivo utilizando nueva tecnología de pantallas táctiles y usando operadoras con mayor cobertura.

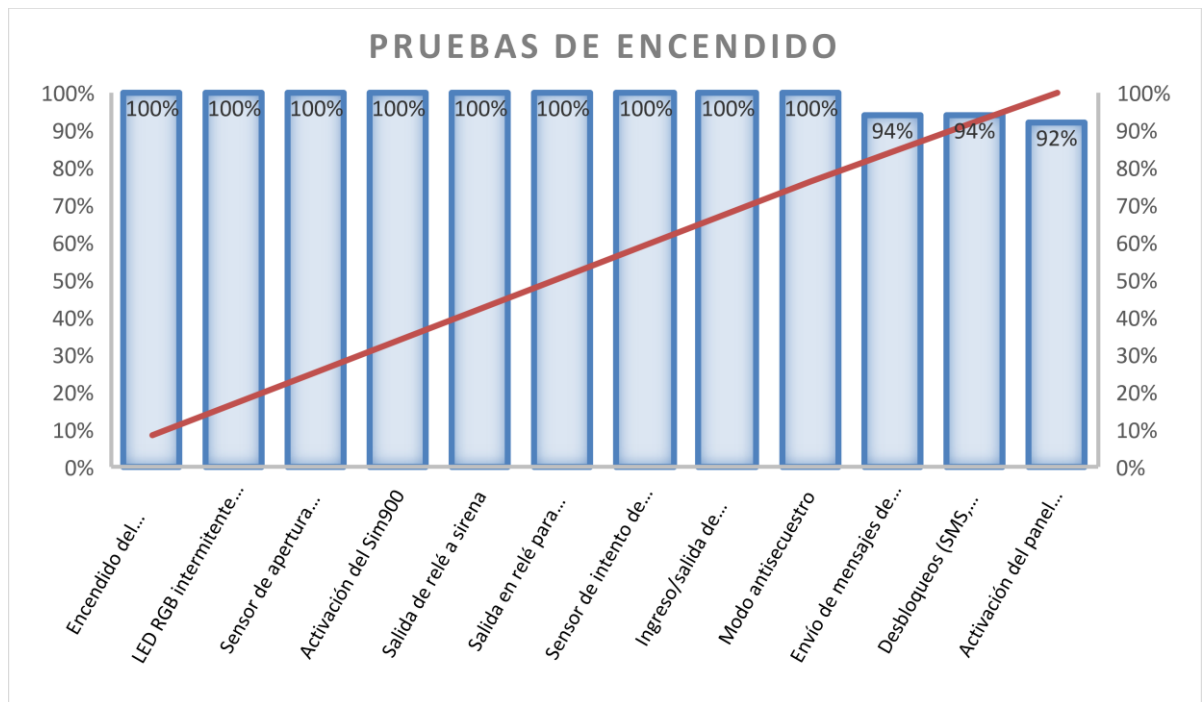


Figura 27 Diagrama de Pareto

Fuente: Autor - 2016

Mediante la fórmula para el cálculo de Defectos (DPU), que es el número de defectos en una muestra dividido entre el número de unidades de la muestra, la fiabilidad del sistema viene dado por: $DPU = (\# \text{ defectos}) / \# \text{muestras}$; aplicándolo a nuestra tabla 3, se tiene:

$$DPU = (4+3+3)/50$$

$$DPU = 20\%$$

Así mismo, aplicando el método de cálculo para fiabilidad en todo el sistema (R_s), mediante la fórmula: $R_s = R_1 \times R_2 \times R_3 \times \dots \times R_n$, donde R_1 es la fiabilidad del componente 1, R_2 del componente 2 y así sucesivamente, en este caso:

$$R_s = 100 * 100 * 100 * 100 * 100 * 100 * 100 * 100 * 100 * 100 * 94 * 92 * 92 \text{ (\%)}$$

$$R_s = 81\%$$

Esto quiere decir que, de cada 10 arranques del sistema, 2 tendrán algún defecto y no se completan en su totalidad, ya sea por mala digitalización del teclado o por deficiente cobertura en la operadora Movistar, la fiabilidad total del sistema es del 81%, es decir que 8 de cada 10 arranques se cumplirán a satisfacción.

4.6 Pruebas de operatividad

Una vez realizadas las pruebas de comprobación, se realizan las siguientes pruebas de operatividad y funcionamiento del sistema, estas pruebas ayudan a descartar errores de operación y proveen una idea real y final del sistema, una vez realizadas estas pruebas a satisfacción, se procederá al ensamblaje final del prototipo.

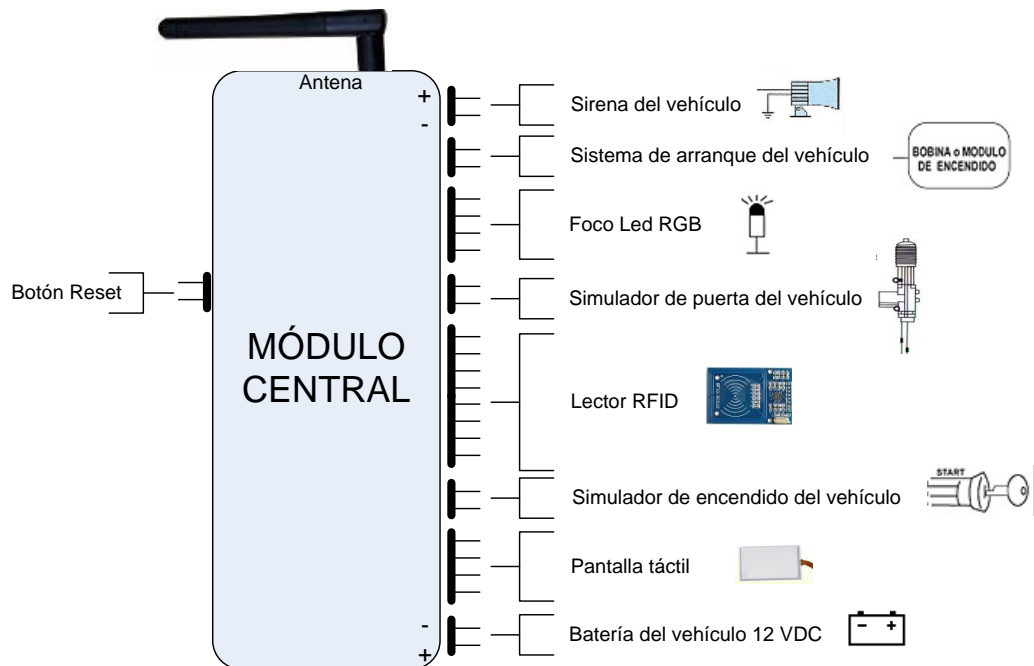


Figura 28 Conexión de dispositivos al módulo central

Fuente: Autor - 2016

1. Todos los dispositivos se conectan al módulo o placa central tal como se indica en la figura 28, luego de encender la fuente de poder del sistema y arrancar el prototipo con una fuente de 12 Voltios - 2 Amperios (tiempo de primer arranque 10 segundos), se comprueba que el Arduino Uno arranca correctamente, esto se verifica por medio del encendido de LED de estado propio del módulo Arduino. De igual forma, se encenderá el módulo SIM900. El LED RGB indicara su estado al parpadear constantemente en color Rojo en espera de la lectura del módulo RFID. La figura 29 brinda una imagen de los elementos del sistema.



Figura 29 Arranque del sistema y sus componentes

Fuente: Autor – 2016

2. El sistema permanece en espera de lectura de la tarjeta RFID mediante su módulo RC522, la tarjeta y el tag con acceso son leídos y confirmados correctamente por el lector RFID, se ratifica por medio del LED RGB que deja de parpadear en color rojo, también se prueba con una tarjeta sin acceso, el lector no la reconoce y el sistema no continúa a la siguiente etapa, el LED RGB continúa parpadeando en espera de una tarjeta con acceso. Este proceso se observa en la figura 30.



Figura 30 Autentificación de tarjeta o tag RFID

Fuente: Autor - 2016

Para este proyecto se habilitó una tarjeta y un tag RFID los mismos que fueron leídos por el módulo Arduino Uno según la guía que se encuentra en el sitio web <http://hetpro-store.com/TUTORIALES/modulo-lector-rfid-rc522-rf-con-arduino/>, luego se confirmó

con un celular Samsung, este celular de alta gama tiene la aplicación que les permite leer este tipo de tarjetas RFID/NFC. En la tabla 4, se presentan los datos obtenidos.

Tabla 4 Lectura hexadecimal RFID

	HEXADECIMAL	DECIMAL
TARJETA RFID 1	83	131
	9E	158
	AC	172
	AB	171
	1A	26
TARJETA RFID 2 (llavero)	53	83
	3E	62
	14	20
	BC	188
	C5	197

Fuente: Autor - 2016

- Mediante un switch se simuló la apertura/cierre de la puerta del vehículo como se observa en la figura 31, esta etapa se comprueba por medio del LED RGB, este cambia al color azul cuando la puerta del dispositivo se cierra. Si se mantiene la puerta abierta del prototipo por más de 60 segundos, el sistema envía un mensaje de texto SMS al usuario y le advierte de esta condición. La puerta del dispositivo debe cerrarse para continuar con la siguiente etapa.



Figura 31 Sensor de apertura y cierre de puerta del automóvil

Fuente: Autor – 2016

4. Para probar la pantalla táctil se digita un código incorrecto y espera la activación del sistema de alerta con la sirena, se realiza el mismo proceso con el código correcto en la pantalla táctil resistiva, en este caso se ingresa la clave 1-2-3-4, el sistema confirma la misma y se podrá encender el vehículo, el LED RGB indica que el proceso es correcto encendiéndose en color verde. Se verifica que el relé actúa mediante el encendido de un pequeño led color verde, estas pruebas envían un SMS con su respectiva advertencia y también recibe un SMS de confirmación, la figura 32 detalla lo anotado.



Figura 32 Sistema de encendido activado

Fuente: Autor - 2016

5. El sistema puede desactivarse cuando el usuario así lo disponga, en especial cuando el vehículo entre a mantenimiento, reparación, lubricación. Para esto debe ingresar al modo de servicio o de mantenimiento, el cual se ingresa con la clave o patrón de encendido, pero ingresada de forma contraria, en este caso 4-3-2-1, la puerta del vehículo debe estar abierta como condición para ingresar a este modo de operación.
6. La función Anti atraco funciona mientras el vehículo esta encendido, si por algún motivo la puerta del conductor se abre, el sistema detecta este escenario como un intento de robo del vehículo o secuestro, el antiatraco se activa; determina un tiempo prudencial de para ingresar la clave de desbloqueo en la pantalla (110 segundos si la puerta se mantiene abierta y 55 segundos si la puerta es cerrada), luego el vehículo se apaga y se activa el relé donde se conecta la sirena, este modo antiatraco permanece activo hasta que el usuario reinicie el sistema por cualquiera de los 3 modos.
7. Para comprobar los 3 tipos de reinicio, se bloquea el sistema por varias ocasiones y se realizan los reinicios: por SMS, por pulsador y botón central de pantalla táctil. El reinicio por mensaje de texto ocurre cuando el usuario envía el mensaje "e10" al chip del módulo Sim900. El reinicio por pulsador se produce al presionar 2 veces consecutivas el botón de reinicio, y por último se reinicia el sistema al presionar el botón central de la pantalla táctil hasta que el LED RGB cambie a color blanco, esto indica que el sistema se reinició, el LED RGB a continuación parpadea en color rojo.
8. La última prueba corresponde al intento de encendido sin cumplir ningún requisito, es decir cuando alguien no conoce los pasos o requerimientos que se debe cumplir para encender el vehículo, el sistema advierte al usuario mediante un mensaje de texto SMS que a su vehículo lo intentan arrancar, y se activa la alarma o sirena, el prototipo no permite encender el vehículo.

Tabla 5 Tabla de mensajes de texto enviados al usuario

CONDICIÓN	TEXTO PREDEFINIDOS PARA SMS
Al concluir el ciclo completo	Auto desbloqueado puede iniciar su viaje
Al colocar clave de servicio	Advertencia!!! Modo mantenimiento
Al salir del modo de servicio	Advertencia!!! Salida modo mantenimiento
Al no cerrar la puerta del vehículo	Advertencia!!! Sensor de puerta detectado, auto bloqueado
Al no colocar la clave correcta	Advertencia!!! Clave Incorrecta o No Ingresada, Auto Bloqueado
Al activarse la función Anti atraco	Advertencia !!! Se ha detectado secuestro, auto bloqueado.
Al apagarse el vehículo	Auto apagado, se ha reiniciado el sistema
Al aplastar el botón central del panel táctil para corregir error	Advertencia!!! Sistema reiniciado
Al intentar prender el vehículo directamente	Advertencia!!! Se intenta arrancar el auto, auto bloqueado

Fuente: Autor – 2016



Figura 33 Captura de pantalla de mensajes de texto enviados al usuario.

Fuente: Autor – 2016

El sistema está diseñado para enviar al usuario mensajes de texto con leyendas predeterminados de acuerdo a la condición que el sistema detecte, esto permite mayor control de los eventos por parte del usuario siempre y cuando el chip o SimCard mantenga un saldo positivo. A continuación, en la tabla 5, se podrán observar los distintos mensajes que el sistema envía al celular del usuario y en la figura 33 la captura de pantalla gracias al software Vysor.

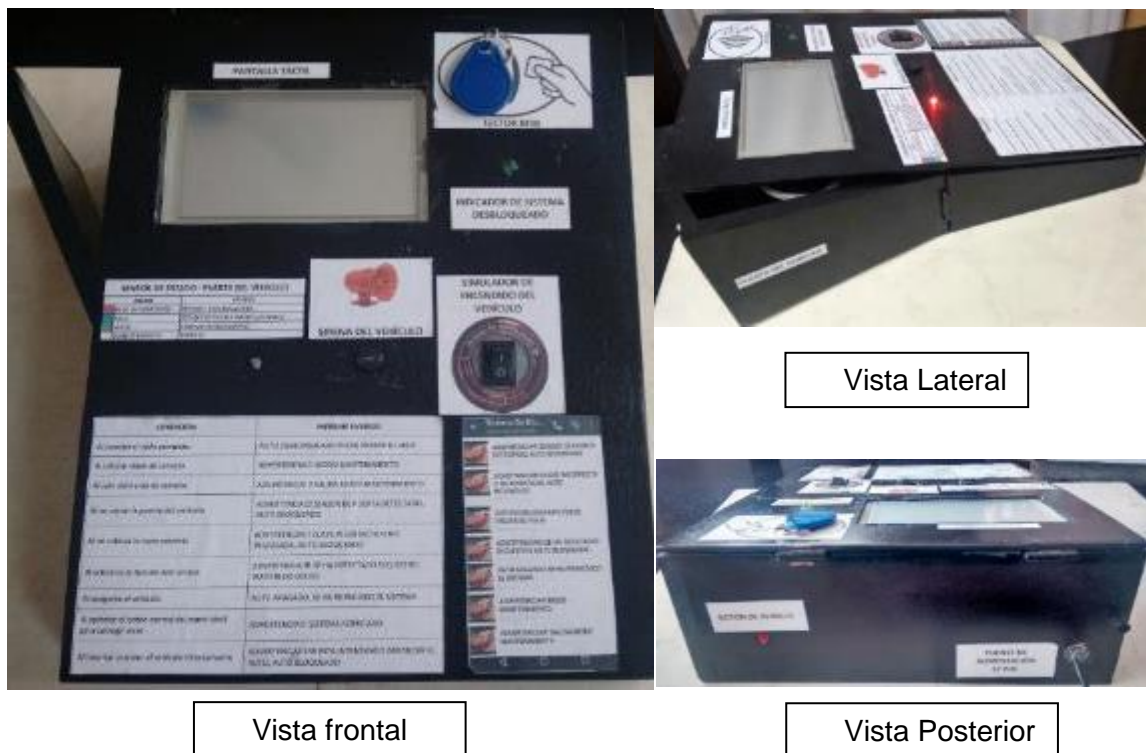


Figura 34 Prototipo final

Fuente: Autor - 2016

Una vez concluidas las pruebas y confirmado el correcto funcionamiento del sistema se construye el prototipo final, el módulo central se lo fabricó con mica acrílica blanca y su tapa con mica transparente mientras que la caja del prototipo es de madera, en la parte frontal se encuentra la pantalla táctil resistiva, un zumbador que simula a la sirena, un interruptor que simula a la llave de encendido del vehículo, el lector RFID y los LEDs que indican los estados del sistema. En la parte lateral se observa el simulador de puerta del vehículo y en la parte posterior se ubicó el botón de reinicio y la entrada de alimentación, también se colocó la señalética respectiva como se observa en la figura 34.

4.7 Análisis de costos

Tabla 6 Costos del prototipo

Cantidad	Detalle	Valor Unitario	Total
1	Arduino uno	\$ 16,00	\$ 16,00
1	Shield gsm sim900	\$ 50,00	\$ 50,00
1	Touch de 3.2"	\$ 20,00	\$ 20,00
1	Módulo RFID RC522	\$ 10,00	\$ 10,00
	Resistencias		
3	100R	\$ 0,03	\$ 0,09
3	4k7	\$ 0,03	\$ 0,09
2	10k	\$ 0,03	\$ 0,06
2	1k	\$ 0,03	\$ 0,06
	Capacitores		
1	470uF	\$ 0,19	\$ 0,19
2	104	\$ 0,12	\$ 0,24
1	10u	\$ 0,09	\$ 0,09
	Circuitos Integrados		
1	7809	\$ 0,45	\$ 0,45
1	OPTOCOUPLER-NPN	\$ 0,64	\$ 0,64
	Transistores		
2	2N3904	\$ 0,04	\$ 0,08
	Diodos		
2	1N4007	\$ 0,13	\$ 0,26
	Misceláneos		
1	Chip Movistar	\$ 10,00	\$ 10,00
6	Borneras	\$ 0,37	\$ 2,22
1	Zócalo 8 pines	\$ 0,08	\$ 0,08
1	Cables y conectores	\$ 4,00	\$ 4,00
1	LED RGB	\$ 1,00	\$ 1,00
2	Leds	\$ 0,06	\$ 0,12
2	Zumbador redondo	\$ 0,67	\$ 1,34
2	Relés 10 A	\$ 1,00	\$ 2,00
2	Placa	\$ 1,60	\$ 3,20
1	Módulo/caja	\$ 25,00	\$ 25,00
1	Programación	\$ 120,00	\$ 120,00
1	Horas-hombre Instalación	\$ 50,00	\$ 50,00
	TOTAL		\$ 317,21

Fuente: Autor – 2016

En el mercado se pueden conseguir varias alternativas para brindar seguridad a los vehículos, los sistemas tradicionales alertan sobre un posible incidente en el vehículo, pero si el propietario no atiende a esta alarma, posiblemente esta no aleje al extraño y pierda su vehículo. Otros dispositivos solo bloquean mecánica o eléctricamente al automotor y otros más sofisticados incluyen tecnología de punta, evitan el robo del auto por medio de bloqueos inteligentes al vehículo.

De estos sistemas, los más eficientes y seguros, son los que integran tecnología nueva, tales como geocalización (GPS), monitoreo del vehículo ya sea a través de video o de SMS, los que integran teclados y todo tipo de dispositivos que a pesar de ser más costosos resultan ser la una mejor opción para evitar ser víctimas de la delincuencia.

Una ventaja del prototipo presentado es que combina varias tecnologías que lo hacen una nueva alternativa para ser instalado en cualquier tipo de vehículo (a excepción de los vehículos eléctricos o híbridos), en la investigación realizada no se encontró ningún sistema que este diseñado de la forma propuesta en este proyecto, por lo que se considera una idea innovadora.

El sistema requiere de un presupuesto de \$317,21 USD como se detalla en la tabla 6, un costo medio para el mercado actual; sin embargo, al producirlo a gran escala, los costos se reducen notablemente, tener en cuenta que los gastos asociados al robo del vehículo son bastante grandes, por lo expuesto su producción es factible y viable al ser una buena opción para el mercado.

CONCLUSIONES

- Respecto al análisis de los distintos sistemas, la mayoría de los vehículos cuentan con un dispositivo de bloqueo a través de alarma tradicional que no brinda la seguridad necesaria al usuario, debido a que los propietarios de los automotores no están familiarizados con las nuevas tecnologías que se desarrollan en el mercado para la protección de sus bienes.
- Desde el punto de vista del diseño presentado se concluye que el uso de tecnología RFID, pantalla táctil resistiva y el envío de SMS presentan al mercado una alternativa accesible, segura, de costo medio que lo hacen atractivo para el usuario que busca proteger sus bienes y así lograr disminuir el índice de robos.
- En cuanto a la implementación del prototipo desarrollado innova a los sistemas tradicionales del mercado, personas extrañas no tendrán el conocimiento del funcionamiento de dicho dispositivo debido a que no existe un sistema que cuente con estas características.
- En cuanto a la validación del prototipo, se usó el método de prueba de defectos con un resultado del 81% de fiabilidad de un total de 50 pruebas realizadas, estas fallas no fueron producto en sí del prototipo, sino por errores de usuario en cuanto al ingreso de la clave o insuficiencia de saldo, también por cobertura de red celular en el lugar donde se realizaron las pruebas.
- Las tecnologías RFID/NFC aún en desarrollo en estos últimos años dan lugar a nuevas aplicaciones para esta tecnología, fueron probadas en celulares, control de acceso, procesos bancarios etc, aplicaciones donde demostró ser muy segura y de bajo costo.

RECOMENDACIONES

- Para la instalación de este sistema en un vehículo se deberá analizar bien la ubicación del panel táctil, este debe estar bien asegurado a una superficie sólida para evitar roturas del mismo, así también debe estar accesible para el usuario.
- Tener en cuenta la calibración de las pantallas táctiles, ya que las probadas dieron diferentes lecturas a pesar de ser nuevas e iguales, así mismo tener mucho cuidado con el bus de datos que posee el panel táctil ya que es extremadamente delicado.
- A pesar de ser muy similares, cada shield sim900 puede cambiar en su configuración de pines, especialmente para TX, RX y autoarranque, esto depende del fabricante por lo que se deberá revisar este aspecto con la documentación disponible en la web.
- El envío de mensajes de texto por medio del módulo SIM900 dependerá del chip o SimCard instalado en el módulo, de la cobertura de la operadora celular y del sitio donde se encuentre el vehículo, en este caso en particular el chip instalado es de la operadora Movistar, sencillamente por efectos prácticos, el sector norte de Quito (San José de Morán) donde se realizaron las pruebas no tuvo la respuesta adecuada de la operadora Claro, por otro lado, la operadora CNT presenta caídas de red constantes en este sector, a pesar de lo mencionado, se recomienda el uso de un chip Claro, por su principal ventaja en cobertura a nivel nacional..
- Es necesario tener saldo positivo en el chip o SimCard para el envío de mensajes si se desea dar un completo uso del sistema.

BIBLIOGRAFÍA

Poma García, José (2015). Proyecto domótica con Arduino. Proyecto: Programación aplicada, Universidad Nacional del Callao, Perú. Recuperado de:
<http://documents.mx/documents/proyecto-domotica-con-arduino-informe-pdf.html>

Regalado Pacheco, Juan (2015). Curso de supervivencia con Arduino, creado a partir de la obra en Arduino.cc, Canarias, España. Recuperado de:
<https://poemaselectrodomesticos.files.wordpress.com/2011/10/libreto.pdf>

Martínez de Carvajal Hedrich, Ernesto (20 de diciembre de 2015). 100 Proyectos de Robótica con Bitbloq y Arduino (2ª edición). Martínez de Carvajal Hedrich. p. 386.

Carlos Tapia, Héctor (2013). Evaluación de la plataforma arduino e implementación de un sistema de control de posición horizontal (tesis de pregrado). Universidad Politécnica Salesiana, Guayaquil, Ecuador. Recuperado de
<https://es.scribd.com/doc/240130661/11/Tabla-2-1-Modelos-de-placas-Arduino-Modelos-microcontroladores>.

Tomasi Wayne. (2003) Sistemas de Comunicaciones Electrónicas (4ª edición). Pearson Educación, México, p.898.

Arduino. (2016). ARDUINO. Recuperado de
<https://www.arduino.cc/en/Guide/Introduction#>
<https://www.arduino.cc/en/Main/ArduinoBoardUno>

Diario El Comercio (2014, 02 julio). Mafias usan nueva tecnología para robar carros. Recuperado de: <http://www.elcomercio.com/actualidad/ecuador-alarmas-robo-autos-seguro-gps-tecnologia.html>.

REFERENCIAS BIBLIOGRAFICAS

Arduino. (2016). *Módulo RFID-RC522 Escritura y Lectura*. Obtenido de Módulo Lector RFID-RC522 RF con Arduino: <http://hetpro-store.com/TUTORIALES/modulo-rfid-rc522-rf-escritura-lectura/>

Blog. (03 de 2016). *Inmovilizador electrónico*. Obtenido de Aficionados a la Mecánica: <http://www.aficionadosalamecanica.com/inmovilizador.htm>

Blog, A. d. (2013). *Inmovilizadores y Alarmas* . Obtenido de <http://inmovilizadoresyalarmas.blogspot.com/>

Bunker Electronic. (2016). <http://www.bunkeralarms.com>. Obtenido de <http://www.bunkeralarms.com/principal/manuales.htm>:

Garcia, A. B. (2006). *Telefonia Movil*. Obtenido de Garcia-Saez: http://www.info-ab.uclm.es/labelec/Solar/Comunicacion/Telefonia_movil/index.htm

K-twin. (2011). <http://blog.k-tuin.com>. Obtenido de Cómo funcionan las pantallas de iPad e iPhone: <http://www.k-tuin.com/blog/como-funciona-la-pantalla-del-ipad/>

Martinez, E. (2011). La evolución de la telefonía móvil. *Revista RED*, 6. Obtenido de http://s3.amazonaws.com/academia.edu.documents/32975756/La_evolucion_de_la_telefonia_movil.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1483490120&Signature=HGx3pBq%2Bz04aPIbXbnEiXqKffH0%3D&response-content-disposition=inline%3B%20filename%3DLECTURAS_L

Minterior. (2015). *Más de 10 mil autos y motos recuperadas y retenidas, a escala nacional*. Obtenido de Ministerio del Interior: <http://www.ministeriointerior.gob.ec/mas-de-10-mil-autos-y-motos-recuperadas-y-retenidas-a-escala-nacional/>

Minterior. (2015). *Se implementa 'Plan Impacto' frente al robo de vehículos en Quito*. Obtenido de Ministerio del Interior: <http://www.ministeriointerior.gob.ec/se-implementa-plan-impacto-frente-al-robo-de-vehiculos-en-quito/>

Regalado, J. G. (2011).

<https://poemaselectrodomesticos.files.wordpress.com/2011/10/libreto.pdf>.

Obtenido de Curso de supervivencia con Arduino:

<https://poemaselectrodomesticos.files.wordpress.com/2011/10/libreto.pdf>

Rosinelys, S. R. (03 de 2013). *Tecnología RFID*. Obtenido de

<http://www.eoi.es/blogs/scm/2013/03/06/tecnologia-rfid/>

Smartech. (2016). <http://smartech.com.ec/>. Obtenido de <http://smartech.com.ec/>

Tapia Carlos, M. H. (2013). *EVALUACION DE LA PLATAFORMA ARDUINO E IMPLEMENTACION DE UN SISTEMA DE CONTROL DE POSICION HORIZONTAL*. Obtenido de

<https://es.scribd.com/document/240130661/Arduino>

Tomas, B. (2007). *Sabelotodo.org*. Obtenido de Sistema de encendido.

ANEXOS

ANEXO A:

MANUAL DE USUARIO



**SISTEMAS DE SEGURIDAD PARA VEHÍCULOS USANDO UNA PANTALLA
TÁCTIL, RFID Y MENSAJES DE TEXTO**


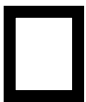
**Autor:
EDIN LIMA**

Primera edición, 2016


1. Introducción

LITHIUM es un sistema de bloqueo que permite proteger su vehículo mediante la implementación de tecnología RFID, touch o pantalla táctil resistiva, y envío de SMS, para prevenir el robo de un vehículo, ofrece al usuario tranquilidad, comodidad y seguridad.

2. Simbología del Manual:

	Peligro
	Importante Información Adicional

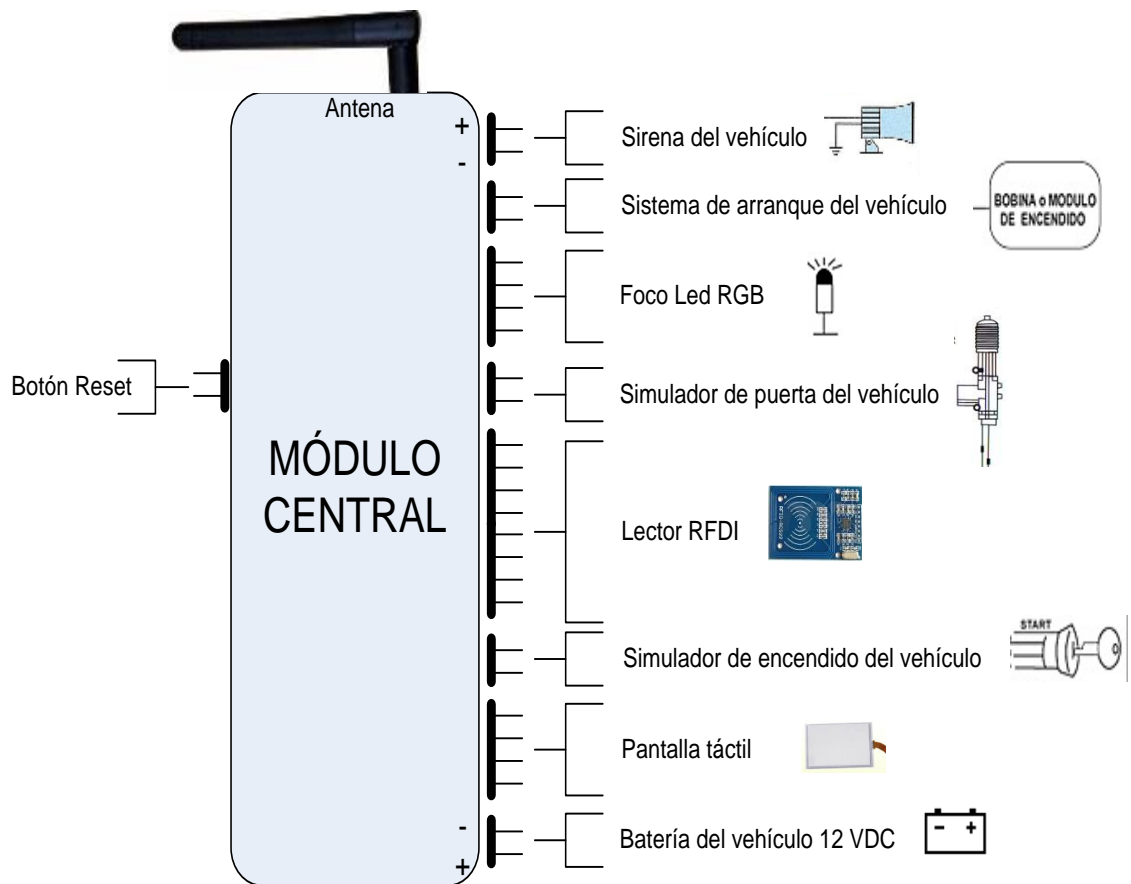
3. Precauciones Generales

	Este producto debe ser instalado únicamente por personal técnico calificado, su uso inadecuado puede generar quemaduras graves e incluso la muerte. Revise bien las conexiones antes de arrancar el sistema. Nunca tope la placa de circuito impreso con objetos metálicos cuando el módulo esté conectado.
---	---

4. Descripción

Lithium permite mejorar el diseño de los sistemas clásicos de seguridad implementado nuevas tecnologías tales como una pantalla táctil, etiquetas RFID y mensajes de texto.

Así mismo, comprobar los resultados en forma real con la evidencia de los mensajes de texto al usuario del sistema, es parametrizable para cualquier tipo de vehículo.



Descripción:

1. Módulo central.
2. Botón de reset
3. Sirena del vehículo
4. Al sistema de arranque o bloqueo inteligente
5. Indicador de estado – LED RGB
6. Al sensor de apertura de puerta.
7. Tarjetas RFID (incluye llavero y tarjeta)
8. Interruptor de encendido del vehículo
9. Pantalla táctil resistiva.
10. Alimentación desde la batería.

5. Guía de uso

El sistema de seguridad del vehículo funciona de la siguiente forma:

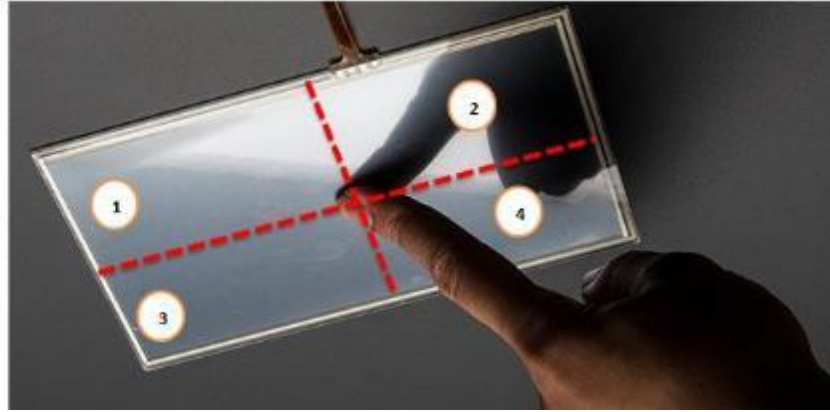
- a. El usuario debe aproximar su tarjeta o llavero RFID instalado en un sitio oculto del vehículo, usted deberá acercarse al vehículo de forma que este sea detectado por el lector.



- b. El usuario abrirá la puerta del vehículo, con la llave o con el control de radiofrecuencia, el sistema detectará la apertura y cierre de la puerta mediante el pulsador.



- c. El usuario al ingresar al vehículo, deberá colocar un patrón o clave en la pantalla táctil resistentiva (el técnico le entregará su clave), la pantalla táctil tiene 4 cuadrantes para configurar los códigos, cada cuadrante equivale a un dígito.



Adicional, en el centro de la pantalla táctil se diseñó y programó un botón de activación/reset del sistema.

Mientras el sistema está activo realiza el bloqueo inteligente (bloqueo el sistema de arranque, de encendido o algún circuito vital del vehículo), al cumplir el ciclo completo de verificación, el bloqueo inteligente se desactivará y podrá encender el vehículo.

Control anti-atraco,

- a. El modo antiatraco se activa cuando el vehículo está encendido y se abre la puerta del conductor, en este caso, se deberá ingresar la clave de desbloqueo en el lapso de 60 segundos, de no ser así, el sistema deberá cortar el paso de gasolina y encenderá la alarma del vehículo. De igual forma, el sistema envía un SMS con la respectiva advertencia. La función del antiatraco es la inmovilización del vehículo en caso de robo o secuestro.

Estos 3 requisitos podrán ser visualizados mediante un LED tricolor o RGB, el color rojo parpadeante indicará que el sistema está activo, al cumplir el primer requisito (RFID) este dejará de parpadear, luego al abrir la puerta del vehículo este cambiará a color azul en espera del código en el panel táctil, el LED RGB se pondrá verde una vez cumplido los 3 requisitos.

Modo de servicio o Mantenimiento

Para la opción del modo de servicio o mantenimiento, se ingresará un código o patrón adicional (es el mismo código de usuario digitado al revés) mientras la puerta del vehículo esté abierta. En este caso el LED RGB se apaga por completo. En este modo,

el vehículo podrá encenderse sin ninguna restricción, ideal cuando necesite servicio técnico, de mantenimiento o lavado del automotor

Desbloques

Para el modo de desbloqueo existen 3 alternativas:

- La puerta deberá estar abierta, presionar el botón central del panel táctil hasta que se encienda por 2 ocasiones el LED en color blanco.
- A través del código celular que deberá ser enviado por el usuario al chip del vehículo, para este caso se usó el código "e10".
- Mediante un pulsador que se encuentra en el módulo principal.

6. Guía rápida para solución de problemas

Problema más comunes	Solución
No equipo no se enciende.	Verificar que la batería del vehículo este en buen estado. Revise el conector del módulo central.
El equipo enciende, pero no funciona el lector RFID.	Verificar que este encendido el modulo central, revise las conexiones del lector, al inicio y al final del bus de datos. Revise si no existen cables rotos.
El equipo enciende, pero no funciona sensor de la puerta	Verificar las conexiones del sensor al inicio y al final, revise que no existan cables rotos.
El equipo enciende, pero no funciona la pantalla táctil	Verificar las conexiones del sensor al inicio y al final, revise que no existan cables rotos. El bus de datos de la pantalla táctil es bastante frágil, revise este dispositivo con extremo cuidado.
El equipo funciona pero no llegan los SMS al usuario.	Revise la alimentación del módulo sim900, revise si el chip está correctamente colocado. Revise si el chip tiene saldo.

7. Contacto de la organización:

Ing. Edin Lima

Gerente Corporativo CEO Teléfono 024516056 Cel. 0983820181

Dirección: San José de Moran, Quito – Ecuador

Página web: [https:// www.edin&aso.com.ec](https://www.edin&aso.com.ec)

ANEXO B

Modelos de placas Arduino con su respectivo microcontrolador

Modelos placas Arduino	Modelos microcontroladores uC
Arduino Due	AT91SAM3X8E
Arduino Leonardo	Atmega 32U4
Arduino uno	Atmega 328p
Arduino Duemilanove	Atmega 168
Arduino Pro 3.3V/8MHz	Atmega 328
Arduino Pro 5V/16MHz	Atmega 328
Arduino Mega 2560 R3	Atmega 2560
Arduino Mega	Atmega 1280
Mega Pro 3.3V	Atmega 2560
Mega Pro 5V	Atmega 2560
Arduino Mini 05	Atmega 328
Pro Micro 5V/16MHz	Atmega 32U4
Pro Micro 3.3V/8MHz	Atmega 32U4

Fuente: CARLOS TAPIA, HECTOR MANZANO (2013).

ANEXO C

Microcontrolador Atmega328 - Resumen de características Técnicas

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Resumen de características Técnicas Arduino UNO

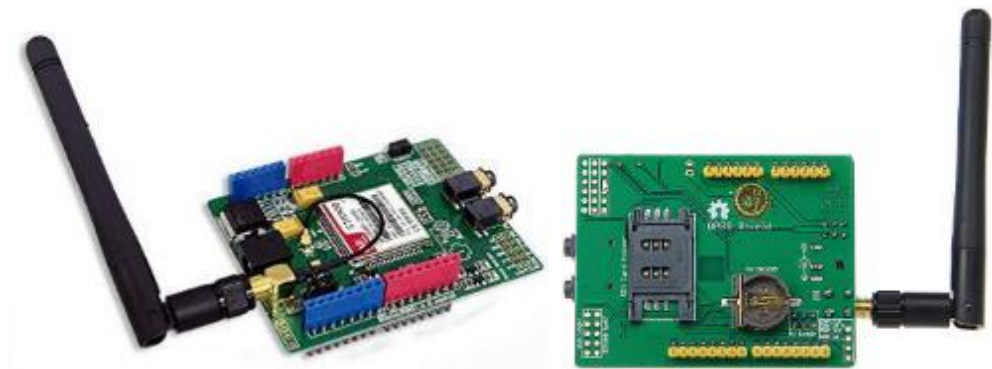
Fuente: Arduino - Website

ANEXO D

Especificaciones Módulo Gsm Sim900

Especificaciones Módulo Gsm Sim900

- o Totalmente compatible con Arduino
- o Conexión con el puerto serial
- o Quad-Band 850/ 900/ 1800/ 1900 Mhz
- o GPRS multi-slot clase 10/8GPRS mobile station clase B
- o Compatible GSM fase 2/2+Clase 4 (2 W (AT) 850 / 900 MHz)
- o Clase 1 (1 W (AT) 1800 / 1900MHz)TCP/UP embebido
- o Soporta RTC
- o Consumo de 1.5 mA (susp)



b)

b)

Módulo GSM Sim900. a) Vista frontal. b) Vista posterior

Fuente: <http://www.ebay.com/itm/Geetech-New-SIM900-Quad-band-GSM-GPRS-Shield-SMS-MMS-SIMCOM-UART-for-Arduino-/181014840970?hash=item2a2553468a:g:DwQAAOxyLm9TDvef>

Interfaz de radio de GSM en diferentes frecuencias

Existen 4 bandas de frecuencias (850, 900, 1800 ,1900):

Banda GSM	Nombre	Canales	Uplink (MHz)	Downlink (MHz)
850	GSM 850	128 - 251	824,0 - 849,0	869,0 - 894,0
900	P-GSM 900	0-124	890,0 - 915,0	935,0 - 960,0
	E-GSM 900	974 - 1023	880,0 - 890,0	925,0 - 935,0
	R-GSM 900	n/a	876,0 - 880,0	921,0 - 925,0
1800	GSM 1800	512 - 885	1710,0 - 1785,0	1805,0 - 1880,0
1900	GSM 1900	512 - 810	1850,0 - 1910,0	1930,0 - 1990,0

ANEXO E

Características del Módulo Lector RFID-RC522 RF

Características del Módulo Lector RFID-RC522 RF

- Modelo: MF522-ED
- Corriente de operación: 13-26mA a 3.3V
- Corriente de stand by: 10-13mA a 3.3V
- Corriente de sleep-mode: <80uA
- Corriente máxima: 30mA
- Frecuencia de operación: 13.56Mhz
- Distancia de lectura: 0 a 60mm
- Protocolo de comunicación: SPI
- Velocidad de datos máxima: 10Mbit/s
- Dimensiones: 40 x 60 mm
- Temperatura de operación: -20 a 80°C
- Humedad de operación: 5%-95%
- Máxima velocidad de SPI: 10Mbit/s
- Incluye pines, llavero y tarjeta

ANEXO F

Programa para calibración de pantalla resistiva

```
//código para leer las coordenadas X y Y
//Los pines analogicos pueden ser usados
//como pines digitales con los números 14
//(entrada analogica número 0) hasta 19
//(entrada analogica número 5).
//boton 1: x=9-11 y= 17-19
//boton 2: x=10-12 y= 45-49
//boton 3: x=19-22 y= 15-17
//boton 4: x=21-23 y= 45-48
#define y_alto A0//redefino los pines para ser configurados como digitales
//y en analogas uso A0, A1
#define x_bajo A1//15
#define y_bajo A2//16
#define x_alto A3//17

#include <SPI.h>
#include <RFID.h>
#define SS_PIN 10
#define RST_PIN 18

#include <EEPROM.h>

RFID rfid(SS_PIN, RST_PIN);
// Setup variables:
int serNum0;
int serNum1;
int serNum2;
int serNum3;
int serNum4;

int valor0=131;
int valor1=158;
int valor2=172;
int valor3=171;
int valor4=26;

int valorb0=83;
int valorb1=62;
int valorb2=20;
int valorb3=188;
int valorb4=197;
```

```

int bandera;
int cordenada_xx;
int cordenada_yy;

String string=0;
String caracter=0;
String caracterx=0;
int tecla = 0;
String clave="1234";
int temporizador=0;

void setup()
{
  // EEPROM.write(0, 0);
  ///////////////////////////////////////////////////configuracion serial
  Serial.begin(9600);
}
//////////////////////////////////////
void loop()

{
  leertouch();
  if (cordenada_xx>=3 && cordenada_xx<=7){
    if (cordenada_yy>=5 && cordenada_yy<=9){
      Serial.println("BOTON UNO");
    posicion();
    }
  }
  if ((cordenada_xx>=4) && (cordenada_xx<=8)){
    if ((cordenada_yy>=50) && (cordenada_yy<=57)){
      Serial.println("BOTON DOS");
    posicion();
    }
  }
  if ((cordenada_xx>=22) && (cordenada_xx<=28)){
    if ((cordenada_yy>=7) && (cordenada_yy<=12)){
      Serial.println("BOTON TRES");
    posicion();
    }
  }
  if ((cordenada_xx>=20) && (cordenada_xx<=25)){
    if ((cordenada_yy>=49) && (cordenada_yy<=55)){
      Serial.println("BOTON CUATRO");
    posicion();
    }
  }
}

void leer_x()

```



```

{
  pinMode (x_alto, OUTPUT);
  pinMode (x_bajo, OUTPUT);
  pinMode (y_alto, INPUT);
  pinMode (y_bajo, INPUT);
  digitalWrite (x_alto, HIGH);
  digitalWrite (x_bajo, LOW);
  Serial.print("el valor de VALOR X es...");
  int cordenada_x = analogRead(A2); // A2 ES Y BAJO TENER CUIDADO
  Serial.println(cordenada_x);
  cordenada_xx=cordenada_x;
}
////////////////////////////////////
void leer_y()
{
  pinMode (x_alto, INPUT);
  pinMode (x_bajo, INPUT);
  pinMode (y_alto, OUTPUT);
  pinMode (y_bajo, OUTPUT);
  digitalWrite (y_alto, HIGH);
  digitalWrite (y_bajo, LOW);
  Serial.print("el valor de VALOR Y es...");
  int cordenada_y = analogRead(A1);
  Serial.println(cordenada_y);
  cordenada_yy=cordenada_y;
}
void leertouch(){
  leer_x();
  delay(400);
  leer_y();
  delay(400);
  Serial.println(" ");
}
void posicion(){
  Serial.println(cordenada_xx);
  Serial.println(cordenada_yy);
  delay(1000);
}

void analisis0(){
if (rfid.serNum[0]==valor0){
  analisis1();
}
}
void analisis1(){
if (rfid.serNum[1]==valor1){
  analisis2();
}
}
}

```

```

void analisis2(){
if (rfid.serNum[2]==valor2){
analisis3();
}
}

void analisis3(){
if (rfid.serNum[3]==valor3){
analisis4();
}
}
void analisis4(){
if (rfid.serNum[4]==valor4){
analisis5();
}
}
void analisis5(){
Serial.println("A");
EEPROM.write(0, 1);
Serial.println("RESETEANDO...");
delay(1000);
asm volatile (" jmp 0");
}
}
void analisisb(){
if (rfid.serNum[0]==valorb0){
analisisb1();
}
}
void analisisb1(){
if (rfid.serNum[1]==valorb1){
analisisb2();
}
}
void analisisb2(){
if (rfid.serNum[2]==valorb2){
analisisb3();
}
}
void analisisb3(){
if (rfid.serNum[3]==valorb3){
analisisb4();
}
}
void analisisb4(){
if (rfid.serNum[4]==valorb4){
analisisb5();
}
}
}

```

```
void analisisb5(){
  Serial.println("B");
  EEPROM.write(0, 1);
  Serial.println("RESETEANDO...");
  delay(1000);
  asm volatile (" jmp 0");
}
void caracterxcaracter(){
  caracter=string;
}
void encerartecla(){
  tecla=0;
}
void verificarclave(){
  if (clave==caracter){
    bandera=0;
    caracter=0;
    string=0;
    Serial.println("CLAVE CORRECTA");
    Serial.println("RESETEANDO...");
    delay(1000);
    EEPROM.write(0, 0);
    asm volatile (" jmp 0");
  }
}
void timeout(){
  Serial.println("TIME OUT RESET...");
  delay(1000);
  EEPROM.write(0, 0);
  asm volatile (" jmp 0");
}
```

ANEXO G

Programa general del sistema:

```
//ELIMINÉ LA CONEXION SERIAL PARA QUE FUNCIONE
//PIN 0 y 1
//código para leer las coordenadas X y Y
//Los pines analogicos pueden ser usados
//como pines digitales con los números 14
//(entrada analogica número 0) hasta 19
//(entrada analogica número 5).
//boton 1: x=15-17 y= 10-12
//boton 2: x=11-25 y= 13-30
//boton 3: x=13-17 y= 67-82
//boton 4: x=14-18 y= 11-31
//BOTON CENTRAL X=9-11 Y=47-53
#define y_alto A0//redefino los pines para ser configurados como digitales
//y en analogas uso A0, A1
#define x_bajo A1//15
#define y_bajo A2//16
#define x_alto A3//17

#include <SPI.h>
#include <RFID.h>
#define SS_PIN 10
#define RST_PIN 18

#include <EEPROM.h>

#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8);

RFID rfid(SS_PIN, RST_PIN);
// Setup variables:
int serNum0;
int serNum1;
int serNum2;
int serNum3;
int serNum4;

int valor0=131;
int valor1=158;
int valor2=172;
int valor3=171;
int valor4=26;

int valorb0=83;
int valorb1=62;
int valorb2=20;
int valorb3=188;
int valorb4=197;

int bandera;
int cordenada_xx;
int cordenada_yy;

String string=0;
String character=0;
String characterx=0;
int tecla = 0;
String clave="1234";
String clavex="4321";

int temporizador=0;
int temporizadora=0;
int dato;
int suiche=6;
int rojo=2;
int verde=3;
int azul=4;
int rele=1;
int buzzer=0;
```

```

int sirena=5;
int sensor=19;

int k;
char inchar;

void setup()
{
    ///////////////configuracion serial
    //Serial.begin(9600);Se desactiva para que suene zumbador
    // si se activa "serial begin" se prueba con monitor serial
    SPI.begin();
    rfid.init();

    pinMode(sensor, INPUT);
    pinMode(suiche, INPUT);
    pinMode(rojo, OUTPUT);
    pinMode(verde, OUTPUT);
    pinMode(azul, OUTPUT);
    pinMode(rele, OUTPUT);
    pinMode(buzzer, OUTPUT);
    pinMode(sirena, OUTPUT);

    digitalWrite(rojo,HIGH);
    digitalWrite(verde,HIGH);
    digitalWrite(azul,HIGH);
    digitalWrite(rele,LOW);
    digitalWrite(buzzer,LOW);
    digitalWrite(sirena,LOW);

    SIM900.begin(19200);

    SIM900power();
    delay(10000);

    SIM900.print("AT+CMGF=1\r"); // set SMS mode to text
    delay(100);
    SIM900.print("AT+CNMI=2,2,0,0,0\r");
    // blurt out contents of new SMS upon receipt to the GSM shield's serial out
    delay(100);
}
//////////////////////
void loop()
{
    if (bandera==0){
        digitalWrite(rojo,LOW);
        if (rfid.isCard()) {
            if (rfid.readCardSerial()) {
                if (rfid.serNum[0] != serNum0
                    && rfid.serNum[1] != serNum1
                    && rfid.serNum[2] != serNum2
                    && rfid.serNum[3] != serNum3
                    && rfid.serNum[4] != serNum4
                ) {

                    serNum0 = rfid.serNum[0];
                    serNum1 = rfid.serNum[1];
                    serNum2 = rfid.serNum[2];
                    serNum3 = rfid.serNum[3];
                    serNum4 = rfid.serNum[4];

                    analisis0();
                    analisisb();

                }
            }
        }

        rfid.halt();
        digitalWrite(rojo,HIGH);
        delay(40);
    }
}

```

```

dato = digitalRead(sensor);
  if (dato == LOW){
    sendSMSARRANQUE();
    bandera=20;
  }
}

if (bandera==1){

  dato = digitalRead(suiche);
  if (dato == HIGH){
    bandera=2;
    digitalWrite(azul,LOW);
    Serial.println("PUERTA ABIERTA");
    delay(3000);
    temporizador=0;
    temporizadora=0;
    serNum0=0;
    serNum1=0;
    serNum2=0;
    serNum3=0;
    serNum4=0;
  }

  temporizadora=temporizadora +1;
  delay(10);

if (temporizadora ==5000){
  sendSMSPUERTA();
  timeout();
}
}

  if (bandera==2){
    dato = digitalRead(suiche);
    if (dato == LOW){
      Serial.println("PUERTA CERRADA");
      bandera=3;
      temporizador=0;
      temporizadora=0;
      serNum0=0;
      serNum1=0;
      serNum2=0;
      serNum3=0;
      serNum4=0;
      digitalWrite(azul,HIGH);
      delay(2000);
      digitalWrite(azul,HIGH);
    }

    temporizadora=temporizadora +1;
    delay(10);

if (temporizadora ==5000){
  sendSMSPUERTA();
  timeout();
}
}

  if (bandera==3){// etapa de ingreso de clave

  leertouch();

if (cordenada_xx>=14 && cordenada_xx<=17){
  if (cordenada_yy>=10 && cordenada_yy<=16){
    Serial.println("BOTON UNO");
    indicador();
    caracterx="1";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
  }
  posicion();
}
}

```

```

}

    if ((cordenada_xx>=15) && (cordenada_xx<=17)){
    if ((cordenada_yy>=72) && (cordenada_yy<=84)){
    Serial.println("BOTON DOS");
    indicador();
    caracterx="2";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
    posicion();
    }
}

if ((cordenada_xx>=2) && (cordenada_xx<=4)){
    if ((cordenada_yy>=11) && (cordenada_yy<=18)){
    Serial.println("BOTON TRES");
    indicador();
    caracterx="3";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
    posicion();
    }
}

    if ((cordenada_xx>=3) && (cordenada_xx<=4)){
    if ((cordenada_yy>=74) && (cordenada_yy<=84)){
    Serial.println("BOTON CUATRO");
    indicador();
    caracterx="4";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
    posicion();
    }
}

    if ((cordenada_xx>=9) && (cordenada_xx<=11)){
    if ((cordenada_yy>=42) && (cordenada_yy<=50)){
    Serial.println("ENCERAR");
    indicador();
    caracterx="5";
    caracter=0;
    string=0;
    temporizador=0;
    temporizadora=0;
    delay(100);
    encerartecla();
    posicion();
    }
}

    verificarclave();
    verificarclavex();
    temporizador=temporizador +1;
    delay(1);

    if (temporizador ==50){
    sendSMSTECLADO();
    timeout();
    }
}

if (bandera ==4){//etapa clave correcta MODO VIAJE
    digitalWrite(verde,LOW);
    digitalWrite(rele,HIGH);
    dato = digitalRead(suiche);
    if (dato == HIGH){

        botonreset();
    }
}

```

```

        if (bandera==4) {
            digitalWrite(rojo,HIGH);
            digitalWrite(azul,HIGH);
            digitalWrite(verde,HIGH);
            bandera=5;
            temporizador=0;
            temporizadora=0;
            Serial.println("Temporizador Antiatraco...");
            delay(2000);
        }

    }
}

if (bandera==5){//antiatraco con temporizador
digitalWrite(verde,LOW);
dato = digitalRead(suiche);
    if (dato == LOW){
        bandera=6;
        temporizador=0;
        temporizadora=0;
        Serial.println("INGRESE CLAVE");
    }

}

delay(20);
digitalWrite(verde,HIGH);
delay(20);

        temporizadora=temporizadora +1;
        delay(10);

if (temporizadora ==2000){
    sendSMSANTIATRACO();
    Serial.print("TEMPORIZADOR CONCLUIDO");
    bandera=20;
    temporizador=0;
    temporizadora=0;
    digitalWrite(rele,LOW);
    delay(1000);
}

}

if (bandera==6){// etapa de segundo ingreso de clave

    leertouch();

    if (cordenada_xx>=14 && cordenada_xx<=17){
        if (cordenada_yy>=10 && cordenada_yy<=16){
            Serial.println("BOTON UNO");
            indicador();
            caracterx="1";
            string+=caracterx;
            caracterxcaracter();
            delay(100);
            encerartecla();
        }
    }

    if ((cordenada_xx>=15) && (cordenada_xx<=17)){
        if ((cordenada_yy>=72) && (cordenada_yy<=84)){
            Serial.println("BOTON DOS");
            indicador();
            caracterx="2";
            string+=caracterx;
            caracterxcaracter();
            delay(100);
            encerartecla();
        }
    }
}
}

```



```

}

if ((cordenada_xx>=2) && (cordenada_xx<=4)){
  if ((cordenada_yy>=11) && (cordenada_yy<=18)){
    Serial.println("BOTON TRES");
    indicador();
    caracterx="3";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
    posicion();
  }
}

if ((cordenada_xx>=3) && (cordenada_xx<=4)){
  if ((cordenada_yy>=74) && (cordenada_yy<=84)){
    Serial.println("BOTON CUATRO");
    indicador();
    caracterx="4";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
    posicion();
  }
}

if ((cordenada_xx>=9) && (cordenada_xx<=11)){
  if ((cordenada_yy>=42) && (cordenada_yy<=50)){
    Serial.println("ENCERAR");
    indicador();
    caracterx="5";
    caracter=0;
    string=0;
    temporizador=0;
    temporizadora=0;
    delay(100);
    encerartecla();
    posicion();
  }
}

verificarclave2();
temporizador=temporizador +1;
delay(1);

if (temporizador ==50){
  sendSMSANTIATRACO();
  Serial.print("NO SE DETECTO INGRESO DE CLAVE");
  bandera=20;
  Serial.println("BLOQUEADO...");
  delay(1000);
}
}

if(bandera==20){//etapa de bloqueo
Serial.println("AUTO BLOQUEADO");

digitalWrite(rele,LOW);
digitalWrite(sirena,HIGH);
delay(1000);
bandera=21;
Serial.println("ESPERANDO DESBLOQUEO...");
}

if(bandera==21){//esperando REINICIO DE SISTEMA
  dato = digitalRead(suiche);
  if (dato == HIGH){
    botonreset();
  }
}

```

```

recibirSMS();
}

if(bandera==30){//etapa de mantenimiento
digitalWrite(rele,HIGH);
leertouch();
dato = digitalRead(suiche);
    if (dato == HIGH){

if (cordenada_xx>=14 && cordenada_xx<=17){
    if (cordenada_yy>=10 && cordenada_yy<=16){
Serial.println("BOTON UNO");
indicador();
    caracterx="1";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
posicion();
    }
}

    if ((cordenada_xx>=15) && (cordenada_xx<=17)){
if ((cordenada_yy>=72) && (cordenada_yy<=84)){
Serial.println("BOTON DOS");
indicador();
    caracterx="2";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
posicion();
}
}

if ((cordenada_xx>=2) && (cordenada_xx<=4)){
    if ((cordenada_yy>=11) && (cordenada_yy<=18)){
Serial.println("BOTON TRES");
indicador();
    caracterx="3";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
posicion();
}
}

if ((cordenada_xx>=3) && (cordenada_xx<=4)){
    if ((cordenada_yy>=74) && (cordenada_yy<=84)){
Serial.println("BOTON CUATRO");
indicador();
    caracterx="4";
    string+=caracterx;
    caracterxcaracter();
    delay(100);
    encerartecla();
posicion();
}
}

    if ((cordenada_xx>=9) && (cordenada_xx<=11)){
if ((cordenada_yy>=42) && (cordenada_yy<=50)){
Serial.println("ENCERAR");
indicador();
    caracterx="5";
    caracter=0;
    string=0;
    temporizador=0;
    temporizadora=0;
    delay(100);
    encerartecla();
posicion();
}
}
}
}

```

```

    verificarclavey();
}
}

} // llave void loop

void leer_x()
{
    pinMode (x_alto, OUTPUT);
    pinMode (x_bajo, OUTPUT);
    pinMode (y_alto, INPUT);
    pinMode (y_bajo, INPUT);
    digitalWrite (x_alto, HIGH);
    digitalWrite (x_bajo, LOW);
    int cordenada_x = analogRead(A2); // A2 ES Y BAJO TENER CUIDADO
    cordenada_xx=cordenada_x;
}
////////////////////////////////////
void leer_y()
{
    pinMode (x_alto, INPUT);
    pinMode (x_bajo, INPUT);
    pinMode (y_alto, OUTPUT);
    pinMode (y_bajo, OUTPUT);
    digitalWrite (y_alto, HIGH);
    digitalWrite (y_bajo, LOW);
    int cordenada_y = analogRead(A1);
    cordenada_yy=cordenada_y;
}

void leertouch(){
    leer_x();
    delay(400);
    leer_y();
    delay(400);

}

void posicion(){
    Serial.println(cordenada_xx);
    Serial.println(cordenada_yy);
    delay(1000);
}

void analisis0(){
if (rfid.serNum[0]==valor0){
    analisis1();
}
}

void analisis1(){
if (rfid.serNum[1]==valor1){
    analisis2();
}
}

void analisis2(){
if (rfid.serNum[2]==valor2){
    analisis3();
}
}

void analisis3(){
if (rfid.serNum[3]==valor3){
    analisis4();
}
}

void analisis4(){
if (rfid.serNum[4]==valor4){
    analisis5();
}
}

```

```

}
}

void analisis5(){
  Serial.print("TARJETA: ");
  Serial.println("A");
  bandera=1;
  digitalWrite(rojo,HIGH);
  digitalWrite(azul,HIGH);
  digitalWrite(verde,HIGH);
  EEPROM.write(0, 1);
  Serial.println("RESETEANDO...");
  delay(1000);
}

void analisisb(){
  if (rfid.serNum[0]==valorb0){
    analisisb1();
  }
}

void analisisb1(){
  if (rfid.serNum[1]==valorb1){
    analisisb2();
  }
}

void analisisb2(){
  if (rfid.serNum[2]==valorb2){
    analisisb3();
  }
}

void analisisb3(){
  if (rfid.serNum[3]==valorb3){
    analisisb4();
  }
}

void analisisb4(){
  if (rfid.serNum[4]==valorb4){
    analisisb5();
  }
}

void analisisb5(){
  Serial.print("TARJETA: ");
  Serial.println("B");
  bandera=1;
  digitalWrite(rojo,HIGH);
  digitalWrite(azul,HIGH);
  digitalWrite(verde,HIGH);
  EEPROM.write(0, 1);
  Serial.println("RESETEANDO...");
  delay(1000);
}

void caracterxcaracter(){
  caracter=string;
}

void encerartecla(){
  tecla=0;
}

void verificarclave(){
  if (clave==caracter){
    bandera=4;
    digitalWrite(rele,HIGH);
    digitalWrite(rojo,HIGH);
    digitalWrite(azul,HIGH);
    digitalWrite(verde,LOW);
    caracter=0;
  }
}

```

```

    string=0;
    temporizador=0;
    temporizadora=0;
    Serial.println("CLAVE CORRECTA");
    Serial.println("AUTO DESBLOQUEADO PUEDE INICIAR SU VIAJE");
    delay(1000);
    sendSMSVIAJE();
}
}

void verificarclave2(){
    if (clave==caracter){
        bandera=4;
        digitalWrite(rojo,HIGH);
        digitalWrite(azul,HIGH);
        digitalWrite(verde,LOW);
        caracter=0;
        string=0;
        Serial.println("CLAVE CORRECTA");
        delay(1000);
        temporizador=0;
        temporizadora=0;
    }
}

void verificarclavex(){
    if (clavex==caracter){

        dato = digitalRead(suiche);
        if (dato == HIGH){
            digitalWrite(rele,HIGH);
            bandera=30;
            digitalWrite(rojo,HIGH);
            digitalWrite(azul,HIGH);
            digitalWrite(verde,HIGH);
            caracter=0;
            string=0;
            temporizador=0;
            temporizadora=0;
            Serial.println("CLAVE CORRECTA MANTENIMIENTO");
            sendSMSSERVICIO();
            delay(1000);
        }
    }
}

void verificarclavey(){
    if (clave==caracter){

        digitalWrite(rele,LOW);
        bandera=0;
        digitalWrite(rojo,HIGH);
        digitalWrite(azul,HIGH);
        digitalWrite(verde,HIGH);
        caracter=0;
        string=0;
        temporizador=0;
        temporizadora=0;
        Serial.println("CLAVE CORRECTA SALIR MANTENIMIENTO");
        delay(1000);

        sendSMSSERVICIOSALIDA();
    }
}

void timeout(){

    Serial.println("TIME OUT RESET...");
    delay(1000);
    bandera=20;
    serNum0=0;
    serNum1=0;
    serNum2=0;
}

```

```

        serNum3=0;
        serNum4=0;
        temporizador=0;
        temporizadora=0;
        digitalWrite(rojo,HIGH);
        digitalWrite(verde,HIGH);
        digitalWrite(azul,HIGH);
    }

void SIM900power()
// software equivalent of pressing the GSM shield "power" button
{

    digitalWrite(9, HIGH);
    delay(1000);
    digitalWrite(9, LOW);
    delay(2000);

}

void sendSMSTECLADO()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);
    SIM900.println("ADVERTENCIA!!! CLAVE INCORRECTA O NO INGRESADA, AUTO BLOQUEADO");//
message to send
    delay(100);
    SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(100);
    SIM900.println();
    delay(5000); // give module time to send SMS

}

void sendSMSPUERTA()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);
    SIM900.println("ADVERTENCIA!!! SENSOR DE PUERTA DETECTADO, AUTO BLOQUEADO");//
message to send
    delay(100);
    SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(100);
    SIM900.println();
    delay(5000); // give module time to send SMS

}

void sendSMSANTIATRACO()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);
    SIM900.println("ADVERTENCIA!!! SE HA DETECTADO SECUESTRO, AUTO BLOQUEADO");// message
to send
    delay(100);
    SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(100);
    SIM900.println();
    delay(5000); // give module time to send SMS

}

void sendSMSDESBLOQUEO()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);
    SIM900.println("ADVERTENCIA!!! SISTEMA REINICIADO");// message to send

```

```

    delay(100);
    SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(100);
    SIM900.println();
    delay(5000); // give module time to send SMS
}

void sendSMSSEVICIO()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);
    SIM900.println("ADVERTENCIA!!! MODO MANTENIMIENTO");// message to send
    delay(100);
    SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(100);
    SIM900.println();
    delay(5000); // give module time to send SMS
}

void sendSMSSEVICIOSALIDA()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);
    SIM900.println("ADVERTENCIA!!! SALIDA MODO MANTENIMIENTO");// message to send
    delay(100);
    SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(100);
    SIM900.println();
    delay(5000); // give module time to send SMS
}

void sendSMSARRANQUE()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);
    SIM900.println("ADVERTENCIA!!! SE ESTA INTENTANDO ARRANCAR EL AUTO, AUTO
BLOQUEADO");// message to send
    delay(100);
    SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(100);
    SIM900.println();
    delay(5000); // give module time to send SMS
}

void sendSMSVIAJE()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);
    SIM900.println("AUTO DESBLOQUEADO PUEDE INICIAR SU VIAJE");// message to send
    delay(100);
    SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
    delay(100);
    SIM900.println();
    delay(5000); // give module time to send SMS
}

void sendSMSAPAGADO()
{
    SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
    delay(100);
    SIM900.println("AT + CMGS = \"+593983820181\""); //numero de celular
    delay(100);

```

```

SIM900.println("AUTO APAGADO, SE HA REINICIADO EL SISTEMA");// message to send
delay(100);
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
delay(100);
SIM900.println();
delay(5000); // give module time to send SMS
}

void indicador(){
  digitalWrite(buzzer, HIGH);
  delay(100);
  digitalWrite(buzzer, LOW);
}

void botonreset(){

  leertouch();

  if ((cordenada_xx>=9) && (cordenada_xx<=11)){
    if ((cordenada_yy>=42) && (cordenada_yy<=50)){
      Serial.println("BOTON RESETEO PRESIONADO");
      indicador();
      digitalWrite(rojo,LOW);
      digitalWrite(verde,LOW);
      digitalWrite(azul,LOW);
      delay(3000);
      digitalWrite(rojo,HIGH);
      digitalWrite(verde,HIGH);
      digitalWrite(azul,HIGH);
      delay(500);
      leertouch();

      if ((cordenada_xx>=7) && (cordenada_xx<=13)){
        if ((cordenada_yy>=31) && (cordenada_yy<=66)){
          indicador();
          digitalWrite(sirena,LOW);

          digitalWrite(rojo,LOW);
          digitalWrite(verde,LOW);
          digitalWrite(azul,LOW);
          delay(2000);

          Serial.println("1...");

          serNum0=0;
          serNum1=0;
          serNum2=0;
          serNum3=0;
          serNum4=0;
          temporizador=0;
          temporizadora=0;

          digitalWrite(rojo,HIGH);
          digitalWrite(verde,HIGH);
          digitalWrite(azul,HIGH);

          digitalWrite(rele,LOW);

          if (bandera==21){
            sendSMSDESBLOQUEO();
          }

          if (bandera==4){
            sendSMSAPAGADO();
          }

          bandera=0;
        }
      }
      digitalWrite(rojo,HIGH);
      digitalWrite(verde,HIGH);
      digitalWrite(azul,HIGH);
    }
  }
}

```



```

    }
    }

void recibirSMS()
{
  //If a character comes in from the cellular module...
  if(SIM900.available() >0)
  {
    inchar=SIM900.read();
    if (inchar=='e')
    {
      delay(10);

      inchar=SIM900.read();
      if (inchar=='l')
      {
        delay(10);
        inchar=SIM900.read();
        if (inchar=='0')
        {

          indicador();
          digitalWrite(sirena,LOW);

          digitalWrite(rojo,LOW);
          digitalWrite(verde,LOW);
          digitalWrite(azul,LOW);
          delay(2000);

          Serial.println("1...");

          serNum0=0;
          serNum1=0;
          serNum2=0;
          serNum3=0;
          serNum4=0;
          temporizador=0;
          temporizadora=0;

          digitalWrite(rojo,HIGH);
          digitalWrite(verde,HIGH);
          digitalWrite(azul,HIGH);

          digitalWrite(rele,LOW);

          if (bandera==21){
            sendSMSDESBLOQUEO();
          }

          if (bandera==4){
            sendSMSAPAGADO();
          }

          bandera=0;

        }
      }
    }
    SIM900.println("AT+CMGD=1,4"); // delete all SMS
  }
}
}
}

```