

Universidad Internacional del Ecuador



Facultad de Ingeniería Mecánica Automotriz

Tesis de grado para la obtención del Título de Ingeniería en Mecánica Automotriz

**Implementación de un sistema electrónico por medio de NFC y BLUETOOTH
para el encendido, apertura y cierre de puertas del Chevrolet Corsa Evolution
Hatchback**

Héctor Israel Cevallos Velásquez

Director: Ing. Gorky Reyes

Quito, noviembre 2016

Certificación

Yo, HÉCTOR ISRAEL CEVALLOS VELÁSQUEZ, declaro bajo juramento que el trabajo descrito es de mi autoría, que no ha sido presentado anteriormente para ningún grado y que es consultado de la bibliografía detallada.

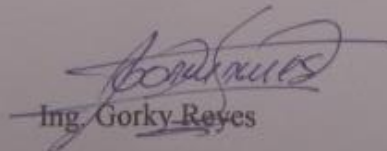
Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo que establece la ley de propiedad intelectual.



Héctor Israel Cevallos Velásquez

1714980784

Yo, Ing. Gorky Reyes, certifico que conozco el señor Héctor Israel Cevallos Velásquez, autor del presente trabajo, siendo responsable de su originalidad y autenticidad.



Ing. Gorky Reyes

Dedicatoria.

La presente tesis, dedico a mis padres y hermanos que me han brindado su apoyo incondicional en el transcurso de esta etapa, han sido mi camino y mi fortaleza para cada día construir mi futuro, me enseñaron que para alcanzar y cumplir con los objetivos que me proponga debía luchar y siempre mirar hacia adelante, observar la meta sin ningún temor, es por esto que no hay cosa más gratificante que dedicar mi esfuerzo de todos estos años a mi familia que es lo máspreciado en mi vida y sentirme orgulloso de este gran logro.

Dedico también a mi enamorada Michelle Recalde, la cual de una u otra manera me ha brindado su ayuda, compartiendo mi sueño y afición por mi carrera, dándome la energía necesaria para no decaer y seguir con mis propósitos tanto en la vida profesional como personal.

Por último, agradezco a las personas que a lo largo de esta etapa estuvieron presentes y que supieron apoyarme en su momento para verme salir adelante.

Agradecimiento

Agradezco a Dios por guiar mi camino siempre, ya que me ha dado la fortaleza necesaria para superar los retos que la vida me ha puesto día a día y en este momento me ayudado a culminar con éxito una etapa trascendental en mi vida.

A mis padres y hermanos, quienes con su amor y sacrificio me brindaron la oportunidad de estudiar la carrera que sería mi futuro, gracias a su apoyo he salido adelante en esto que no es solo una carrera, sino una pasión que disfrutare toda mi vida.

A la Universidad Internacional de Ecuador, a las autoridades y docentes que directa o indirectamente aportaron con sus conocimientos, enseñanzas e infraestructura necesaria en la Carrera de Ingeniería en Mecánica Automotriz, que forman profesionales íntegros y sobre todo personas que sean capaces de salir al mundo y afrontar las adversidades del campo automotor que está siempre a la vanguardia.

Quiero agradecer especialmente a mi Director de Tesis, que con sus conocimientos y colaboración nos fue posible el desarrollo de este proyecto innovador, que va a la vanguardia con la tecnología en el campo Automotriz.

Índice general

Certificación	iii
Dedicatoria	iv
Agradecimiento	v
Índice general	vi
Índice de tablas	xiv
Índice de gráficos	xv
Resumen	xxii
Abstract.....	xxiii
CAPÍTULO 1	1
1.1. Planteamiento, formulación y sistematización del problema.....	1
1.2. Justificación del problema.....	1
1.3. Objetivos.	2
1.3.1. Objetivo General.....	2
1.3.2. Objetivos Específicos.....	2
1.4. Delimitación.....	2
1.4.1. Delimitación de Contenido.	2
CAPÍTULO 2	3
2. Electricidad y electrónica.	3
2.1. Introducción.	3
2.2. Partes de un circuito.	3
2.3. Magnitudes eléctricas.....	4

2.3.1.	Voltaje.....	4
2.3.2.	Intensidad eléctrica.	5
2.3.2.1.	Corriente continua.....	5
2.3.2.2.	Corriente alterna.....	6
2.3.3.	Resistencia.	6
2.3.4.	Campo magnético.....	7
2.4.	Componentes electrónicos.	7
2.4.1.	Circuito integrado.....	7
2.4.1.1.	Circuito integrado y su evolución.	8
2.4.1.2.	Fotolitografía.....	10
2.4.2.	Transistores	10
2.4.3.	Diodo rectificador.	11
2.4.4.	Capacitor.	11
2.4.5.	Micro controlador.....	11
2.4.6.	Led.....	11
2.4.7.	Cristal de cuarzo.....	12
2.4.8.	Arduino UNO.....	12
2.4.8.1.	Tipos de Arduino.	13
2.4.8.2.	Comparación Arduino UNO con otro tipo de placas programables. ...	14
2.4.8.3.	Componentes Arduino UNO.....	16
2.4.8.4.	Elementos Arduino UNO.....	17
	18

.....	22
2.4.8.5. Señales de Arduino.	22
2.4.8.5.1. Señales digitales	22
2.4.8.5.2. Señales analógicas.....	22
2.4.8.6. Características internas Arduino UNO.	23
2.4.9. NFC/RFID.....	24
2.4.9.1. Introducción.....	24
2.4.9.2. Diferenciación.....	25
2.4.10. NFC.....	26
2.4.10.1. Definición.....	26
2.4.10.2. Áreas de contribución desarrolladas con NFC.	27
2.4.11. RFID.....	27
2.4.11.1. Definición.....	27
2.4.11.2. Funcionamiento y modo de trabajo.	28
2.4.11.3. Reseña histórica del RFID.....	28
2.4.11.4. Norma ISO/IEC 14443.....	29
2.4.11.5. Elementos del RFID.	30
2.4.11.5.2. Tarjetas y Tags RFID.	31
2.4.12. Bluetooth.....	32
2.4.12.1. Estados del módulo.....	32
2.4.12.2. Diagrama módulo Bluetooth HC-05.....	33
2.4.13. Botón de encendido SPARCO	33

2.4.14.	Módulo relé de 2 canales	34
2.5.	Componentes eléctricos del vehículo.....	35
	36
CAPÍTULO 3.....		39
3.	Funcionamiento del sistema de Alarma Chevrolet.....	39
3.1.	Alarma Chevrolet.....	39
3.1.1.	Control remoto con clave.....	39
3.1.2.	Módulo de alarma	39
3.1.3.	Actuadores de puertas.....	40
3.1.4.	Bloqueo del volante.....	41
3.1.4.1.	Bloqueo mecánico/manual.....	41
3.1.4.2.	Bloqueo eléctrico	41
3.1.4.2.1.	Componentes del sistema del bloqueo del volante eléctrico.....	43
3.1.5.	Conexiones y funcionamiento de la alarma Chevrolet.....	44
CAPÍTULO 4.....		45
4.	Implementación del circuito electrónico controlado con Arduino UNO, NFC/RFID y Bluetooth.....	45
4.1.	Introducción	45
4.2.	Conexión de actuadores.....	45
4.3.	Programación en Arduino.....	46
4.3.1.	Introducción al entorno de programación.....	46
4.3.2.	Programación de Arduino UNO y componentes RFID y Bluetooth... 48	

4.3.2.1.	Variables.....	51
4.3.2.2.	Asignaciones compuestas.....	52
4.3.2.3.	Operadores de comparación.....	53
4.3.2.4.	Constantes de programación.	54
4.3.2.5.	Control de flujo.	55
4.3.2.6.	Entradas y salidas digitales.	56
4.3.2.7.	Tiempo.	58
4.3.2.8.	Serie.....	58
4.3.2.9.	Código ASCII.....	59
4.3.3.	Comparación Arduino UNO original vs genérico.	61
4.3.4.	Bluetooth HC-05	63
4.3.4.1.	Introducción.....	63
4.3.4.2.	Módulo Bluetooth HC-05 como maestro.....	63
4.3.4.3.	Módulo Bluetooth HC-05 como esclavo.	63
4.3.4.4.	Configuración inicial.	63
4.3.4.5.	Diagrama de conexión Bluetooth HC-05 con Arduino UNO.....	66
4.3.5.	NFC	67
4.3.5.1.	Modo de envío y recepción de datos.....	67
4.3.6.	RFID.....	68
4.3.6.1.	Diagrama tarjeta RFID.....	69
4.3.6.2.	Diagrama tag RFID.	69
4.3.6.3.	Diagrama de conexión módulo RFID con Arduino UNO.....	70

4.3.7. Características necesarias de los dispositivos móviles.....	71
4.4. Sistema de arranque, apertura y cierre de puertas.	72
4.4.1. Bloque del volante Chevrolet Corsa Evolution,	72
4.4.2. Diagrama de conexión botón de encendido.	73
4.4.3. Diagrama de conexión actuadores de puertas con un módulo relé.	74
4.4.4. Diagrama de alimentación del módulo relé y los actuadores de puertas.	75
4.4.5. Diagrama de conexión del módulo relé con el Arduino UNO del contacto del vehículo.	76
4.4.6. Diagrama de conexión del contacto del vehículo con el módulo relé y la bocina.	77
4.4.7. Diagrama de conexión y alimentación de la cajuela del vehículo con el módulo relé y el actuador.	78
4.4.8. Diagrama de comunicación entre el Arduino UNO de las puertas con el Arduino UNO del contacto del vehículo.	79
4.4.9. Diagrama de comunicación entre el Arduino UNO de las puertas con el Arduino UNO de la cajuela.	80
4.4.10. Diagrama de comunicación entre el Arduino UNO de la cajuela con el Arduino UNO del contacto del vehículo.	81
4.5. Diseño de la aplicación móvil en App Inventor.....	82
4.5.1. Introducción.	82
4.5.2. Inicio en App Inventor.	84
4.5.3. Entorno de programación.	86

4.5.4. Programación visual.....	87
4.5.5. Programación en bloques.	88
4.5.6. Diseño de la App.....	89
4.5.7. Pre-requisitos para el uso de la App “CORSA_C”.	90
4.5.8. Aplicación “CORSA_C”.....	93
4.6. Diseño de las cajas de los módulos.....	94
4.6.1. SolidWorks.....	94
4.6.1.1. Diseño caja Arduino UNO.....	95
4.6.1.2. Diseño caja módulo RFID.	95
4.6.1.3. Diseño caja módulo Bluetooth HC-05.....	95
4.6.1.4. Diseño caja módulo relé de 2 canales.	96
4.6.2. Corte Laser.	96
4.6.3. Diseños finales.....	97
CAPÍTULO 5.....	98
5. Pruebas de funcionamiento.....	98
5.1. Pruebas de conexión mediante Bluetooth.	98
5.1.1. Conexión al Bluetooth	98
5.1.2. Prueba del contacto y apagado del vehículo	99
5.1.2.1. Vehículo en contacto.....	99
5.1.2.2. Vehículo apagado.....	100
5.1.3. Prueba de la apertura y cierre de puertas	100
5.1.3.1. Apertura de puertas	100

5.1.3.2. Cierre de puertas.....	101
5.1.4. Prueba de la cajuela	101
5.1.4.1. Apertura de la cajuela.....	101
5.2. Pruebas de conexión mediante NFC/RFID.	102
5.2.1. Prueba del contacto y apagado del vehículo	102
5.2.1.1. Vehículo en contacto.....	102
5.2.1.2. Vehículo apagado.....	103
5.2.1.3. Intruso.....	104
5.2.2. Prueba de apertura y cierre de puertas	105
5.2.2.1. Apertura de puertas	106
5.2.2.2. Cierre de puertas.....	107
5.2.2.3. Intruso.....	108
5.2.3. Prueba de la cajuela	109
5.2.3.1. Cajuela abierta.....	109
5.2.3.2. Cajuela cerrada.....	110
5.2.3.3. Intruso.....	111
CONCLUSIONES	113
RECOMENDACIONES.....	114
BIBLIOGRAFÍA	115
ANEXOS	117

Índice de tablas

Tabla 2.1 Evolución de los sistemas integrados.....	9
Tabla 2.2 Tipos de Arduino.....	13
Tabla 2.3 Arduino UNO vs Otras placas programables.....	14
Tabla 2.4 Reseña histórica del RFID.....	28
Tabla 2.5 Comunicación SPI con el microcontrolador.....	31
Tabla 4.1 Código ASCII, Hexadecimal y su simbología.....	60
Tabla 4.2 Arduino UNO original vs genérico.....	61
Tabla 4.3 Comando AT.....	64
Tabla 4.4 Características de los Smartphone.....	71

Índice de gráficos

Figura 2.1 Esquema elemental de un átomo	3
Figura 2.2 Circuito eléctrico	4
Figura 2.3 Medición de voltaje de batería en el vehículo.....	4
Figura 2.4 Corriente eléctrica	5
Figura 2.5 Corriente Continua	5
Figura 2.6 Corriente alterna.....	6
Figura 2.7 Representación de la resistencia.....	6
Figura 2.8 Campo magnético.....	7
Gráfico 2.9 Circuito integrado.....	8
Figura 2.10 Circuito integrado lijado	10
Figura 2.11 Transistor.....	10
Figura 2.12 Logotipo Arduino.....	12
Figura 2.13 Arduino UNO.....	12
Figura 2.14 Componentes de Arduino UNO	16
Figura 2.15 Elementos Arduino UNO	17
Figura 2.16 Elementos Arduino UNO	18
Figura 2.17 Microcontrolador ATMEGA 328P-PU	22
Figura 2.18 Microcontrolador Arduino UNO.....	22
Figura 2.19 Tecnología inalámbrica.....	25
Figura 2.20 Tecnología inalámbrica.....	26

Figura 2.21 Tecnología inalámbrica.....	26
Figura 2.22 Áreas de desarrollo con NFC	27
Figura 2.23 Tecnología RFID.....	27
Figura 2.24 Lector, tarjeta y tag RFID	28
Figura 2.25 Norma ISO/IEC.....	29
Figura 2.26 Lector RFID diseñado en SolidWorks	30
Figura 2.27 Integrado MFRC522	30
Figura 2.28 Diagrama módulo Bluetooth HC-05	33
Figura 2.29 Logotipo SPARCO.....	33
Figura 2.30 Botón de encendido SPARCO	34
Figura 2.31 Módulo relé de 2 canales.....	35
Figura 2.32 Proceso interno del electrolito en la batería	35
Figura 2.33 Descripción de la batería del vehículo	36
Figura 2.34 Conexión del alternador	37
Figura 2.35 Funcionamiento y arranque del vehículo	38
Figura 3.1 Control de alarma Chevrolet.....	39
Figura 3.2 Módulo de Alarma Chevrolet.....	40
Figura 3.3 Actuador de puertas del vehículo.....	40
Figura 3.4 Parte interna del actuador de puertas	41
Figura 3.5 Bloqueo eléctrico BMW	42
Figura 3.6 Componentes del bloqueo eléctrico del volante BMW.....	43
Figura 3.7 Conexiones Alarma Chevrolet	44

Figura 4.1 Actuador de cajuela.....	46
Figura 4.2 Inicio del programa Arduino.....	46
Figura 4.3 Diagrama de flujo de la programación en Arduino.....	47
Figura 4.4 Tomado del código de programación de inicio por defecto.....	48
Figura 4.5 Tomado del código de programación de las puertas.....	49
Figura 4.6 Tomado del código de programación de las puertas.....	49
Figura 4.7 Tomado del código de programación de las puertas.....	50
Figura 4.8 Tomado del código de programación de las puertas.....	50
Figura 4.9 Tomado del código de programación de las puertas.....	51
Figura 4.10 Tomado del código de programación de las puertas.....	51
Figura 4.11 Código de programación para declarar variables.....	52
Figura 4.12 Tomado del código de programación de las puertas.....	52
Figura 4.13 Tomado del código de programación de las puertas.....	52
Figura 4.14 Tomado del código de programación de las puertas.....	53
Figura 4.15 Tomado del código de programación de las puertas.....	53
Figura 4.16 Tomado del código de programación de las puertas.....	53
Figura 4.17 Tomado del código de programación de las puertas.....	54
Figura 4.18 Tomado del código de programación de las puertas.....	54
Figura 4.19 Tomado del código de programación de las puertas.....	55
Figura 4.20 Tomado del código de programación de las puertas.....	55
Figura 4.21 Tomado del código de programación de las puertas.....	56
Figura 4.22 Tomado del código de programación de las puertas.....	56

Figura 4.23 Tomado del código de programación del contacto del vehículo.....	57
Figura 4.24 Tomado del código de programación del contacto del vehículo.....	57
Figura 4.25 Tomado del código de programación de las puertas	57
Figura 4.26 Tomado del código de programación de las puertas	58
Figura 4.27 Tomado del código de programación del contacto del vehículo.....	58
Figura 4.28 Tomado del código de programación del contacto del vehículo.....	59
Figura 4.29 Programación con código ASCII	59
Figura 4.30 Chevrolet Corsa Evolution equipado con Arduino	61
Figura 4.31 Arduino UNO original vs genérico	62
Figura 4.32 Módulo Bluetooth HC-05 diseñado en SolidWorks	63
Figura 4.33 Emparejamiento al Bluetooth CORSA_C desde dispositivo móvil.....	65
Figura 4.34 Diagrama de conexión Bluetooth HC-05 con Arduino UNO	66
Figura 4.35 Partes internas de un dispositivo NFC	67
Figura 4.36 Modo de envío y recepción de datos.....	67
Figura 4.37 Tecnología inalámbrica.....	68
Figura 4.38 Parte interna de una tarjeta RFID.....	69
Figura 4.39 Parte interna de un Tag RFID	69
Figura 4.40 Diagrama de conexión módulo RFID con Arduino UNO	70
Figura 4.41 Bloqueo mecánico/manual del volante	72
Figura 4.42 Diagrama de conexión botón de encendido	73
Figura 4.43 Diagrama de conexión actuadores de puertas con un módulo relé	74
Figura 4.44 Diagrama de alimentación del módulo relé y los actuadores de puertas	75

Figura 4.45 Diagrama de conexión del módulo relé con Arduino UNO del contacto del vehículo	76
Figura 4.46 Diagrama de conexión del contacto del vehículo con el módulo relé y la bocina.....	77
Figura 4.47 Diagrama de conexión y alimentación de la cajuela del vehículo con el módulo relé y el actuador.	78
Figura 4.48 Diagrama de comunicación entre Arduino UNO de las puertas con el Arduino UNO del contacto del vehículo	79
Figura 4.49 Diagrama de comunicación entre Arduino UNO de las puertas con el Arduino UNO de la cajuela	80
Figura 4.50 Diagrama de comunicación entre Arduino UNO de la cajuela con el Arduino UNO del contacto del vehículo.	81
Figura 4.51 Bienvenido a MIT App Inventor	82
Figura 4.49 Página de inicio App Inventor.....	84
Figura 4.52 Inicio de sesión o registro con una cuenta de Google	84
Figura 4.51 Barra de herramientas de la página de inicio de App Inventor	85
Figura 4.53 Página de inicio de proyectos.....	85
Figura 4.54 Entorno de programación.....	86
Figura 4.55 Programación visual.....	87
Figura 4.56 Programación en bloques	88
Figura 4.57 Diseño de la App.....	89
Figura 4.58 Guardar proyecto.....	90

Figura 4.59 Conectar AI Companion	90
Figura 4.60 Descargar MIT AI2 Companion en el Smartphone	91
Figura 4.61 Escanear código QR emitido por AI Compannion	91
Figura 4.62 Emulación de la App CORSA_C	92
Figura 4.63 Guardar App en formato. apk.....	92
Figura 4.64 Aplicación CORSA_C	93
Figura 4.65 SolidWorks.....	94
Figura 4.66 Diseño caja de Arduino UNO en SolidWorks	95
Figura 4.67 Diseño caja de módulo RFID en SolidWorks	95
Figura 4.68 Diseño caja de módulo Bluetooth HC-05 en SolidWorks	95
Figura 4.69 Diseño caja de módulo relé de 2 canales	96
Figura 4.70 Corte laser	96
Figura 4.71 Máquina de corte laser	97
Figura 4.72 Figuras listas en pantalla para la máquina de corte laser	97
Figura 4.73 Cajas cortadas en laser y ensambladas.....	97
Figura 5.1 Prueba de conexión con Bluetooth CORSA_C.....	99
Figura 5.2 Prueba del contacto del vehículo mediante Bluetooth	99
Figura 5.3 Prueba apagado del vehículo mediante Bluetooth	100
Figura 5.4 Prueba de apertura de puertas mediante Bluetooth.....	100
Figura 5.5 Prueba de cierre de puertas mediante Bluetooth	101
Figura 5.6 Prueba de apertura de cajuela mediante Bluetooth	101
Figura 5.7 Prueba de contacto y apagado del vehículo mediante NFC/RFID.....	102

Figura 5.8 Prueba de vehículo en contacto mediante NFC/RFID	103
Figura 5.9 Prueba de apagado del vehículo mediante NFC/RFID	104
Figura 5.10 Prueba de intruso en el contacto del vehículo mediante NFC/RFID	105
Figura 5.11 Prueba de apertura y cierre de puertas mediante NFC/RFID.....	105
Figura 5.12 Prueba de apertura puertas mediante NFC/RFID.....	106
Figura 5.13 Prueba de cierre puertas mediante NFC/RFID.....	107
Figura 5.14 Prueba de intruso en puertas mediante NFC/RFID.....	108
Figura 5.15 Prueba de la cajuela mediante NFC/RFID	109
Figura 5.16 Prueba de la cajuela abierta mediante NFC/RFID	110
Figura 5.17 Prueba de la cajuela cerrada.....	111
Figura 5.18 Prueba de intruso en la cajuela mediante NFC/RFID	112

Resumen

El proyecto descrito en el presente documento está basado en la implementación de un mecanismo de seguridad para apertura y cierre de puertas por medio de un dispositivo móvil con el sistema NFC/RFID y Bluetooth para el Chevrolet Corsa Evolution, adicionalmente de un botón de encendido con inmovilizador, que sustituye al switch de encendido convencional que necesita una llave.

La presente propuesta consiste en cambiar el sistema convencional de la alarma Chevrolet por un mecanismo electrónico que haga uso de un dispositivo móvil o tarjetas electrónicas con tecnología NFC/RFID, que realicen la función de llave digital, a partir del uso de la placa de ARDUINO UNO, con sus conectores Bluetooth y RFID estarán colocados en los parabrisas delantero y trasero del vehículo, así como en el tablero de instrumentos.

Por otro lado, el sistema de seguridad electrónica, implementado en el Chevrolet Corsa Evolution estará diseñado con estrictos estándares de codificación que serán imposibles de clonar, con el fin de impedir la sustracción del vehículo.

Finalmente, en el proyecto se desarrollará una aplicación personalizada, diseñada y programada en el entorno de App Inventor, con el fin de cumplir con los propósitos de la apertura y cierre de puertas del vehículo Chevrolet Corsa Evolution, como mando a distancia, además del contacto del mismo para poder hacer uso de los distintos accesorios como son el radio, luces, aire acondicionado, etc.

Abstract

The project described in this document is based on the implementation of a safety mechanism for opening and closing doors through a mobile device with NFC/RFID and Bluetooth system for Chevrolet Corsa Evolution, additionally the implementation includes a power button with immobilizer which replaces the original key switch.

On the other hand, the principal objective of this investigation is to change the conventional Chevrolet alarm system by an electronic mechanism that makes use of a mobile device or electronic cards with NFC/RFID technology, which performs the digital key function, also the mobile device will be connected to ARDUINO UNO plate, and its connectors Bluetooth and RFID will be placed in the front and rear windshields of the vehicle, as well as on the dashboard.

On the other hand, the electronic security system implemented in the Chevrolet Corsa Evolution will be designed with strict coding standards that will be impossible to clone, in order to prevent the theft of the vehicle.

Finally, the project will develop an app designed and programmed according the platform of App Inventor as remote control in order to reach the purpose of the opening and closing doors of the vehicle Chevrolet Corsa Evolution and also the use of the radio, lights, air conditioning.

CAPÍTULO 1

1.1. Planteamiento, formulación y sistematización del problema.

Implementación de un mecanismo de seguridad para apertura y cierre de puertas por medio de un dispositivo móvil con el sistema NFC y Bluetooth para el Chevrolet Corsa Evolution, adicionalmente será incluido un botón de encendido con inmovilizador, que sustituye al switch de encendido convencional que necesita una llave.

El modelo será implementado mediante una placa de ARDUINO UNO, con el conector Bluetooth y dispositivos NFC/RFID respectivos, que estarán colocados en los parabrisas delantero y trasero del vehículo, así como en el tablero de instrumentos.

El sistema de seguridad electrónica, que será implementado en el Chevrolet Corsa Evolution estará diseñado con estrictos estándares de codificación que serán imposibles de clonar, con el fin de impedir la sustracción del vehículo.

1.2. Justificación del problema.

Cambiar el sistema convencional de la alarma Chevrolet por un mecanismo electrónico que haga uso de un dispositivo móvil o tarjetas electrónicas con tecnología NFC/RFID permite abrir y cerrar las puertas del vehículo de una manera fácil, práctica y segura, mediante sistemas BLUETOOTH Y NFC/RFID; lo cual reemplazará las llaves convencionales, por un Smartphone compatible que realice la función de llave digital.

El contacto convencional que lo realiza la llave, es reemplazado por el mecanismo NFC/RFID y BLUETOOTH para que permita arrancar el vehículo y hacer uso de los accesorios (radio, aire acondicionado, encendedor, entre otros).

1.3. Objetivos.

1.3.1. Objetivo General.

Implementar un sistema tecnológico para seguridad vehicular, basado en el diseño de un modelo de componentes para la apertura y cierre de las puertas del vehículo, que considera la sustitución del encendido convencional por un sistema de encendido electrónico con inmovilizador.

1.3.2. Objetivos Específicos.

- ✚ Diseñar un mecanismo que facilite el accionamiento de los seguros del vehículo, a través de un mando Bluetooth y NFC/RFID.
- ✚ Desarrollar un sistema de codificación que permita el encendido del vehículo por medio de un botón con inmovilizador.
- ✚ Desarrollar una aplicación móvil para sistemas ANDROID

1.4. Delimitación.

1.4.1. Delimitación de Contenido.

El alcance de este proyecto va enfocado específicamente al campo de la seguridad de la alarma del Chevrolet Corsa Evolution, que abarca diversos aspectos para la implementación del nuevo mecanismo con tecnología ARDUINO, NFC/RFID, BLUETOOTH y encendido electrónico por botón.

CAPÍTULO 2

2. Electricidad y electrónica.

2.1. Introducción.

La electricidad se origina a partir de una partícula de tamaño muy pequeño denominada electrón y forma parte del átomo. La distribución de los electrones, protones y neutrones en el átomo se distribuye de la siguiente manera:

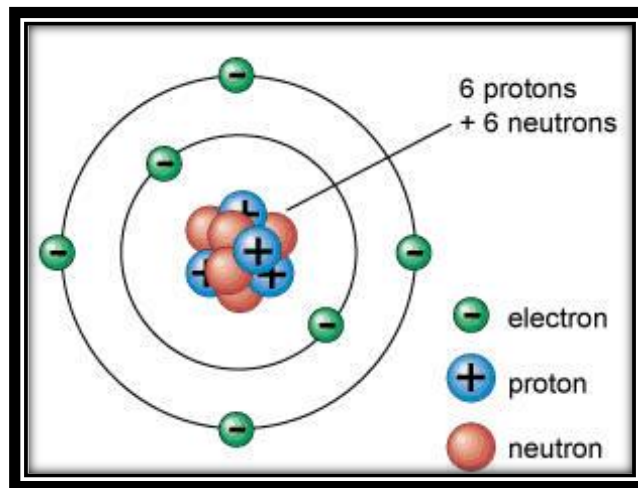


Figura 2.1 Esquema elemental de un átomo

Fuente: http://historiaybiografias.com/el_atomo/

2.2. Partes de un circuito.

- ❖ **Conductor:** Es un hilo de material conductor a través del cual circulan elementos portadores de carga (corriente eléctrica), que permite la unión de dos o más elementos de un circuito.
- ❖ **Generador:** Es una máquina que transforma la energía mecánica en energía eléctrica.
- ❖ **Nodo:** Es el punto de unión de varios elementos eléctricos como: resistencias, condensadores, entre otros.

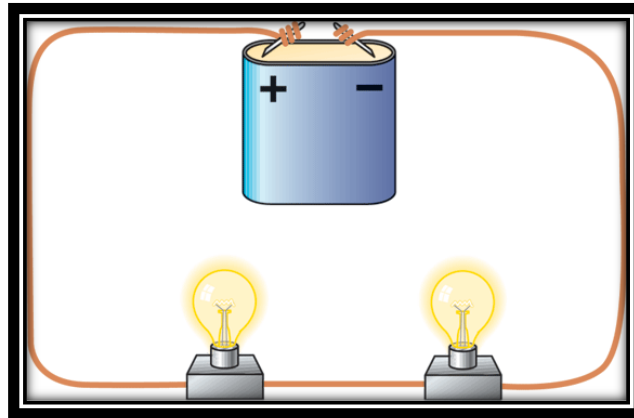


Figura 2.2 Circuito eléctrico

Fuente: <http://corrienteelectronica-yeckan.blogspot.com/p/circuitos-electricos.html>

2.3. Magnitudes eléctricas

2.3.1. Voltaje.

Es la diferencia de potencial que existe entre dos puntos, que permite el viaje de los electrones a lo largo de un circuito eléctrico y su unidad es el voltio.

En la Figura 2.3 se puede apreciar la medición de voltaje en la batería del vehículo, se conecta el cable negro al borne negativo y el cable rojo al borne positivo de la misma.



Figura 2.3 Medición de voltaje de batería en el vehículo

Fuente: <https://autopartesgonzalez.com/blog/alternadores-generadores-autopartes-usadas-gonzalez.html>

2.3.2. Intensidad eléctrica.

Según Artero (2013), dice que la intensidad eléctrica: “Es una magnitud eléctrica que se define como la cantidad de carga eléctrica que pasa en un determinado tiempo a través de un punto en concreto de un material conductor.” (Artero, 2013)

Por lo tanto, es la circulación de electrones a través de un circuito cerrado como se puede observar en la Figura 2.4, van de un polo negativo a un polo positivo y puede ser medido en Amperios (A).

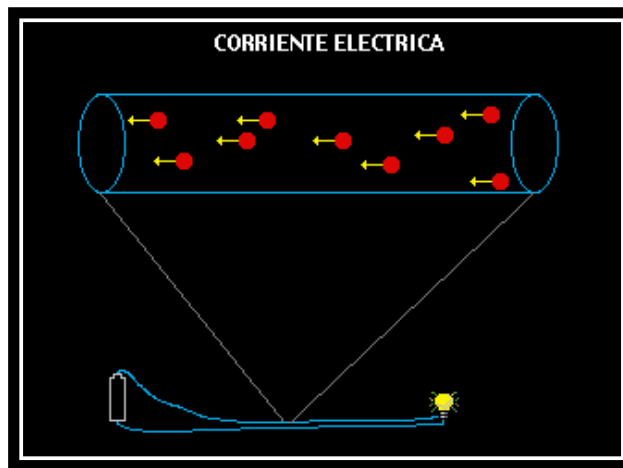


Figura 2.4 Corriente eléctrica

Fuente: <http://www.taringa.net/comunidades/ciencia-con-paciencia/6340631/I-Electrodinamica-Todo-lo-que-necesitas-saber.html>

2.3.2.1. Corriente continua

El flujo de electrones se produce en una sola dirección, es decir, cuándo va solamente de polo negativo a polo positivo.

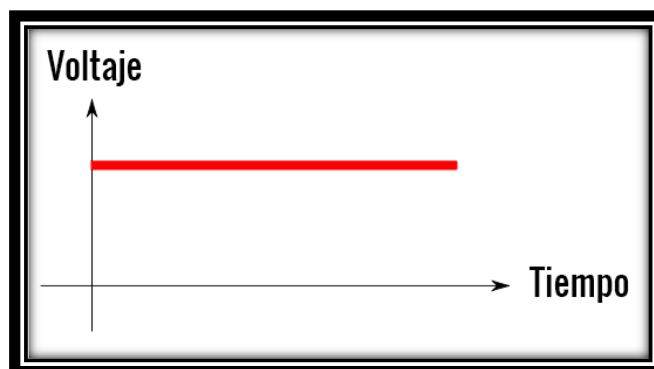


Figura 2.5 Corriente Continua

Fuente: <http://www.taringa.net/post/ciencia-educacion/18602871/Que-es-Corriente-alterna-y-Corriente-Continua.html>

2.3.2.2. Corriente alterna

El flujo de electrones cambia de polaridad en función del tiempo, es decir, viajan desde el polo negativo al positivo y se establece un sentido de dirección, mientras que si viajan de polo positivo a negativo se invierte el sentido de dirección.

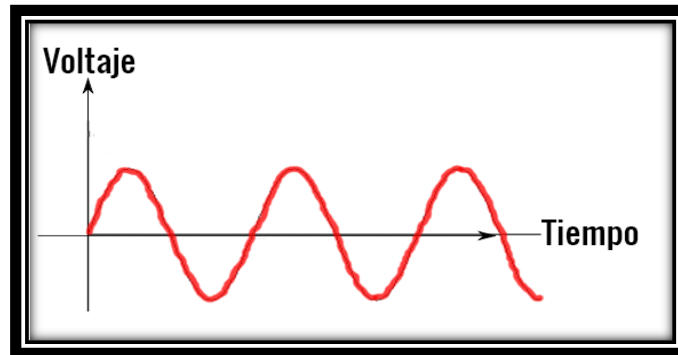


Figura 2.6 Corriente alterna

Fuente: <http://www.taringa.net/post/ciencia-educacion/18602871/Que-es-Corriente-alterna-y-Corriente-Continua.html>

2.3.3. Resistencia.

Es la oposición de electrones que se genera en un circuito eléctrico cerrado, al paso, y se mide en Ohmios.

Según Artero (2013), dice que: “Cuanto mayor sea la resistencia de este componente, más dificultad tendrán los electrones para atravesarlo, hasta incluso el extremo de imposibilitar la existencia de electricidad.” (Artero, 2013)

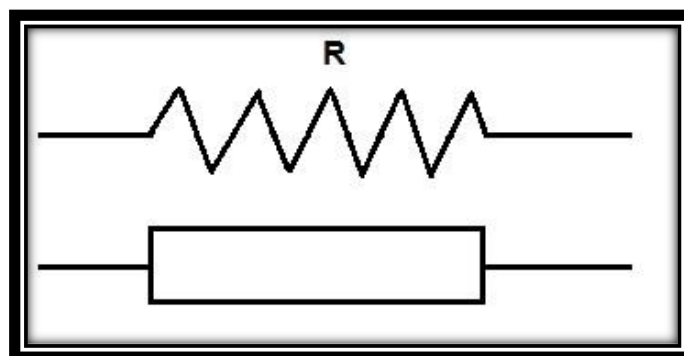


Figura 2.7 Representación de la resistencia

Fuente: <https://ingelibreblog.wordpress.com/2014/01/29/introduccion-a-la-teoria-de-circuitos-conceptos-fundamentales-ii/>

2.3.4. Campo magnético.

Campo magnético es la fuerza que se genera producto del movimiento de cargas eléctricas, dicho flujo aumenta o decrece según la distancia que exista a la fuente que provoca el campo.

Según Boylestad (2011), el campo magnético: “Puede ser representado por líneas de flujo magnético parecidas a las líneas de flujo eléctrico. Sin embargo, las líneas de flujo magnético no tienen puntos de origen o terminación como las líneas de flujo eléctrico, sino que existen lazos continuos.” (Boylestad, 2011)

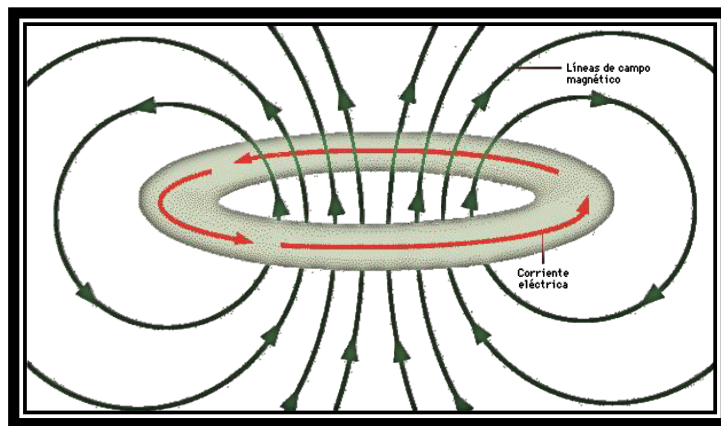


Figura 2.8 Campo magnético

Fuente: http://patricioconcha.ubb.cl/maquinas/paginas/c_fund_gme_materia.html

2.4. Componentes electrónicos.

Los componentes electrónicos son aquellos elementos que forman parte de un circuito electrónico.

2.4.1. Circuito integrado.

Un circuito integrado, está construido a base de un soporte de silicio, que se encuentra internamente en un encapsulado negro con pines de metal, definidos cada uno con su nomenclatura.

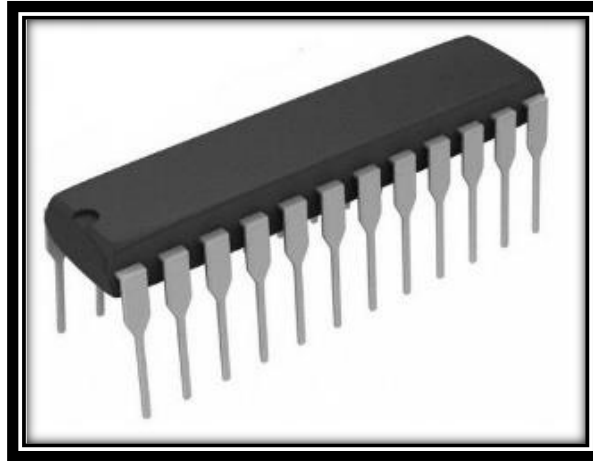


Gráfico 2.9 Circuito integrado

Fuente: <http://timerime.com/es/evento/3844093/Evolucin+de+los+circuitos+integrados/>

2.4.1.1. Circuito integrado y su evolución.

Un circuito integrado también conocido como microchip, por su capacidad de almacenar pequeños datos que lleva información; está hecho a base de material semi-conductor en el cual están circuitos electrónicos que se fabrican comúnmente mediante fotolitografía.

Tabla 2.1 Evolución de los sistemas integrados.

SSI	MSI	LSI	VLSI	ULSI	GLSI
Small Scale Integration	Medium Scale Integration	Large Scale Integration	Very Large Scale Integration	Ultra Large Scale Integration	Giga Large Scale Integration
1960-1963	1969-1974	1975-1984	1985-1989	1900-2015	2016
Contiene 100 elementos	Contiene entre 100 a 3000 elementos	Contiene de 1001 a 10000 transistores	Contiene de 10001 a 100000 transistores	Contiene 75 millones de transistores	Contiene más de 1 millón de transistores
Cumplía con funciones básicas	Utilizado en radios, televisores y calculadoras	Contiene de 100 a 100000 componentes	Es el primer microprocesador	Utilizados para crear micro-procesadores complejos	-
Empleado para las primeras computadoras	Reemplazo a los SSI	Utilizado en grandes computadoras llamada mainframes	Tecnología de semi- conductores	-	-

Fuente: Israel Cevallos

2.4.1.2. Fotolitografía.

La fotolitografía es el proceso por el cual se realiza la fabricación de circuitos integrados y semi-conductores, que transfiere un patrón a una oblea desde una foto-máscara a la cual se la llama retícula.

El silicio es el material con el cual están recubiertos los circuitos integrados y es procesado en la industria en forma de obleas, empleándose como sustrato litográfico, que logra el grabado de un dibujo sobre el silicio.

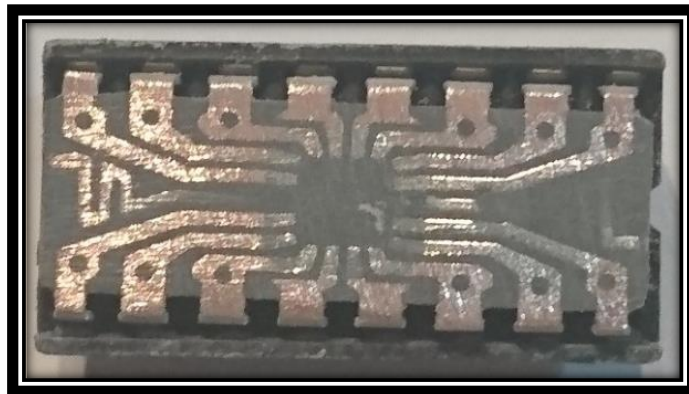


Figura 2.10 Circuito integrado lijado
Fuente: Israel Cevallos

2.4.2. Transistores

Un transistor es un elemento electrónico que posee material semiconductor que puede trabajar como un switch electrónico o como un amplificador de señales electrónicas.

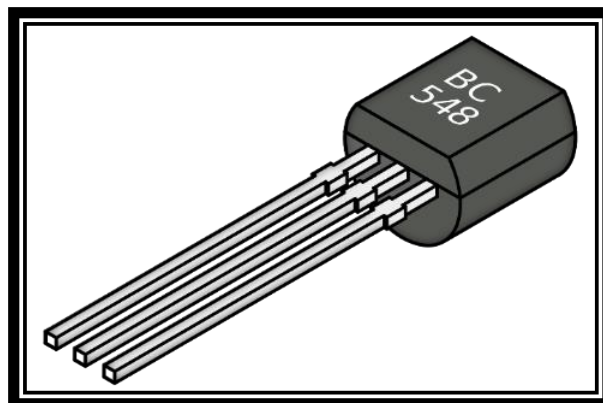


Figura 2.11 Transistor
Fuente: <http://www.kitelectronica.com/2016/02/como-probar-un-transistor.html>

2.4.3. Diodo rectificador.

Según Marticonera (2008), dice que: “El diodo es un componente electrónico que permite el paso de la corriente en un sólo sentido. Cuando un diodo tiene polarización directa, es decir cuando el positivo de la fuente de alimentación va conectado al ánodo y el negativo al cátodo, permite la circulación de la corriente eléctrica. Si se aplica al diodo una polaridad inversa, es decir el positivo de la fuente al cátodo y el negativo al ánodo, no permite la circulación de la corriente eléctrica. La polaridad del diodo viene señalada en el cuerpo de éste por medio de una banda, la cual nos indica el extremo correspondiente al cátodo.” (Marticonera, 2008)

2.4.4. Capacitor.

Es un elemento capaz de almacenar energía eléctrica, es un componente pasivo que se carga en el momento en que es energizado con una fuente eléctrica.

2.4.5. Micro controlador

“Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Se puede decir con toda propiedad que un microcontrolador es una microcomputadora completa encapsulada en un circuito integrado.” (ELECTRONICA ESTUDIO.COM, s.f.)

2.4.6. Led

Un led es un diodo emisor de luz, convierte en luz la corriente suministrada de bajo voltaje que pasa por su chip.

El led constituye un elemento semi-conductor, el color de luz que emite el led, depende del material conductor empleado en la fabricación del mismo.

2.4.7. Cristal de cuarzo

Es un elemento usado comúnmente en electrónica digital y radiofrecuencia, su principal función es la de producir tensiones alternas estabilizando la frecuencia.

2.4.8. Arduino UNO

Arduino es una plataforma que permite crear prototipos con código abierto basados en la sencillez de software y hardware.



Figura 2.12 Logotipo Arduino
Fuente: <https://vimeo.com/90398978>

Por otro lado, tiene la capacidad de recibir mediante sus entradas impulsos desde diferentes sensores que se le puede adaptar, de esta forma se guardará la información de la programación en su microcontrolador ATMEGA 328P-PU en el Lenguaje de Programación propio de Arduino.



Figura 2.13 Arduino UNO
Fuente: <https://solarbotics.com/product/50450/>

2.4.8.1. Tipos de Arduino.

Tabla 2.2 Tipos de Arduino

Modelo	Micro-controlador	Voltaje de entrada	Voltaje del sistema	Frecuencia de reloj	Entradas y salidas digitales	Entradas analógicas	Señal por anchura de pulso	Memoria Flash	Interfaz de programación	Dimensiones (cm)
ARDUINO UNO	ATMEGA328P	7-12 v	5 v	16 MHz	14	6	6	32 kb	USB	7,5 x 5,3
ARDUINO MEGA	ATMEGA2560	7-12 v	5 v	16 MHz	54	16	14	256 kb	USB	10,16 x 5,3
ARDUINO LEONARDO	ATMEGA32U4	7-12 v	5 v	16 MHz	20	12	7	32 kb	USB	6,9 x 5,3
ARDUINO DUE	AT91SAM3X8E	5-12 v	3,3 v	84 MHz	54	12	12	512 kb	USB	10,16 x 5,3
ARDUINO NANO	ATMEGA168	7-12 v	5 v	16 MHz	14	8	6	2 kb	USB	1,8 x 4,3
ARDUINO MINI	ATMEGA 328	7-9 v	5 v	16 MHz	14	6	8	32 kb	Cable Serial	3 x 1,8
ARDUINO ADK	ATMEGA2560	7-12 v	5 v	16 MHz	54	16	15	256 kb	USB	3,3 x 1,8
ARDUINO FIO	ATMEGA 328P	3,3-12 v	3,3 v	8 MHz	14	8	6	32 Kb	USB	2,8 x 2,5

Fuente: Israel Cevallos

2.4.8.2. Comparación Arduino UNO con otro tipo de placas programables.

Tabla 2.3 Arduino UNO vs Otras placas programables

TABLA COMPARATIVA DE PLACAS ELECTRÓNICAS PROGRAMABLES				
TIPOS DE PLACAS		ARDUINO UNO	PARALLAX BASIC STAMP	NETMEDIA BX-24
SISTEMAS OPERATIVOS	Windows	X	X	WINDOWS XP
	Compatible con Windows 10	X	X	-
	Mac	X	-	-
	Linux	X	-	-
SOFTWARE DE PROGRAMACIÓN	Software disponible para descarga	X	X	-
	Software disponible para comprar	-	-	\$20
CARACTERÍSTICAS DE FUNCIONAMIENTO	Voltaje de trabajo (V)	7-12 V	9 V	5-12 V
	Amperaje de trabajo (mA)	46 mA	300mA	100 mA
	Lenguaje de programación	Arduino	Basic Stamp	Basic X
	Conexión con PC	Cable tipo AB	Cable Micro USB	Cable VGA
PVP	Precio	\$50	\$80	\$140

Fuente: Israel Cevallos

Como se muestra en la Tabla 2.4 Arduino posee mayores beneficios y ventajas, gracias a la multiplataforma en la cual se ejecuta como el 100% de compatibilidad con todos los sistemas operativos, que se encuentran disponibles hoy en el mercado, del mismo modo su software de programación está disponible para descargarse gratuitamente desde la Página Web Oficial de Arduino (<https://www.arduino.cc/en/Main/Software>).

En cuanto al entorno de trabajo, la programación se realiza en base a librerías predefinidas, lo que hace más fácil introducir los parámetros necesarios para cada requerimiento; de esta manera se transmite los datos a la memoria de la placa de Arduino con el Cable tipo AB usado comúnmente en las impresoras.

Por último, el precio de la placa Arduino UNO se encuentra en \$50,00 con un costo menor en relación a las demás placas detalladas en la Tabla 2.4.

2.4.8.3. Componentes Arduino UNO.

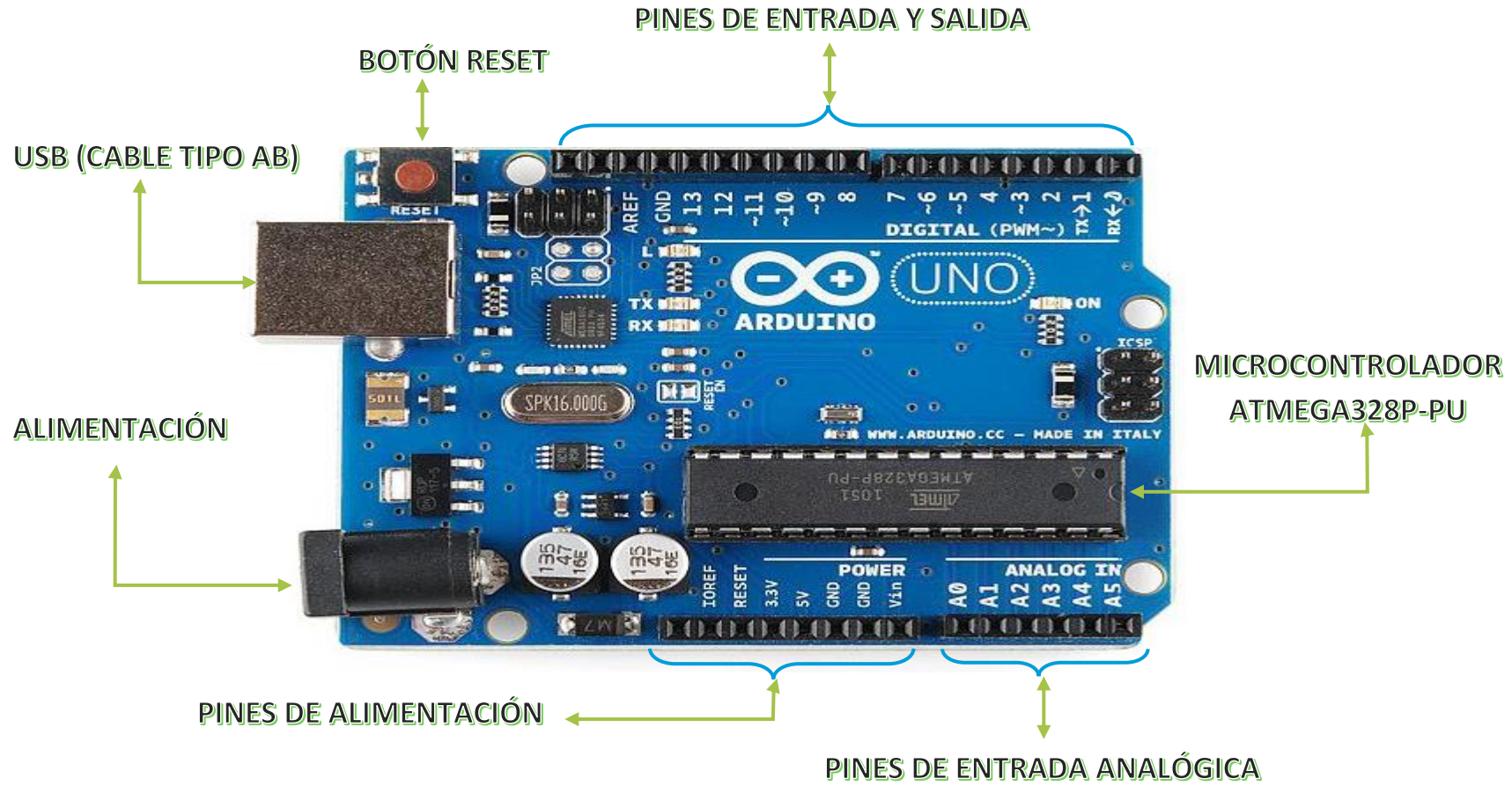


Figura 2.14 Componentes de Arduino UNO

Fuente: Israel Cevallos

2.4.8.4. Elementos Arduino UNO

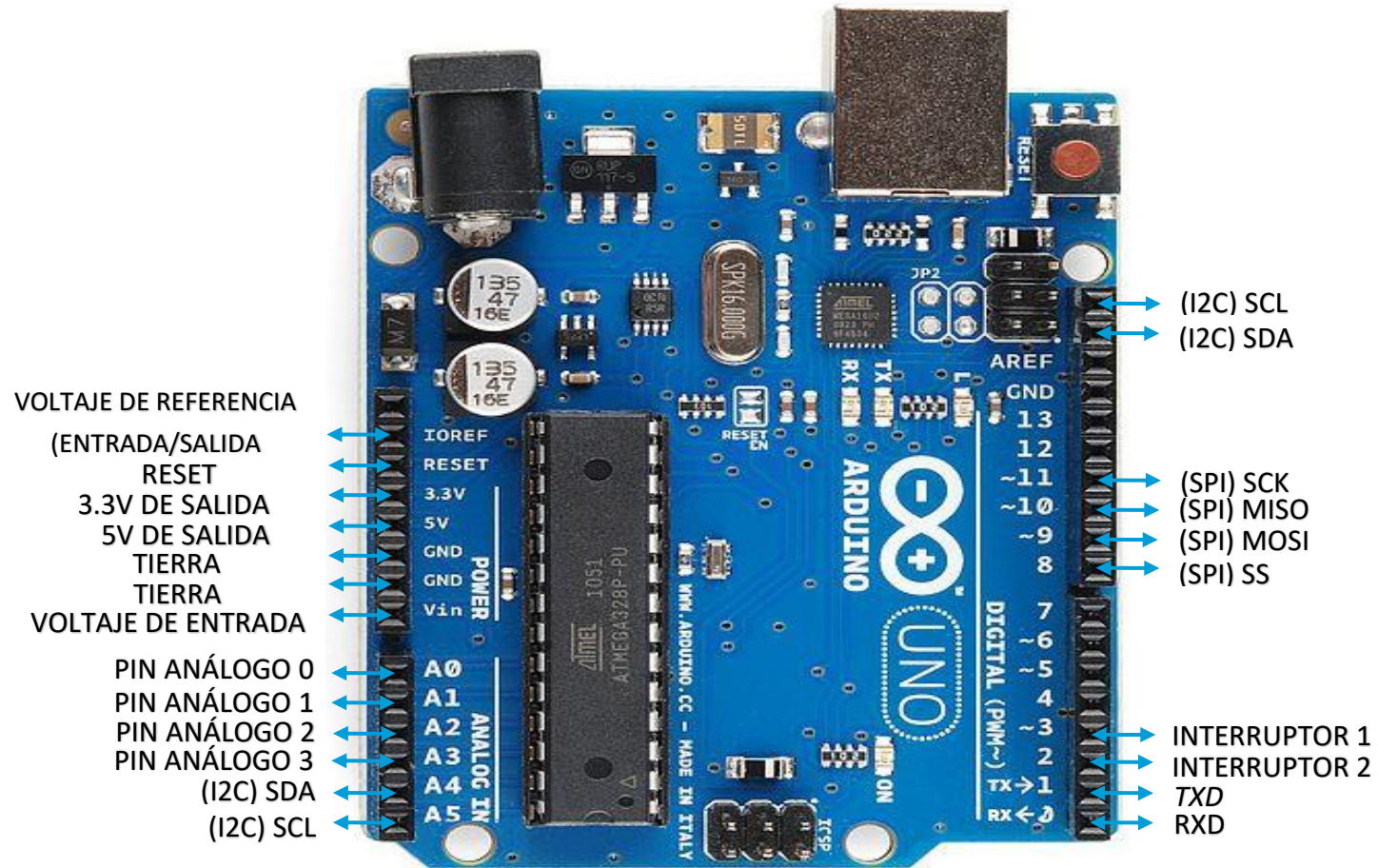


Figura 2.15 Elementos Arduino UNO

Fuente: Israel Cevallos

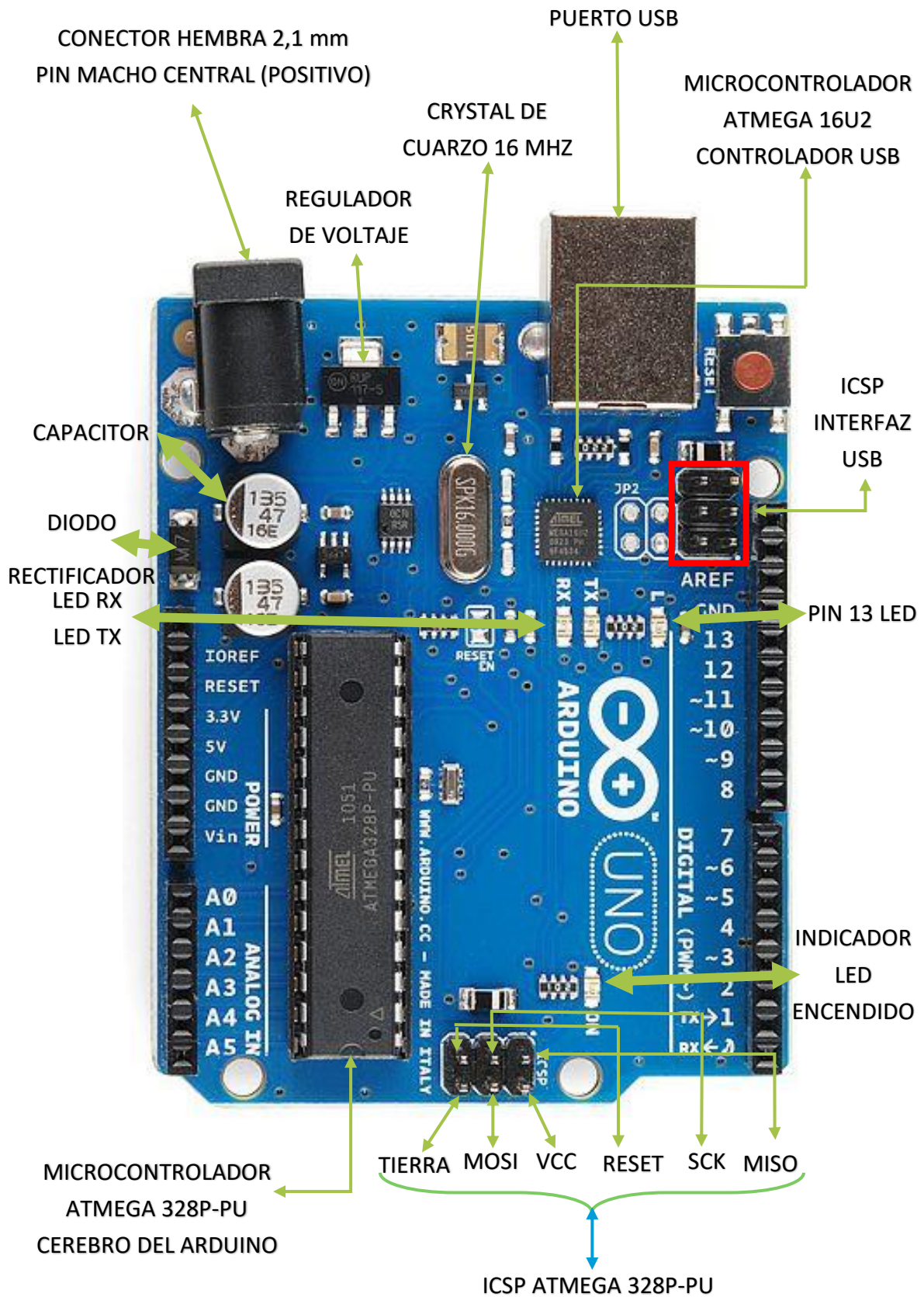


Figura 2.16 Elementos Arduino UNO

Fuente: Israel Cevallos

❖ **Conector USB**

Arduino Uno recibe la alimentación eléctrica mediante su conector USB, también es el medio por el cual se transmite la información del código programado en su propio entorno, mediante la instrucción *Serial.println()*

❖ **Conector de alimentación**

La placa de Arduino puede ser alimentada por su conector hembra con pin macho central, mediante una tensión entre 7 a 12 voltios.

❖ **Pin GND**

La placa de Arduino UNO consta de varios pines GND, que es la abreviatura de “tierra” en inglés; estos pines son usados para conectar a tierra los circuitos externos que fueren acoplados al sistema.

❖ **Pin 5v**

Es el pin encargado de suministrar el voltaje de energía al Arduino UNO

❖ **Pin 3.3 v**

Es el pin encargado de suministrar el voltaje de potencia al Arduino UNO

❖ **Pines de entrada analógicas**

Estos pines son aquellos que leen señales de sensores analógicos, convierte en un valor digital para que podamos dar lectura.

❖ **Pines de entrada y salida (pines digitales)**

Son aquellos pines que van desde el 0 al 13, que pueden ser utilizados tanto para entradas y salidas digitales para una acción y reacción.

❖ **Pines digitales con tilde**

Son aquellos que tienen una tilde (~) al lado de su número, como es el 3,5,6,9,10 y 11; dichos pines pueden ser usados como entradas y salidas normales, o también como modulaciones mediante anchos de pulsos (Pulse Width Modulation/PWM).

❖ **AREF**

Son soportes de referencia analógica, la mayoría de las veces este pin queda vacío, aunque también puede usarse para establecer tensión de referencia externa de 0 a 5 voltios.

❖ **Botón de reinicio**

Este botón mientras se mantiene presionado por unos segundos, se conecta temporalmente a tierra, con el objetivo de borrar el código cargado en la placa de Arduino UNO y poder añadir nueva información según las necesidades.

❖ **Led indicador de alimentación**

Este led estará prendido siempre que la placa Arduino este energizada ya sea, por su conector USB o por el conector de alimentación.

❖ **Led TX**

El Led TX encendido, quiere decir que está en proceso de transmisión de comunicación en serie de datos; es decir, se encenderá cada vez que se cargue un nuevo código al Arduino UNO.

❖ **Led RX**

El Led RX encendido indica que está en proceso de recepción de comunicación en serie de datos, así como cuando ingrese un nuevo código al Arduino UNO.

❖ **Microcontrolador ATMEGA 328P-PU**

Este microcontrolador es aquel que comanda toda la placa de Arduino UNO, que almacena toda información del código que se le cargue.

El ATMEGA 328P-PU es un circuito integrado de alto rendimiento, el cual es programable y de esta forma permite que se realice una tarea antes programada de manera autónoma.

Dicho microcontrolador posee pines de entrada y salida, que tienen como función principal comunicar al exterior la información que este posee. De esta manera en los pines de entrada se conecta los elementos que reciben datos del exterior, mientras que, en los pines de salida se conectará los actuadores, los cuales van a ejecutar la acción requerida que interactúa con el entorno físico externo.

Por otro lado posee 23 pines de entradas y salidas de los cuales 5 pines que constan de tierra, voltaje, soporte de referencia analógica (AREF) tiene una capacidad de 8 bits, una memoria flash de 32 KB la cual permite lectura y escritura simultáneamente, una memoria EEPROM de 1 KB, memoria RAM de 2KB, velocidad de procesamiento de 20 MHZ, posee además 3 temporizadores o contadores de señales, temporizador de vigilancia, tensión mínima de alimentación de 1.8 v, tensión máxima de alimentación de 5.5 v, temperatura de trabajo máxima 85°C.

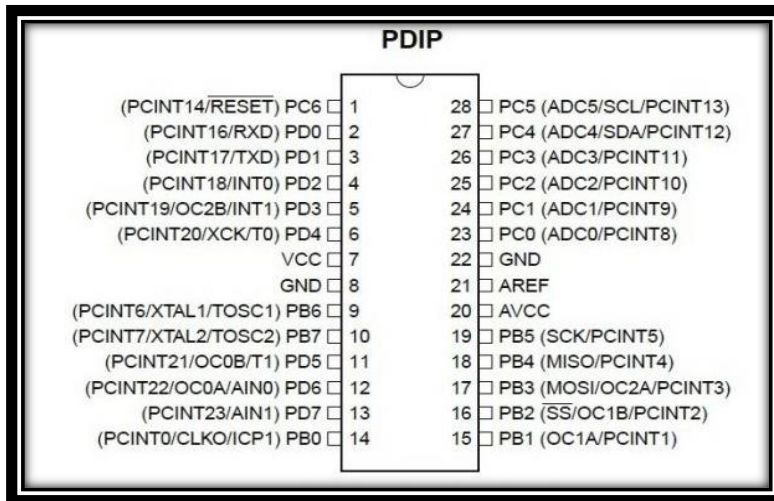


Figura 2.17 Microcontrolador ATMEGA 328P-PU

Fuente: <http://www.electronicoscaldas.com/microcontroladores-atmel/383-microcontrolador-atmega328p-pu.html>

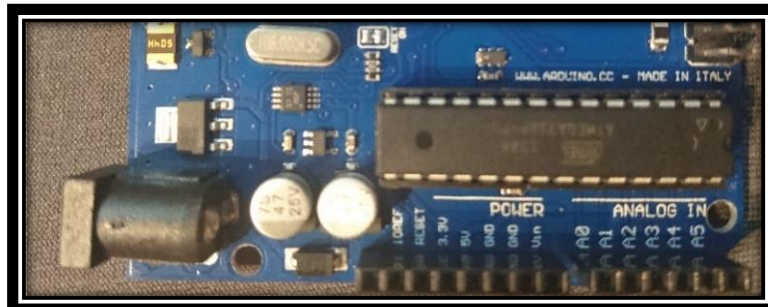


Figura 2.18 Microcontrolador Arduino UNO

Fuente: Israel Cevallos

2.4.8.5. Señales de Arduino.

2.4.8.5.1. Señales digitales

Según Artero (2013), dice que: La “señal digital es aquella que solo tiene un número finito de valores posibles”, es decir, es algo que ya está definido y que tiene acciones establecidas. (Artero, 2013)

2.4.8.5.2. Señales analógicas

Según Artero (2013), dice que: La “señal analógica es aquella que tiene infinitos valores posibles dentro de un rango determinado” (Artero, 2013)

2.4.8.6. Características internas Arduino UNO.

Las características internas de Arduino son las siguientes:

❖ Bits en un microcontrolador

El número de bits es el tamaño de las variables que el microcontrolador puede manejar, es decir, que puede realizar operaciones lógicas y aritméticas con operandos que son los datos, mediante las instrucciones que se ingresaron.

❖ Memoria flash en un microcontrolador

Son los campos en los cuales se puede reprogramar, maneja una velocidad alta de programación y permite varios ciclos de escritura y borrado, una de las mayores ventajas de esta memoria es que puede recibir datos sin necesidad de desmontarla de la placa de Arduino.

❖ Memoria EEPROM en un microcontrolador

Son memorias programables, de lectura únicamente y borrables eléctricamente. EEPROM por sus siglas en inglés significa (Electrical Erasable Programmable Read Only Memory), por otro lado, las acciones de programado y borrado se realizan mediante una computadora, sin embargo, se debe tomar en cuenta que las veces que se puede realizar dichas acciones es finita, por lo tanto, no debe hacerse una reprogramación continua.

❖ Memoria RAM en un microcontrolador

La memoria RAM por otro lado, se encarga principalmente de almacenar los datos necesarios de las variables y cambios de información, motivo por el cual contiene poca capacidad ya que solo almacena la información temporal.

❖ **Velocidad de procesamiento o frecuencia de reloj en un microcontrolador**

Los microcontroladores requieren de un oscilador que genera pulsos de reloj, permite el trabajo en ciclos que dependen de la información introducida.

La velocidad máxima de procesamiento en un microcontrolador es de 20 MHz.

❖ **Contadores de señal en un microcontrolador**

Es necesario que un microcontrolador posea un contador de señal, con la finalidad de generar eventos cada cierto tiempo, que registra acciones precisas.

❖ **Regulador de voltaje**

Controla la cantidad de tensión que llega a la placa de Arduino UNO, tiene como límite de entrada 20 voltios.

❖ **Pin 13 Led**

Es usado como dispositivo de salida, se usa para la depuración.

❖ **Led de encendido**

Indica que la placa Arduino UNO recibe la alimentación, y es útil para la depuración.

2.4.9. NFC/RFID.

2.4.9.1. Introducción.

NFC significa Near Field Communication (Comunicación de campo cerca), y RFID Radio Frequency identification (Identificación de radio frecuencia); son

sistemas basados en la comunicación inalámbrica, lo cual deja atrás a la tecnología del código de barras e incluso a los códigos QR (Quick Response Code).



Figura 2.19 Tecnología inalámbrica

Fuente: <http://computerhoy.com/noticias/internet/tecnologia-thz-comunicaciones-inalambricas-alta-velocidad-40303>

2.4.9.2. Diferenciación.

“RFID es el proceso en el cual objetos son identificados de manera única que usan ondas de radio y NFC es una tecnología especializada dentro de la familia de tecnologías RFID.

Específicamente, NFC es una rama de la tecnología de alta frecuencia (HF) de RFID y ambos operan a una frecuencia de 13.56MHz. NFC fue diseñado para otorgar un intercambio seguro de datos y es capaz de funcionar tanto como un lector NFC como también una etiqueta NFC. Esta característica única permite a los dispositivos NFC comunicarse punto a punto.” (MCI Capacitación, 2015)

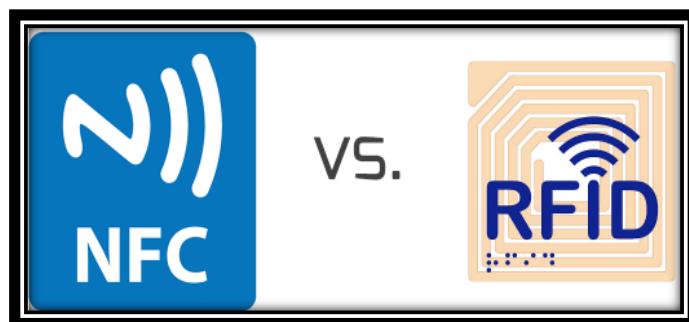


Figura 2.20 Tecnología inalámbrica

Fuente: <http://www.taptrack.com/the-difference-between-nfc-and-rfid/>

2.4.10. NFC.

2.4.10.1. Definición.

El NFC funciona con tecnología inalámbrica con una tasa de transferencia que puede alcanzar los 424 kbit/s por lo que su enfoque más que para la transmisión de grandes cantidades de datos es para comunicación instantánea, es decir, identificación y validación de equipos/personas.



Figura 2.21 Tecnología inalámbrica

Fuente: <http://www.telesa.net/actualidad/25/nfc-qu-es-eso-y-para-qu-sirve.html>

2.4.10.2. Áreas de contribución desarrolladas con NFC.



Figura 2.22 Áreas de desarrollo con NFC
Fuente: Israel Cevallos

2.4.11. RFID.

2.4.11.1. Definición.

Los sistemas RFID son conocidos como tecnología de identificación automática, es decir se compara con la lectura de código de barras.

El fabricante de los módulos RFID es *NXP Semiconductors*, menciona que es un lector y escritor a la misma vez, funciona con una frecuencia de 13,56 MHz, este sistema está basado en la norma ISO/IEC 14443 la que posibilita la comunicación con tarjetas del mismo estándar.



Figura 2.23 Tecnología RFID

Fuente: <http://www.mindfulstreams.com/ideas-for-rfid-technology/>

2.4.11.2. Funcionamiento y modo de trabajo.

“Los años 50 fueron una era de exploración de las técnicas RFID siguiendo la evolución técnica en radio y radar de las décadas de los años 30 y 40”.

(Bejarano, 2014, pág. 20)

Los sistemas RFID poseen dos componentes un emisor y un receptor, que está localizado dentro de un lector RFID y de una tarjeta o tag.

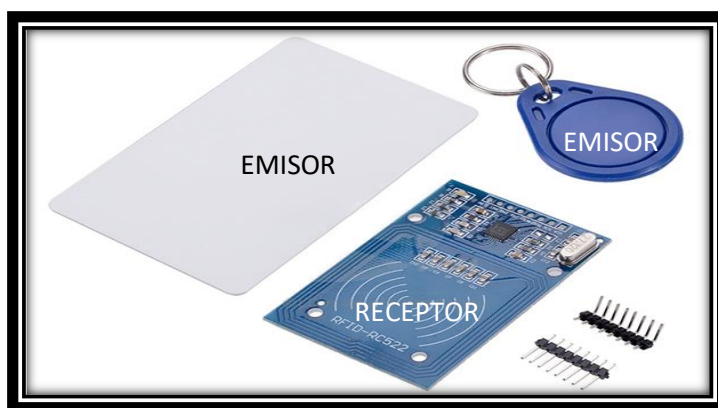


Figura 2.24 Lector, tarjeta y tag RFID

Fuente: Israel Cevallos

2.4.11.3. Reseña histórica del RFID.

Tabla 2.4 Reseña histórica del RFID

TIEMPO	EVENTO
1940-1950	RFID inventado en 1948
1950-1960	Primeras exploraciones de la tecnología RFID
1960-1970	Desarrollo de la teoría RFID.
1970-1980	Explosión del desarrollo del RFID. Adaptación temprana de las implementaciones del RFID
1980-1990	Aplicaciones comerciales del RFID entran al mercado
1990-2000	El RFID se convierte en una parte de la vida diaria
2000-2010	El RFID continua su crecimiento.
2010-	Se postula para ser una de las tecnologías más utilizadas en el futuro

Fuente: Israel Cevallos

2.4.11.4. Norma ISO/IEC 14443.



Figura 2.25 Norma ISO/IEC

Fuente: <http://www.open-std.org/JTC1/SC22/WG9/>

La norma ISO/IEC 14443 trata acerca de las tarjetas de proximidad, los posibles métodos de detección, comunicación y el tiempo empleado para inicializar el proceso.

Define cuáles son los protocolos y comandos de transmisión utilizados por las tarjetas RFID con chip sin contacto físico.

“Es importante destacar sus distintas partes:

- **ISO 14443-1:2008 Part 1:** especifica las características físicas
- **ISO 14443-2:2010 Part 2:** especifica la potencia RF y la interfaz de señal
- **ISO 14443-3:2011 Part 3:** especifica las funciones de inicialización y anticolisión entre chips
- **ISO 14443-4:2008 Part 4:** especifica el protocolo de transmisión” (FQ Ingeniería Electrónica, s.f.)

2.4.11.5. Elementos del RFID.

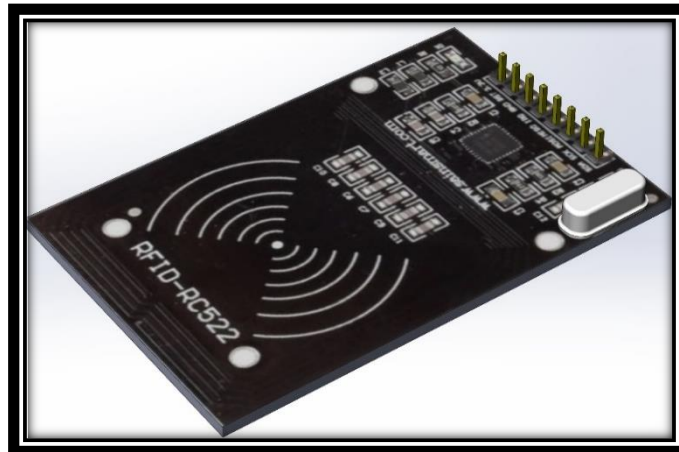


Figura 2.26 Lector RFID diseñado en SolidWorks
Fuente: Israel Cevallos

2.4.11.5.1. Integrado MFRC522

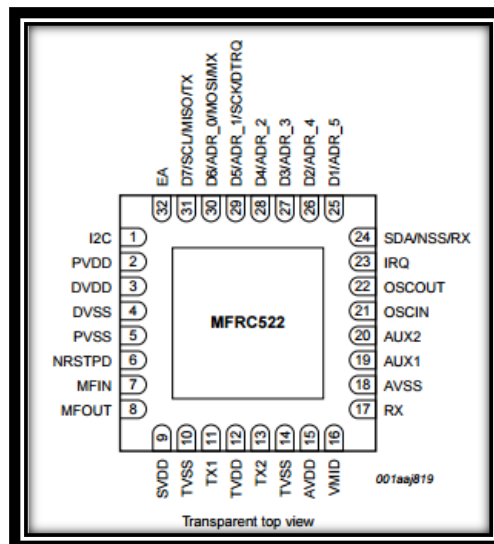


Figura 2.27 Integrado MFRC522

Fuente: http://www.nxp.com/documents/data_sheet/MFRC522.pdf

➤ **Tabla de comunicación Serial de la Interfaz Periférica (SPI) con el microcontrolador.**

Tabla 2.5 Comunicación SPI con el microcontrolador.

PIN	SÍMBOLO	DESCRIPCIÓN
6	NRSTPD	Resetear
24	SDA	Comunicación Serial bus I2C
29	SCK	Reloj Serial SPI
30	MOSI	Master de salida, esclavo de entrada
31	MISO	Master de entrada, esclavo de salida

Fuente: Modificado por Israel Cevallos.

Obtenido de: [/http://www.nxp.com/documents/data_sheet/MFRC522.pdf](http://www.nxp.com/documents/data_sheet/MFRC522.pdf)

La comunicación Serial Bus I2C, se utiliza para denominar la comunicación interna entre circuitos y el accionamiento en el cual se inicializa con un maestro, mientras el esclavo reacciona.

2.4.11.5.2. Tarjetas y Tags RFID.

Las tarjetas y Tags constan con un chip y una antena, el chip se encarga de almacenar en su memoria los datos propios o identificación única.

Hay dos tipos de tarjetas, las pasivas y las activas; en este caso se utiliza de tipo pasivo, ya que estas no necesitan alimentación externa para su funcionamiento, sino que con solo acercarse al lector induce el campo magnético lo que permitirá la lectura de la identificación de cada tarjeta o tag.

Para que la tarjeta responda a una acción debe estar cerca del lector RFID con la finalidad de que esta se energice y pueda concatenarse la información.

La distancia de funcionamiento de la tarjeta con el lector, es de aproximadamente 1 cm y esto depende de las obstrucciones físicas que impidan una buena comunicación.

2.4.12. Bluetooth

El Bluetooth es uno de los elementos que permite conectarse al Chevrolet Corsa Evolution, para permitir la apertura y cierre de puertas, así también el contacto del mismo para el próximo paso el encendido mediante el botón SPARCO.

2.4.12.1. Estados del módulo.

❖ Estado desconectado

Se refiere a que el Bluetooth es alimentado y no se ha establecido una conexión con algún dispositivo y el LED parpadea continuamente.

❖ Estado conectado

El Bluetooth realiza una conexión con otro dispositivo y el LED parpadea dos veces.

❖ Modo AT1

En este estado podemos enviarle comando al Bluetooth con la misma velocidad con la que está configurado; para este proceso es necesario presionar el botón que viene incorporado en el módulo.

❖ Modo AT2

Para este modo es necesario tener presionado el botón incorporado mientras se realiza la alimentación del módulo y también se puede enviar comandos a la velocidad de 38400 baudios, y permite saber la velocidad a la que se trabajó anteriormente.

2.4.12.2. Diagrama módulo Bluetooth HC-05.

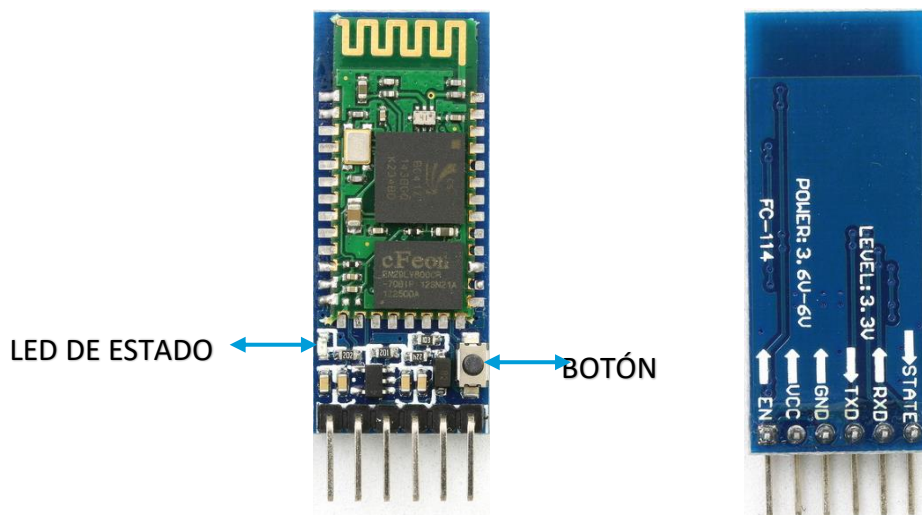


Figura 2.28 Diagrama módulo Bluetooth HC-05

Fuente: Israel Cevallos

2.4.13. Botón de encendido SPARCO

El botón de encendido es un pulsador que va a permitir encender el vehículo, posterior al accionamiento del contacto mediante NFC/RFID o a su vez por medio de Bluetooth con la aplicación CORSA_C.



Figura 2.29 Logotipo SPARCO

Fuente: <http://www.sparcousa.com/>



Figura 2.30 Botón de encendido SPARCO
Fuente: Israel Cevallos

2.4.14. Módulo relé de 2 canales

Un relé permite activar o desactivar el mecanismo de un elemento, es decir energizar o no un actuador de puertas.

Se controla mediante la entrada de datos, funciona con 5 voltios y son capaces de llevar cargas de hasta 10 Amperios.

La corriente que consume las bobinas de los relés debe ser provista por medio del Arduino UNO, por lo tanto, cada bobina consume aproximadamente 90mA, es así que las dos bobinas juntas, consumen un total aproximado de 180mA; este valor es inferior al valor que puede suministrar un puerto USB que son de 500mA.

La alimentación externa que posea el Arduino UNO, hará que varíe la corriente y aumente en un porcentaje de aproximadamente el 100%, y se obtiene como resultado una corriente de 360mA.



Figura 2.31 Módulo relé de 2 canales
Fuente: Israel Cevallos

2.5. Componentes eléctricos del vehículo.

Los componentes eléctricos principales para el funcionamiento del vehículo son:

2.5.1. Batería:

La principal función de la batería es la de entregar la energía necesaria para que el vehículo pueda arrancar, y esta a su vez está conformada de agua y ácido sulfúrico que interactúan entre sí para formar el electrolito que va a permitir la conducción de la electricidad.

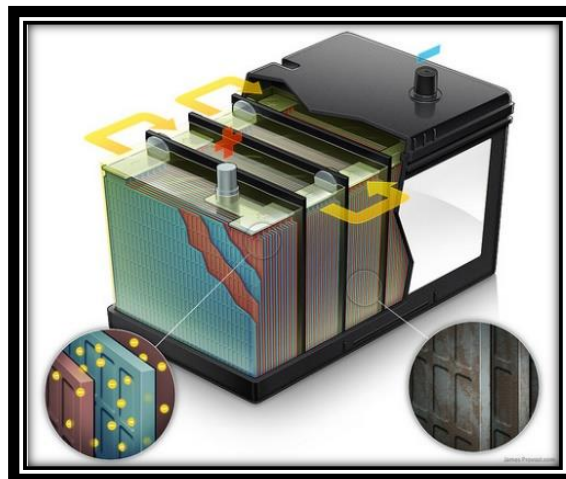


Figura 2.32 Proceso interno del electrolito en la batería
Fuente: http://www.autobodymagazine.com.mx/abm_previo/2012/06/la-bateria-del-automovil/

Para que se genere electricidad se necesita que el electrolito haga un recorrido por las paredes y rejillas de la batería y libere los electrones, en consecuencia, los electrones se unen con una placa positiva lo cual produce 2 voltios de electricidad y continúan el paso por las demás placas y acumula 2 voltios sucesivamente.

Finalmente, al instante que los electrones ya han recorrido el total de las placas, se ha obtenido como resultado 12 voltios.

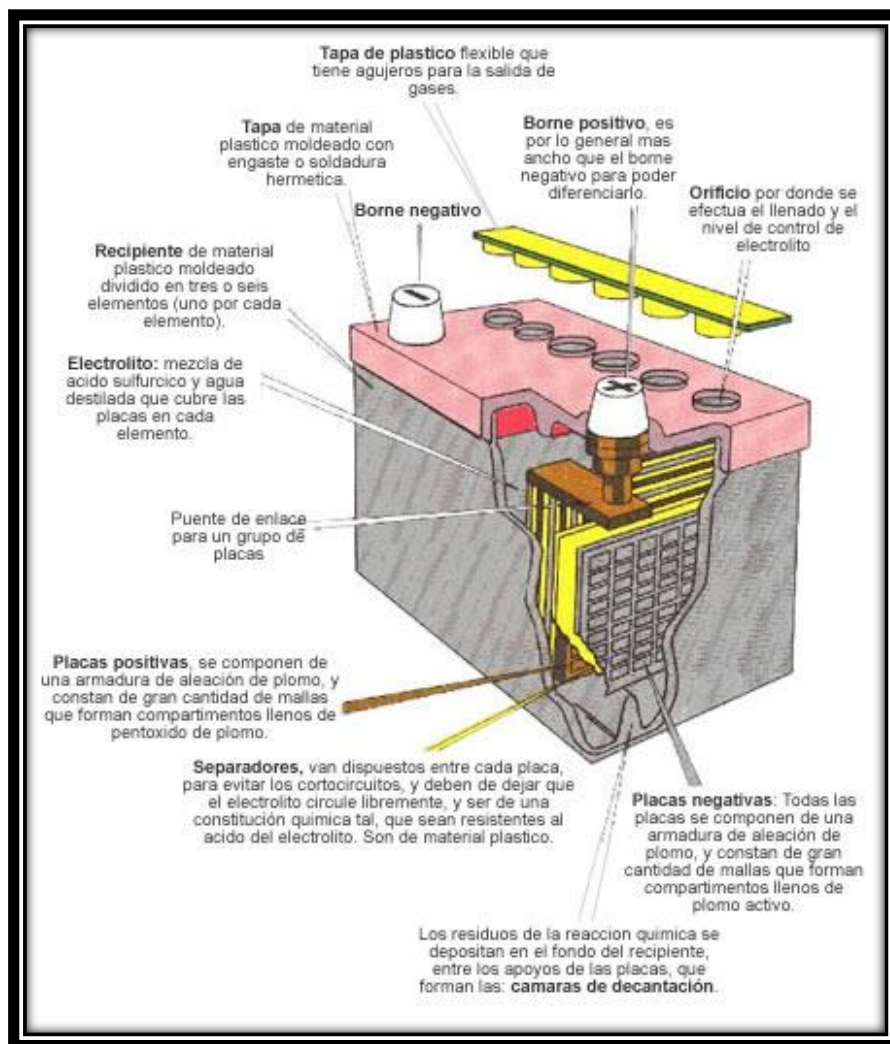


Figura 2.33 Descripción de la batería del vehículo

Fuente: <http://www.areatecnologia.com/baterias-y-acumuladores.htm>

2.5.2. Relé:

El relé es un elemento que funciona mediante el electromagnetismo, lo que hace que al momento que existe una corriente eléctrica débil, abre o cierra un circuito comportándose como un interruptor o gobernador en un circuito eléctrico.

2.5.3. Alternador:

Según Dani (2014), “El alternador es el encargado de proporcionar la energía eléctrica necesaria a los consumidores del automóvil (encendido, luces, motores de limpia-parabrisas, cierre centralizado, etc.), también sirve para cargar la batería. (Dani, 2014)

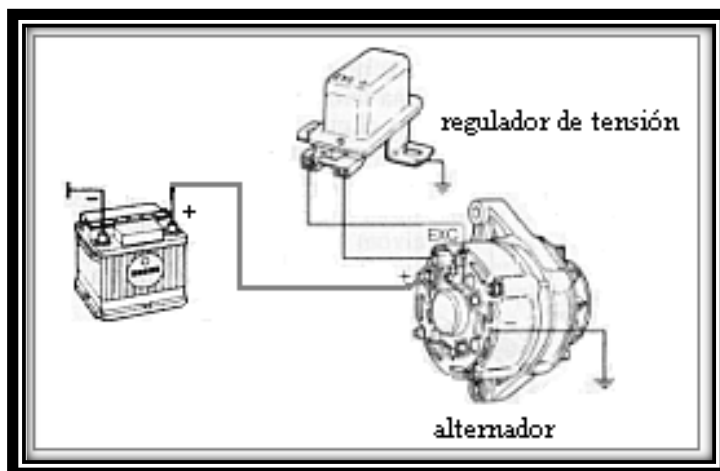


Figura 2.34 Conexión del alternador

Fuente: http://www.aficionadosalamecnica.net/curso_alternador.htm

2.5.4. Regulador de voltaje:

El regulador de voltaje permite mantener el voltaje constante en el vehículo.

2.5.5. Acumulador:

Es el nombre propio de la fuente de voltaje o conocida también como batería.

2.5.6. Arranque:

Es un elemento que proporciona un mínimo número de revoluciones al motor de combustión interna para que este funcione por sí solo, luego de dar aquel impulso inicial.

De esta manera, el funcionamiento y encendido del vehículo será óptimo.

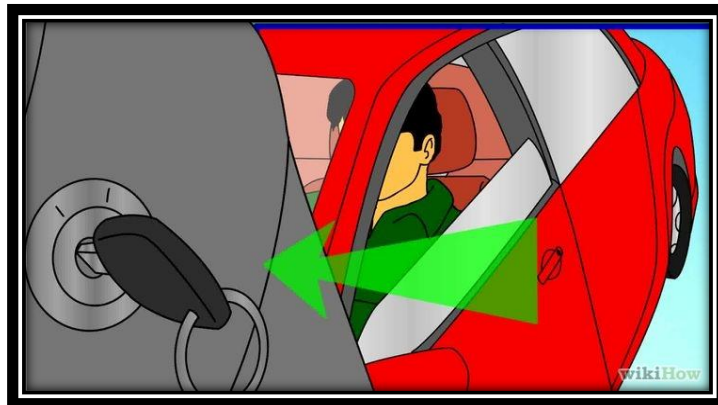


Figura 2.35 Funcionamiento y arranque del vehículo

Fuente: <http://es.wikihow.com/encender-un-carro-manual>

CAPÍTULO 3

3. Funcionamiento del sistema de Alarma Chevrolet.

3.1. Alarma Chevrolet.

La alarma de Chevrolet posee un sistema de control remoto, que permite el armado y desarmado de los seguros de las puertas del vehículo, así también una clave que permite el arranque del vehículo.

3.1.1. Control remoto con clave

Es un sistema de control mediante Rolling Code como se puede observar en la Figura 3.1, que mediante su código de seguridad mediante 3 números y 5 dígitos permite activar los sistemas del vehículo para posteriormente el arranque del mismo.



Figura 3.1 Control de alarma Chevrolet

Fuente: [https://ve.michevystar.com/ChevystarCliente/css/pdf/Manual %20Alarma.pdf](https://ve.michevystar.com/ChevystarCliente/css/pdf/Manual%20Alarma.pdf)

3.1.2. Módulo de alarma

La principal función de este módulo es la de captar las señales que envía el control remoto, así también como la verificación del código emitido por el mismo, para permitir la apertura y cierre de puertas, y también el arranque del vehículo.

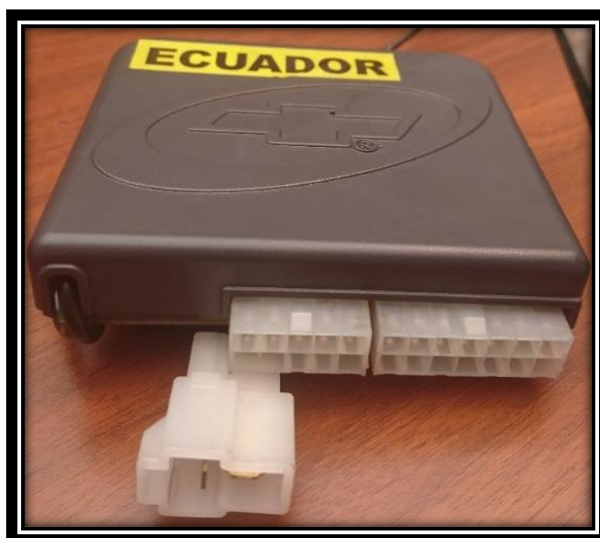


Figura 3.2 Módulo de Alarma Chevrolet
Fuente: Israel Cevallos

3.1.3. Actuadores de puertas.

El actuador de puertas como se muestra en la Figura 3.3, es un elemento que posee un motor eléctrico con un movimiento rotativo que permite mediante engranajes internos, como se puede observar en la Figura 3.4, una oscilación a través de un acople lineal, el cual transmite y transforma el movimiento rotativo en lineal. De esta manera los seguros de la puerta del vehículo se pueden elevar o bajar.



Figura 3.3 Actuador de puertas del vehículo
Fuente: <http://pinoshop.es.tl/Como-Instalar-Seguros-Electricos-Sin-Alarma-En-El-Auto.htm>

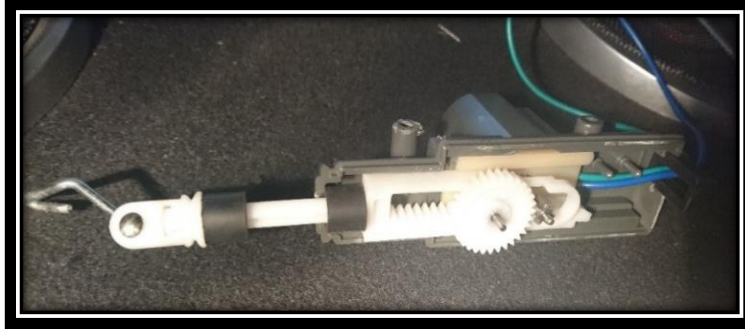


Figura 3.4 Parte interna del actuador de puertas
Fuente: Israel Cevallos

3.1.4. Bloqueo del volante.

Existen dos tipos de bloqueo del volante, mecánico y eléctrico que a continuación se describen cada uno.

3.1.4.1. Bloqueo mecánico/manual

El bloqueo del volante funciona de manera mecánica y manual, ya que es un mecanismo en el cual la llave es la principal herramienta para desactivar este componente, ya que posee un pin y un resorte que cuando la llave no está dentro del switch de encendido y el volante gira levemente, este se traba y no permite su giro; es por esto que la llave al girar dentro del switch hará que el resorte salte y con esto el volante girará libremente.

3.1.4.2. Bloqueo eléctrico

Actualmente en los vehículos que ya no disponen de una cerradura convencional por llave, en el volante viene incorporado un bloqueo eléctrico de columna de dirección (ELV).

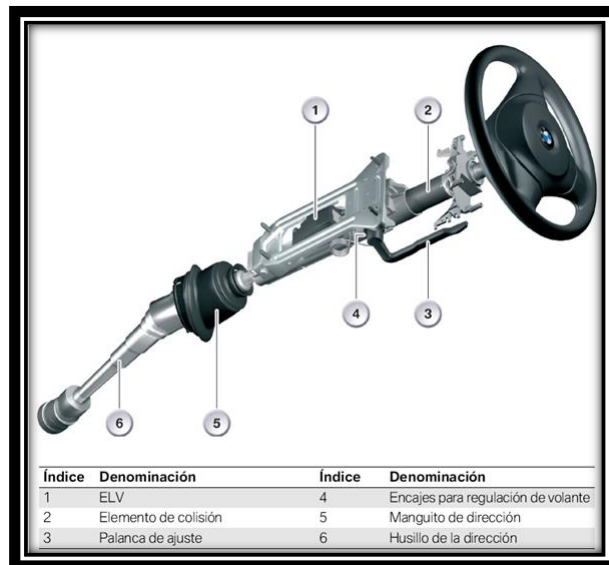


Figura 3.5 Bloqueo eléctrico BMW

Fuente: <http://www.blogmecanicos.com/2015/12/fallo-en-el-bloqueo-electrico-de-la.html>

Según Cazorla (2015), “La columna de dirección forma un conjunto integrado con el ELV, no se pueden cambiar por separado y en caso de disparo del airbag es necesario sustituir el conjunto según versiones.

La función ELV es gestionada por la unidad CAS (Car Access System), ésta es quien ordena el bloqueo o desbloqueo al enclavamiento eléctrico de la columna ELV. Lo efectúa controlando su alimentación y una línea codificada de comunicación.” (Cazorla, 2015)

3.1.4.2.1. Componentes del sistema del bloqueo del volante eléctrico.

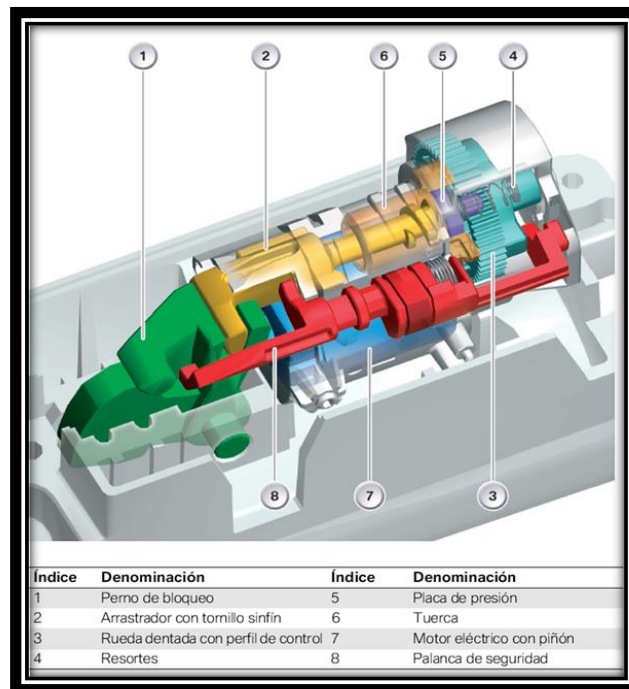


Figura 3.6 Componentes del bloqueo eléctrico del volante BMW

Fuente: <http://www.blogmecanicos.com/2015/12/fallo-en-el-bloqueo-electrico-de-la.html>

3.1.5. Conexiones y funcionamiento de la alarma Chevrolet.

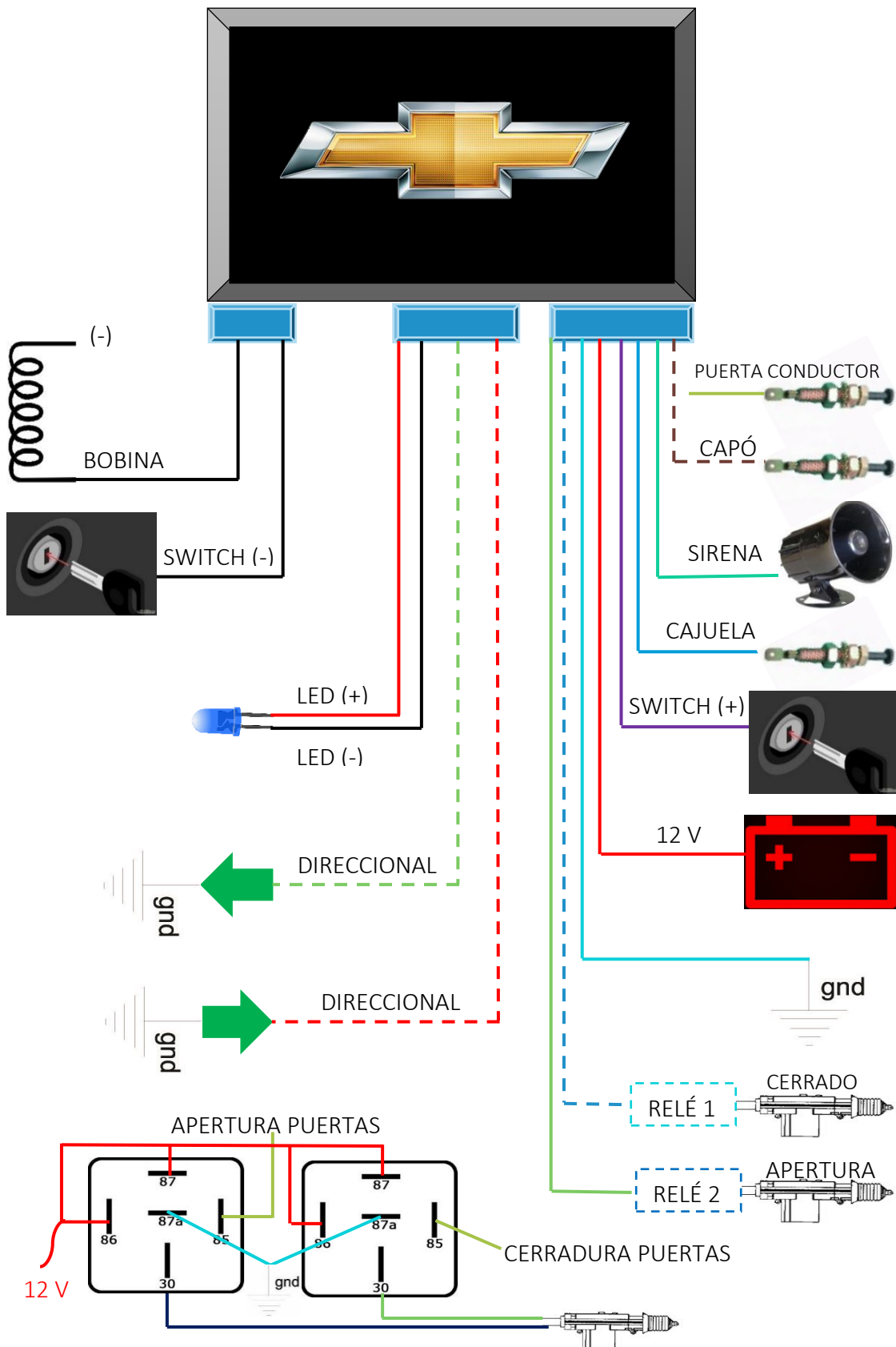


Figura 3.7 Conexiones Alarma Chevrolet
Fuente: Israel Cevallos

CAPÍTULO 4

4. Implementación del circuito electrónico controlado con Arduino UNO, NFC/RFID y Bluetooth.

4.1. Introducción

Mediante Arduino UNO, y sus diferentes componentes como son RFID, Bluetooth y el uso de un dispositivo móvil que cuenta con tecnología NFC, es posible el desarrollo de la implementación para la apertura y cierre de puertas, así también como el contacto del vehículo, por medio de estos componentes antes mencionados y que se detallan a continuación.

Por otro lado, el voltaje de trabajo oscila entre los 7 y 12 voltios lo que permite su uso sin ningún problema con la batería del vehículo al igual que el amperaje de trabajo en el Chevrolet Corsa Evolution.

4.2. Conexión de actuadores.

Los motores de cada una de las puertas accionan la apertura y cierre de los seguros, estos son manejados por actuadores de 2 posiciones, dichos actuadores deben ser conectados a un relé de 5 pines, de los cuales los pines 85 y 86 corresponden a la bobina del relé, el pin 30 es el común y el pin 87a es normalmente abierto y el 87 es normalmente cerrado. Con estos pines podremos energizar con un voltaje de corriente continua de 12 V y 40 A.



Figura 4.1 Actuator de cajuela
Fuente: Israel Cevallos

4.3. Programación en Arduino.

4.3.1. Introducción al entorno de programación.

La Figura 5.12 muestra la iniciación y carga de datos del programa Arduino UNO.



Figura 4.2 Inicio del programa Arduino
Fuente: Israel Cevallos

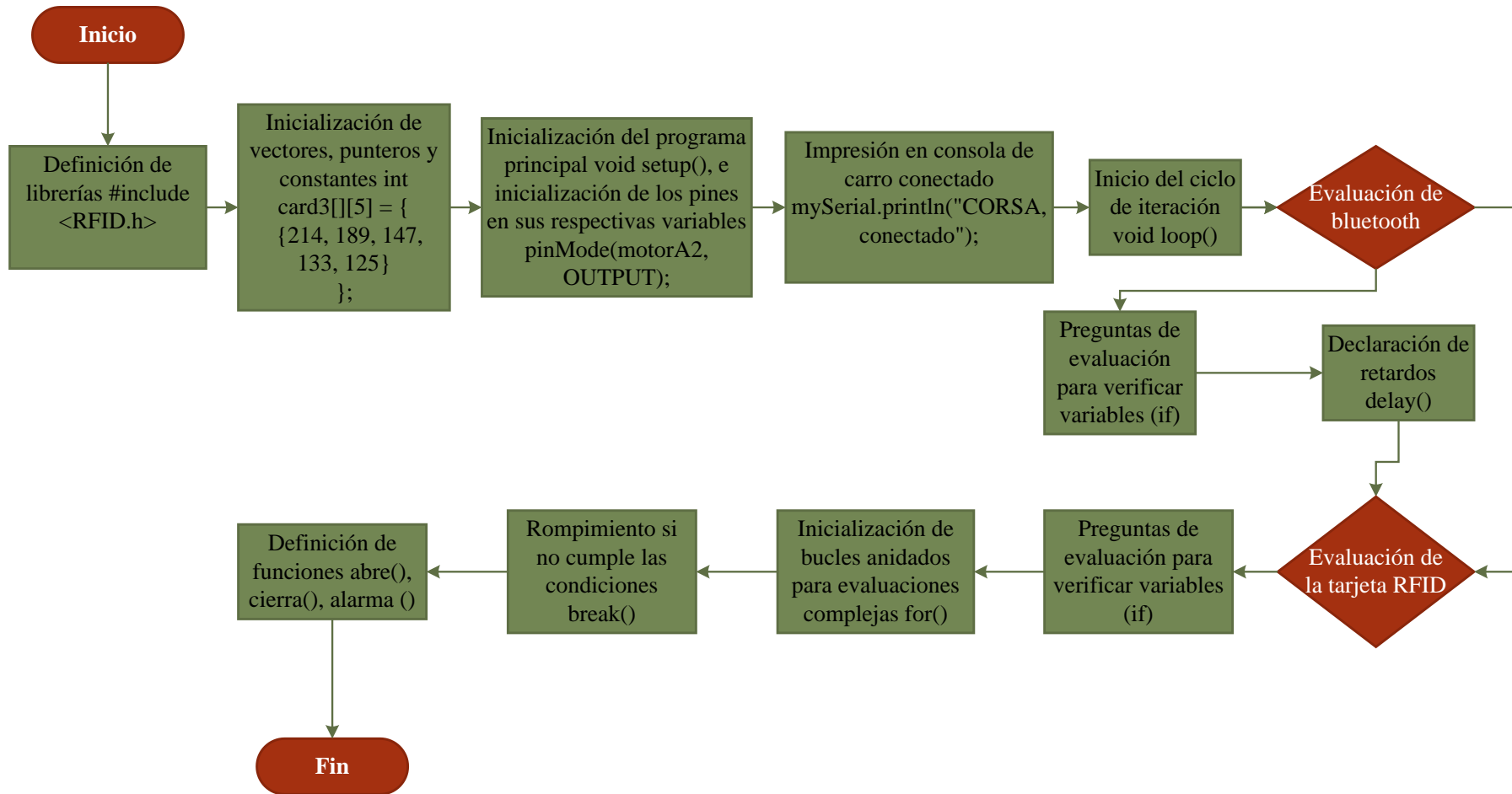


Figura 4.3 Diagrama de flujo de la programación en Arduino

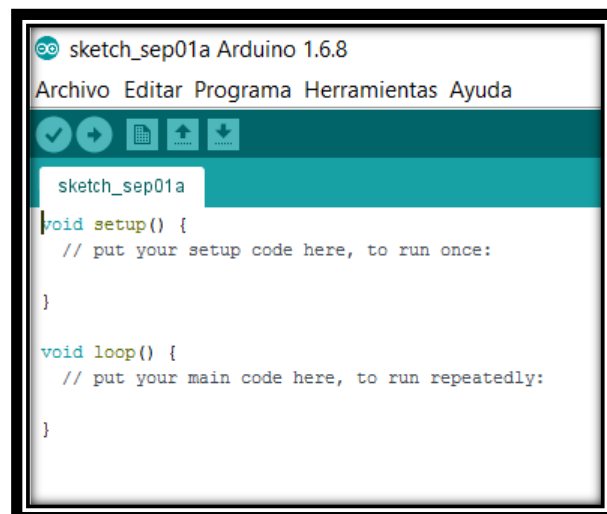
Fuente: Israel Cevallos

- **Barra de herramientas**

- ✔ **Verificar:** Revisa el código en busca de errores
- 📄 **Nuevo:** Crea un nuevo entorno de programación
- ➡ **Subir:** Carga el código a la placa de Arduino
- 📁 **Abrir:** Despliega un menú con todos los códigos de programación guardados.
- 🔊 **Monitor Serie:** Permite mostrar datos enviados por la placa Arduino y su comunicación directa.
- ➡ **Salvar:** Guarda la programación

4.3.2. Programación de Arduino UNO y componentes RFID y Bluetooth.

La programación en Arduino debe contener dos partes en su estructura que son funciones en donde se encuentra declaraciones o instrucciones que debe seguir la placa de Arduino.



```
sketch_sep01a Arduino 1.6.8
Archivo Editar Programa Herramientas Ayuda
sketch_sep01a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Figura 4.4 Tomado del código de programación de inicio por defecto
Fuente: Israel Cevallos

Estas dos funciones especificadas que son *void setup()* y *void loop()* son requeridas para que el programa funcione.

- **Setup ():** Esta función es utilizada una sola vez al inicio del programa, es decir inicializa el puerto serie y se puede establecer el estado inicial de las salidas de la placa.

```
void setup() {
  state = '0';

  Serial.begin(9600);
  SPI.begin();
  rfid.init();
}
```

Figura 4.5 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

- **Loop ():** Su principal función es la ejecución cíclica, es decir que el programa siempre envía respuestas continuamente ante las interacciones programadas y que recibe la placa de Arduino.

```
void loop() {

  if (mySerial.available() > 0) {
    state = mySerial.read();
    Serial.write(state);
  }

  if (Serial.available() > 0) {
    mySerial.write(state);
  }

  if (state == 'f' && estado == 0) {
    abre();
    estado = 1;
    detector = 0;
  }
  else if (state == 'i' && estado == 1) {
    cierra();
    estado = 0;
    detector = 0;
  }
  else if (state == 'a') {
    digitalWrite(salB, LOW);
    digitalWrite(salC, HIGH);
    detector = 0;
    delay(200);
    digitalWrite(salB, HIGH);
    digitalWrite(salC, HIGH);
  }
}
```

Figura 4.6 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

- **Entre llaves {}** : Las llaves se utilizan para delimitar el inicio y final de un bloque de instrucciones o declaraciones.

```
void setup() {
  state = '0';

  Serial.begin(9600);
  SPI.begin();
  rfid.init();

  // Set pins as outputs:
  //pinMode(motorA1, OUTPUT);
  pinMode(motorA2, OUTPUT);
  pinMode(motorB1, OUTPUT);
  pinMode(motorB2, OUTPUT);
  pinMode(salA, OUTPUT);
  pinMode(salB, OUTPUT);
  pinMode(salC, OUTPUT);

  pinMode(bt, OUTPUT);
  // Initialize serial communication at 9600 bits per second:

  mySerial.begin(9600);
  mySerial.println("ISRA, conectado");

  digitalWrite(salA, HIGH);
  digitalWrite(salB, HIGH);
  digitalWrite(salC, HIGH);
  digitalWrite(bt, HIGH);
  //cierra();
}
```

Figura 4.7 Tomado del código de programación de las puertas

Fuente: Israel Cevallos

- **Punto y coma “;”**: Se debe poner un punto y coma luego de finalizar una línea de código con la finalidad de que no se produzca un error de compilación, y también se usa para separar elementos en una declaración.

```
digitalWrite(salA, HIGH);
digitalWrite(salB, HIGH);
digitalWrite(salC, HIGH);
digitalWrite(bt, HIGH);
```

Figura 4.8 Tomado del código de programación de las puertas

Fuente: Israel Cevallos

- **Línea de comentarios “//”:** Son áreas que no están destinadas a ser tomadas en cuenta por el código, lo que permite es añadir la descripción del código y poder comprender el programa sin ocupar espacio en la memoria.

```
const int salA = A0; // PARA ACTIVAR ALARMA
const int salB = A4; // PARA ACTIVAR ALARMA
const int salC = A5; // PARA ACTIVAR ALARMA

int state; //datos por bt
bool access = false; //activa al validar RFID
int detector = 0;
```

Figura 4.9 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

4.3.2.1. Variables.

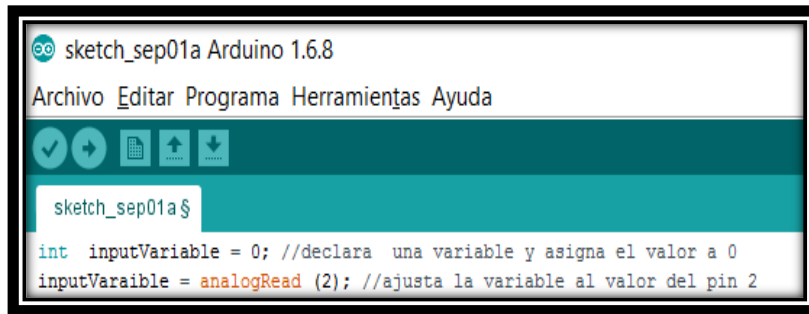
Una variable es aquella que debe ser declarada y a la vez debe contener un valor alfanumérico con la finalidad de que el programa pueda tomar ese valor posteriormente para realizar una acción. El valor de una variable puede cambiar a lo largo del programa.

```
}

int estado = 0;
int etapa = 0;

void loop() {
```

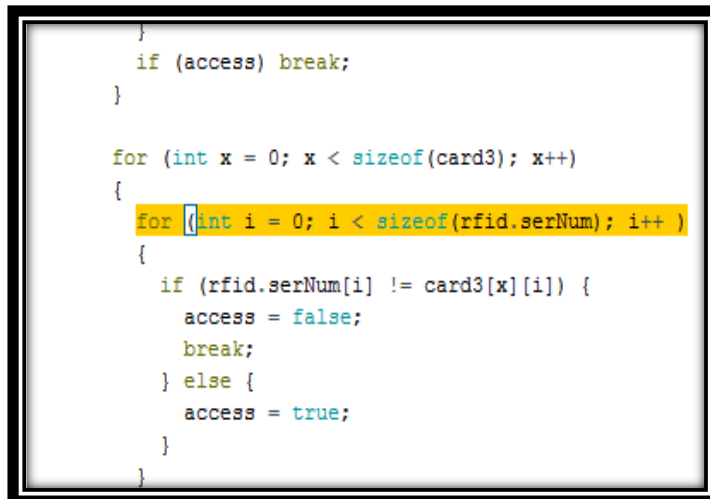
Figura 4.10 Tomado del código de programación de las puertas
Fuente: Israel Cevallos



```
sketch_sep01a Arduino 1.6.8
Archivo Editar Programa Herramientas Ayuda
sketch_sep01a $
int inputVariable = 0; //declara una variable y asigna el valor a 0
inputVariable = analogRead (2); //ajusta la variable al valor del pin 2
```

Figura 4.11 Código de programación para declarar variables
Fuente: Israel Cevallos

En el momento que una variable este asignada debe ser verificada, y determinar si cumple con las condiciones propuestas.



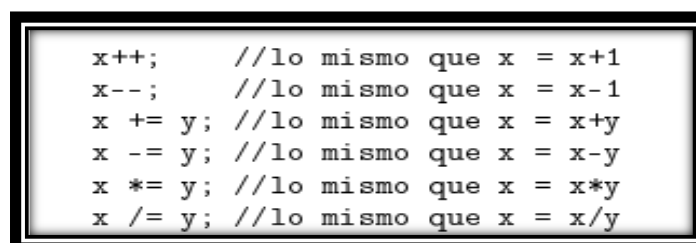
```
}
if (access) break;
}

for (int x = 0; x < sizeof(card3); x++)
{
for (int i = 0; i < sizeof(rfid.serNum); i++)
{
if (rfid.serNum[i] != card3[x][i]) {
access = false;
break;
} else {
access = true;
}
}
}
```

Figura 4.12 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

4.3.2.2. Asignaciones compuestas.

Se combina una operación aritmética con una variable asignada, se utiliza una sentencia *for* en donde incluye dicha asignación compuesta.



```
x++; //lo mismo que x = x+1
x--; //lo mismo que x = x-1
x += y; //lo mismo que x = x+y
x -= y; //lo mismo que x = x-y
x *= y; //lo mismo que x = x*y
x /= y; //lo mismo que x = x/y
```

Figura 4.13 Tomado del código de programación de las puertas
Fuente:http://www.uco.es/aulasoftwarelibre/wpcontent/uploads/2010/05/Arduino_user_manual_es.pdf

```
for (int x = 0; x < sizeof(card3); x++)
{
    for (int i = 0; i < sizeof(rfid.serNum); i++ )
    {
```

Figura 4.14 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

4.3.2.3. Operadores de comparación.

Estas comparaciones se hacen entre una variable o constante, para determinar si una condición en específico es verdadera, se usa sentencia *if*.

```
x == y; //x es igual a y
x != y; //x no es igual a y
x < y; //x es menor que y
x > y; //x es mayor que y
x <= y; //x es menor o igual que y
x >= y; //x es mayor o igual que y
```

Figura 4.15 Tomado del código de programación de las puertas
Fuente:http://www.uco.es/aulasoftwarelibre/wpcontent/uploads/2010/05/Arduino_user_manual_es.pdf

```
if (access && estado == 0) {
    Serial.println("CARRO ABIERTO");
    abre();
    estado = 1;
    access = 0;
    detector = 0;
} else if (access && estado == 1) {
    Serial.println("CARRO cerrado");
    cierra();
    estado = 0;
    access = 0;
    detector = 0;
} else {
    detector = 1;
    Serial.println("INTRUSO");
}
```

Figura 4.16 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

4.3.2.4. Constantes de programación.

Las constantes son valores que usa Arduino y no cambian a lo largo del programa, y permite que el código sea más legible.

- **True/False**

Son constantes Booleanas, eso quiere decir que utilizan niveles lógicos, es decir FALSE se lo define como cero (0), mientras que TRUE se lo define con uno (1).

```
for (int x = 0; x < sizeof(card2); x++)
{
  for (int i = 0; i < sizeof(rfid.serNum); i++ )
  {
    if (rfid.serNum[i] != card2[x][i]) {
      access = false;
      break;
    } else {
      access = true;
    }
  }
  if (access) break;
}
```

Figura 4.17 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

- **High/Low**

High se lo define como uno lógico (1), ON, o 5 voltios; mientras que el Low se lo define como cero lógico (0), OFF o 0 voltios.

```
void abre() {
  digitalWrite(sala, LOW);
  delay(15);
  digitalWrite(sala, HIGH);
  delay(15);
  //digitalWrite(motorA1, HIGH);
  digitalWrite(motorA2, LOW);
  digitalWrite(motorB1, HIGH);
  digitalWrite(motorB2, LOW);
  delay(500);
  Serial.println("ABIERTO");
  //digitalWrite(motorA1, LOW);
  digitalWrite(motorA2, LOW);
  digitalWrite(motorB1, LOW);
  digitalWrite(motorB2, LOW);
}
```

Figura 4.18 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

- **Input/Output**

Estas funciones se las usa con una sentencia *pinMode*, define un pin digital como Input o como Output.

```
// Set pins as outputs:  
//pinMode(motorA1, OUTPUT);  
pinMode(motorA2, OUTPUT);  
pinMode(motorB1, OUTPUT);  
pinMode(motorB2, OUTPUT);  
pinMode(salA, OUTPUT);  
pinMode(salB, OUTPUT);  
pinMode(salC, OUTPUT);
```

Figura 4.19 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

4.3.2.5. Control de flujo.

- **If**

Es una sentencia que verifica si la condición ha sido alcanzada, si es así ejecuta todas las acciones que están dentro de llaves; si la condición es falsa simplemente el programa ignora la sentencia.

```
if (state == 'f' && estado == 0) {  
    abre();  
    estado = 1;  
    detector = 0;  
}
```

Figura 4.20 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

- **Else If**

Dicha sentencia permite tomar decisiones entre una cosa u otra.

```
else if (state == 'i' && estado == 1) {
    cierra();
    estado = 0;
    detector = 0;
}
```

Figura 4.21 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

- **For**

Esta sentencia es usada para que un bloque de declaraciones sea repetido un cierto número de veces.

```
for (int x = 0; x < sizeof(card1); x++)
{
    for (int i = 0; i < sizeof(rfid.serNum); i++)
    {
        if (rfid.serNum[i] != card1[x][i]) {
            access = false;
            break;
        }
    }
}
```

Figura 4.22 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

4.3.2.6. Entradas y salidas digitales.

- **pinMode**

Se usa con la sentencia *void setup ()* para determinar un pin que realice la función de Input u Output.

Los pines digitales de Arduino están declarados por defecto como Input y tienen un estado de impedancia alta.

```
void setup() {  
  
  Serial.begin(9600);  
  SPI.begin();  
  rfid.init();  
  
  // Set pins as outputs:  
  pinMode(arranca, OUTPUT);  
  pinMode(bocina, OUTPUT);  
  // Initialize serial communication at 9600 bits per second:  
  
  pinMode(inA, INPUT_PULLUP);  
  pinMode(inB, INPUT_PULLUP);  
  pinMode(inC, INPUT_PULLUP);  
  pinMode(inD, INPUT_PULLUP);  
  pinMode(inE, INPUT_PULLUP);  
}
```

Figura 4.23 Tomado del código de programación del contacto del vehículo
Fuente: Israel Cevallos

- **digitalRead**

Lee el valor de un pin digital determinado, ya sea este con High o Low variable o constante.

```
int entr1 = digitalRead(inA);  
int entr2 = digitalRead(inB);  
int entr3 = digitalRead(inC);  
int entr4 = digitalRead(inD);  
int entr5 = digitalRead(inE);
```

Figura 4.24 Tomado del código de programación del contacto del vehículo
Fuente: Israel Cevallos

- **digitalWrite**

Devuelve el nivel de High o Low a un pin digital específico.

```
else if (state == 'c') {  
  mySerial.println("cajuela open");  
  //Serial.println("cajuela open");  
  delay(100);  
  digitalWrite(bt, LOW);  
  delay(100);  
  digitalWrite(bt, HIGH);  
  delay(100);  
  state = '0';  
}
```

Figura 4.25 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

4.3.2.7. Tiempo.

- **Delay**

Es la cantidad de tiempo que se pausa el programa o que se activa una señal y produce una acción, el tiempo se lo especifica en milisegundos, por lo que 1000 equivale a 1 segundo.

```
void alarma() {  
  digitalWrite(salA, HIGH);  
  delay(400);  
  digitalWrite(salA, LOW);  
  delay(400);  
}
```

Figura 4.26 Tomado del código de programación de las puertas
Fuente: Israel Cevallos

4.3.2.8. Serie.

- **Serial.begin**

Da apertura al puerto serie y permite la transmisión de datos en serie, la velocidad más común es 9600 baudios.

“Los baudios es la unidad de medida utilizada en comunicaciones, hace referencia al número de intervalos elementales por segundo que supone una señal.” (MASTER MAGAZINE, s.f.)

```
void setup() {  
  Serial.begin(9600);  
}
```

Figura 4.27 Tomado del código de programación del contacto del vehículo
Fuente: Israel Cevallos

- **Serial.println**

Permite imprimir datos al puerto serie, es decir permite visualizar acciones que realiza la placa de Arduino.

```
mySerial.println("ISRA, ARRANCA");  
Serial.println("ISRA, ARRANCA");  
state = '0';
```

Figura 4.28 Tomado del código de programación del contacto del vehículo
Fuente: Israel Cevallos

4.3.2.9. Código ASCII.

```
1010101000110101110101010001101011101010100011  
0101110101010001101011101010100011010111010101  
0001101011101#include <stdio.h>10101000110101110  
1010100011010111010101000110101110101010001101  
int main()  
01110101010001101011101010100011010111010100  
01101011101010100printf("Hola mundo");1010111010  
return 0;  
10100011010111010101000110101101001010100011  
0101110101010001101011010100011010111010101  
0001101011101010100011010111010101000110101110  
10101010101010101010101010101010101010101010101  
PROGRAMACIÓN  
1010001101010101010101010101010101010101010101  
110101010001101011100110101010001101011010010
```

Figura 4.29 Programación con código ASCII
Fuente: <http://ocw.uc3m.es/ingenieria-informatica/programacion-en-lenguaje-c-2013>

El código hexadecimal hoy en día tiene mucha acogida en el ámbito de las ciencias computacionales, ya que las computadoras usan el byte como unidad de memoria, en donde corresponde un byte a dos dígitos hexadecimales.

En cuanto al código ASCII (*American Standard Code for Information Interchange*), es usado ya que las computadoras guardan información en formato digital y también para intercambio de información.

Tabla 4.1 Código ASCII, Hexadecimal y su simbología

ASCII Hex Símbolo	ASCII Hex Símbolo	ASCII Hex Símbolo	ASCII Hex Símbolo
0 0 NUL	16 10 DLE	32 20 (space)	48 30 0
1 1 SOH	17 11 DC1	33 21 !	49 31 1
2 2 STX	18 12 DC2	34 22 "	50 32 2
3 3 ETX	19 13 DC3	35 23 #	51 33 3
4 4 EOT	20 14 DC4	36 24 \$	52 34 4
5 5 ENQ	21 15 NAK	37 25 %	53 35 5
6 6 ACK	22 16 SYN	38 26 &	54 36 6
7 7 BEL	23 17 ETB	39 27 '	55 37 7
8 8 BS	24 18 CAN	40 28 (56 38 8
9 9 TAB	25 19 EM	41 29)	57 39 9
10 A LF	26 1A SUB	42 2A *	58 3A :
11 B VT	27 1B ESC	43 2B +	59 3B ;
12 C FF	28 1C FS	44 2C ,	60 3C <
13 D CR	29 1D GS	45 2D -	61 3D =
14 E SO	30 1E RS	46 2E .	62 3E >
15 F SI	31 1F US	47 2F /	63 3F ?
ASCII Hex Símbolo	ASCII Hex Símbolo	ASCII Hex Símbolo	ASCII Hex Símbolo
64 40 @	80 50 P	96 60 `	112 70 p
65 41 A	81 51 Q	97 61 a	113 71 q
66 42 B	82 52 R	98 62 b	114 72 r
67 43 C	83 53 S	99 63 c	115 73 s
68 44 D	84 54 T	100 64 d	116 74 t
69 45 E	85 55 U	101 65 e	117 75 u
70 46 F	86 56 V	102 66 f	118 76 v
71 47 G	87 57 W	103 67 g	119 77 w
72 48 H	88 58 X	104 68 h	120 78 x
73 49 I	89 59 Y	105 69 i	121 79 y
74 4A J	90 5A Z	106 6A j	122 7A z
75 4B K	91 5B [107 6B k	123 7B {
76 4C L	92 5C \	108 6C l	124 7C
77 4D M	93 5D]	109 6D m	125 7D }
78 4E N	94 5E ^	110 6E n	126 7E ~
79 4F O	95 5F _	111 6F o	127 7F

Fuente: <http://ascii.cl/es/>

Para la aplicación desarrollada en el proyecto, CORSA_C, en el código de Arduino se utiliza códigos hexadecimales que proporcionan un símbolo por cada uno, es decir, para abrir los seguros del vehículo se usa el número “66” en código hexadecimal que equivale a su símbolo “f” que es la letra que enviará el Bluetooth del Smartphone como condición al Arduino para que realice la acción de elevar los seguros del vehículo.

Para el caso de asegurar el vehículo el número que se usa es el “69” en código hexadecimal y su equivalente en símbolo es “i”; para la apertura de la cajuela se usa el “63” en código hexadecimal y su equivalente en símbolo es “c”; en cuanto al contacto del vehículo se usa el “61” con su símbolo equivalente de “a”; y por último el apagado del vehículo con el código hexadecimal “62” y su símbolo “b”.



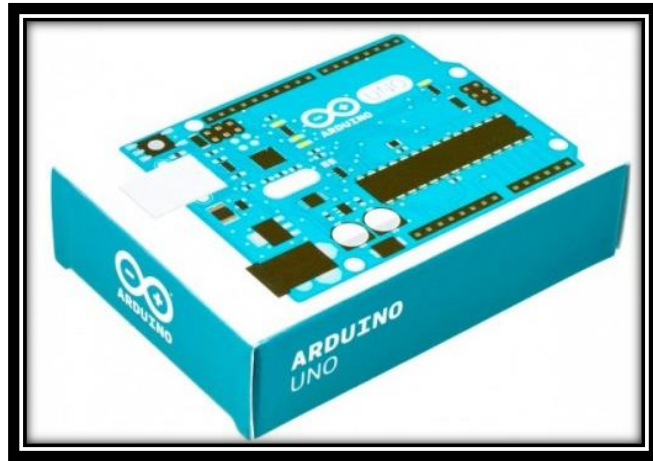
Figura 4.30 Chevrolet Corsa Evolution equipado con Arduino
Fuente: Israel Cevallos

4.3.3. Comparación Arduino UNO original vs genérico.

Tabla 4.2 Arduino UNO original vs genérico

CARACTERÍSTICAS	ARDUINO ORIGINAL	ARDUINO GENÉRICO
EMPAQUE	Caja diseñada exclusivamente para Arduino UNO	Funda plástica sellada
MANUAL DE USUARIO	Incluye	No incluye
CABLE USB	No incluye	Incluye
GRÁFICOS IMPREGNADOS	Calidad superior	Calidad baja
ORIGEN	Made in Italy	Design in Italy
PRECIO	\$50	\$20

Fuente: Israel Cevallos



VS

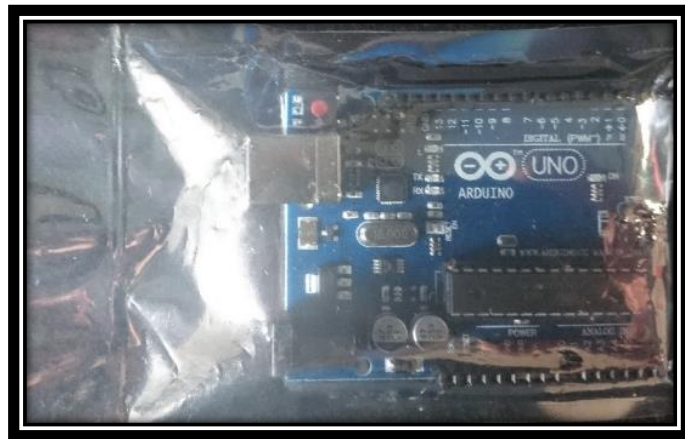


Figura 4.31 Arduino UNO original vs genérico
Fuente: Israel Cevallos

4.3.4. Bluetooth HC-05

4.3.4.1. Introducción.

El mando Bluetooth HC-05, es un módulo maestro-esclavo, es decir que puede recibir conexiones desde un computador, y otros dispositivos que dispongan de conexión Bluetooth.

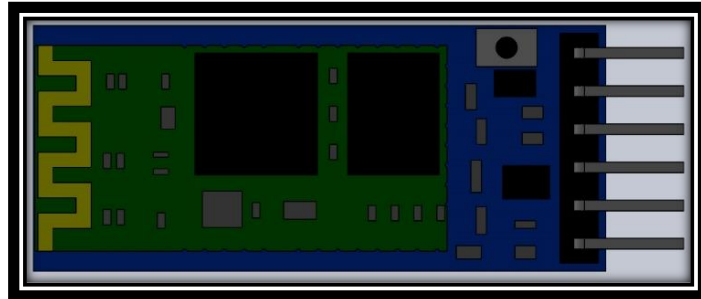


Figura 4.32 Módulo Bluetooth HC-05 diseñado en SolidWorks

Fuente: Israel Cevallos

4.3.4.2. Módulo Bluetooth HC-05 como maestro.

En el instante en que está en modo maestro, éste se conecta con un dispositivo que es esclavo, es decir se puede conectar con otro Bluetooth similar.

4.3.4.3. Módulo Bluetooth HC-05 como esclavo.

El Bluetooth está en esta modalidad, en el momento en que espera a que un dispositivo maestro se conecte a este, en este caso con un Smartphone Sony Xperia Z5.

4.3.4.4. Configuración inicial.

El módulo Bluetooth por defecto viene como esclavo con el nombre de HC-05 y un código de emparejamiento de “1234” o “0000”.

Para realizar el cambio de estos valores por defecto se puede programar el cambio de nombre o clave de acceso por medio de los siguientes códigos:

Tabla 4.3 Comando AT

LISTA DE COMANDOS AT	
COMANDO	FUNCIÓN
AT	Test UART Connection
AT+RESET	Reset Device
AT+VERSION	Query firmware version
AT+ORGL	Restore settings to Factory Defaults
AT+ADDR	Query Device Bluetooth Address
AT+NAME	Query/Set Device Name
AT+RNAME	Query Remote Bluetooth Device's
AT+ROLE	Query/Set Device Role
AT+CLASS	Query/Set Class of Device CoD
AT+IAC	Query/Set Inquire Access Code
AT+INQM	Query/Set Inquire Access Mode
AT+PSWDAT+PIN	Query/Set Pairing Passkey
AT+UART	Query/Set UART parameter
AT+CMODE	Query/Set Connection Mode
AT+BIND	Query/Set Binding Bluetooth Address
AT+POLAR	Query/Set LED Output Polarity
AT+PIO	Set/Reset a User I/O pin

Fuente: Modificado por Israel Cevallos

Obtenido de: <http://www.prometec.net/bt-hc05/>

De esta manera se aplica el cambio de HC-05 por CORSA_C y la clave por defecto se cambia a EVOLUTION

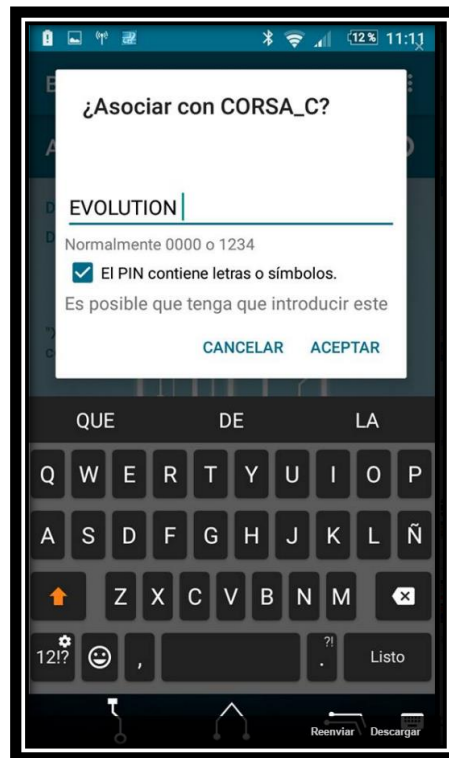


Figura 4.33 Emparejamiento al Bluetooth CORSA_C desde dispositivo móvil
Fuente: Israel Cevallos

4.3.4.5. Diagrama de conexión Bluetooth HC-05 con Arduino UNO.

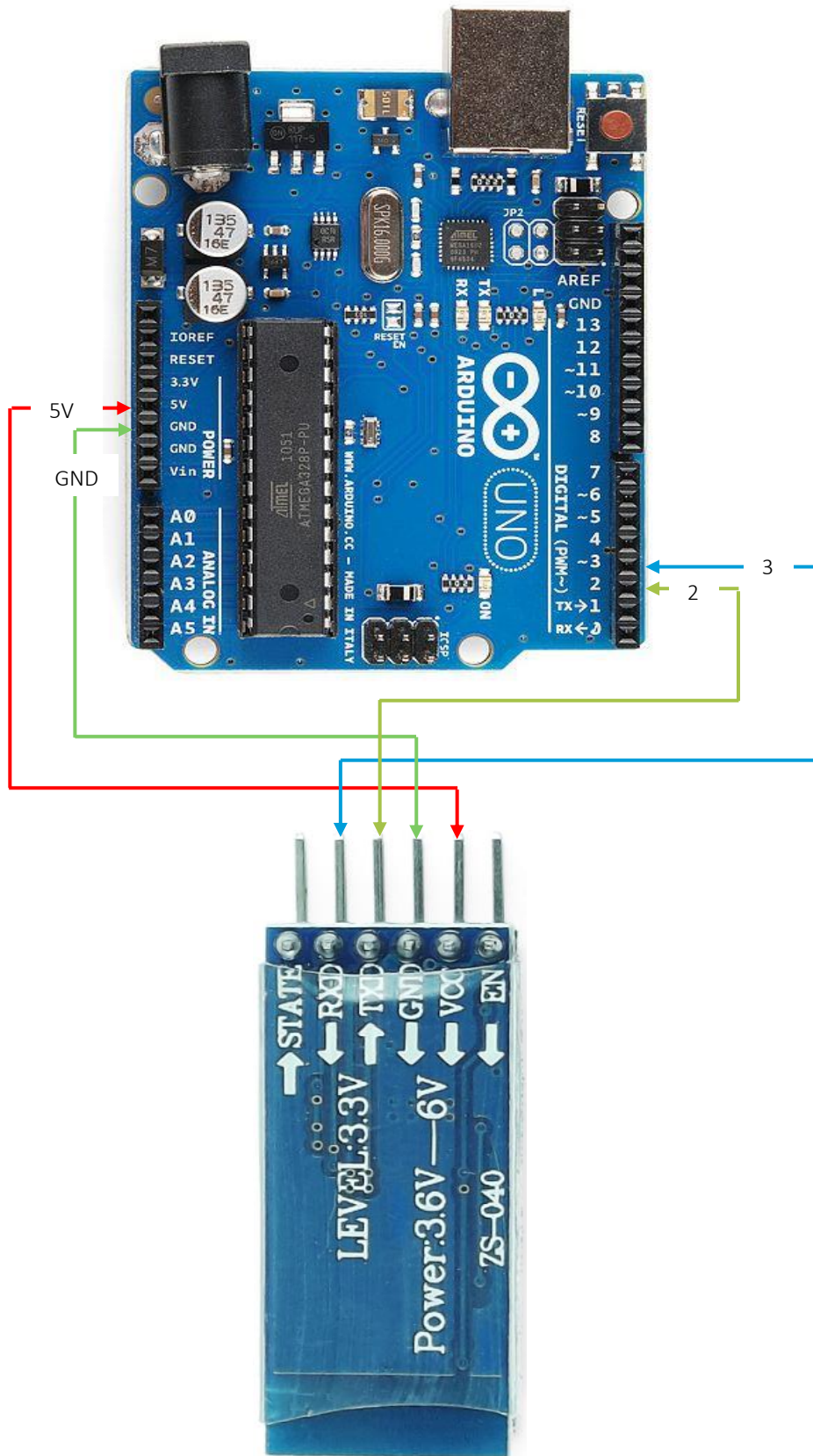


Figura 4.34 Diagrama de conexión Bluetooth HC-05 con Arduino UNO
Fuente: Israel Cevallos

4.3.5. NFC

Se trata de una tecnología inalámbrica que funciona en la banda de los 13.56 MHz, es una plataforma abierta pensada desde el inicio para teléfonos y dispositivos móviles.

Cada dispositivo NFC necesita de una antena que permite el envío y recepción de datos.

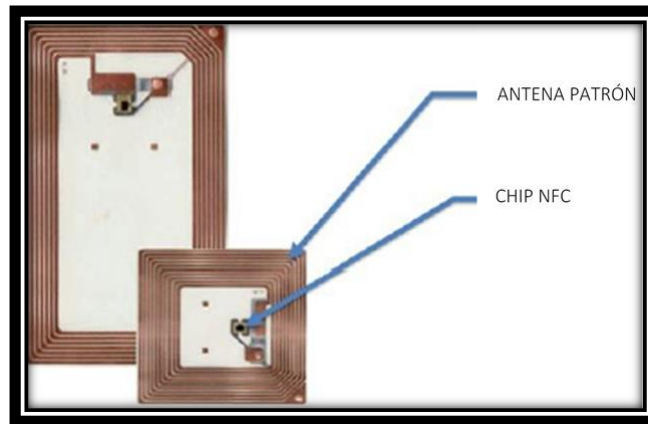


Figura 4.35 Partes internas de un dispositivo NFC
Fuente: Israel Cevallos

4.3.5.1. Modo de envío y recepción de datos.

Los Smartphone con tecnología NFC usan un bucle inductor llamado antena, que utiliza la corriente alterna para producir un campo magnético continuo y de esta manera se crea un lazo y acoplamiento entre otra antena receptora.



Figura 4.36 Modo de envío y recepción de datos
Fuente: (Sabella, NFC FOR DUMMIES, 2016)

El mecanismo funciona inicialmente con el bucle o antena del dispositivo móvil y la unión de un bucle que recibirá los datos, por lo tanto, se forma un transformador de núcleo de aire, en el caso de que haya interrupción de corriente los lazos de campo magnético desaparecerán por completo y no habrá comunicación.

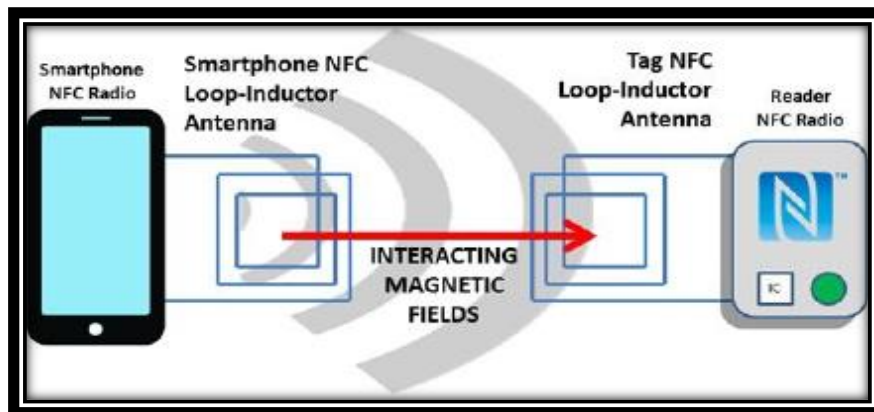


Figura 4.37 Tecnología inalámbrica
Fuente: (Sabella, NFC FOR DUMMIES, 2016)

4.3.6. RFID

Este sistema funciona por medio de radio frecuencia, en donde RFID puede almacenar información única y responde a la acción de una tarjeta o tag previamente configurados.

RFID trabaja mediante ondas de radio frecuencia, lo que permite el reconocimiento de un elemento en particular que posea este sistema.

4.3.6.1. Diagrama tarjeta RFID.

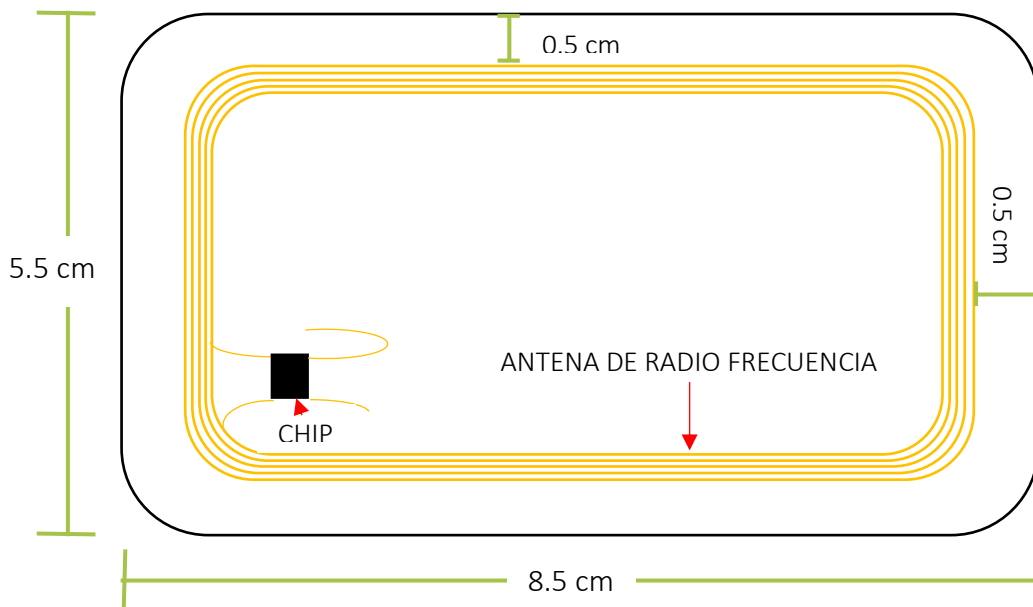


Figura 4.38 Parte interna de una tarjeta RFID
Fuente: Israel Cevallos

4.3.6.2. Diagrama tag RFID.

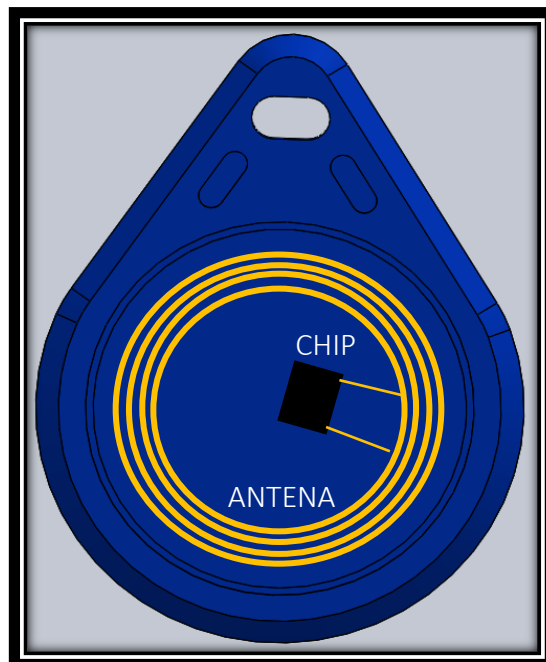


Figura 4.39 Parte interna de un Tag RFID
Fuente: Israel Cevallos

4.3.6.3. Diagrama de conexión módulo RFID con Arduino UNO.

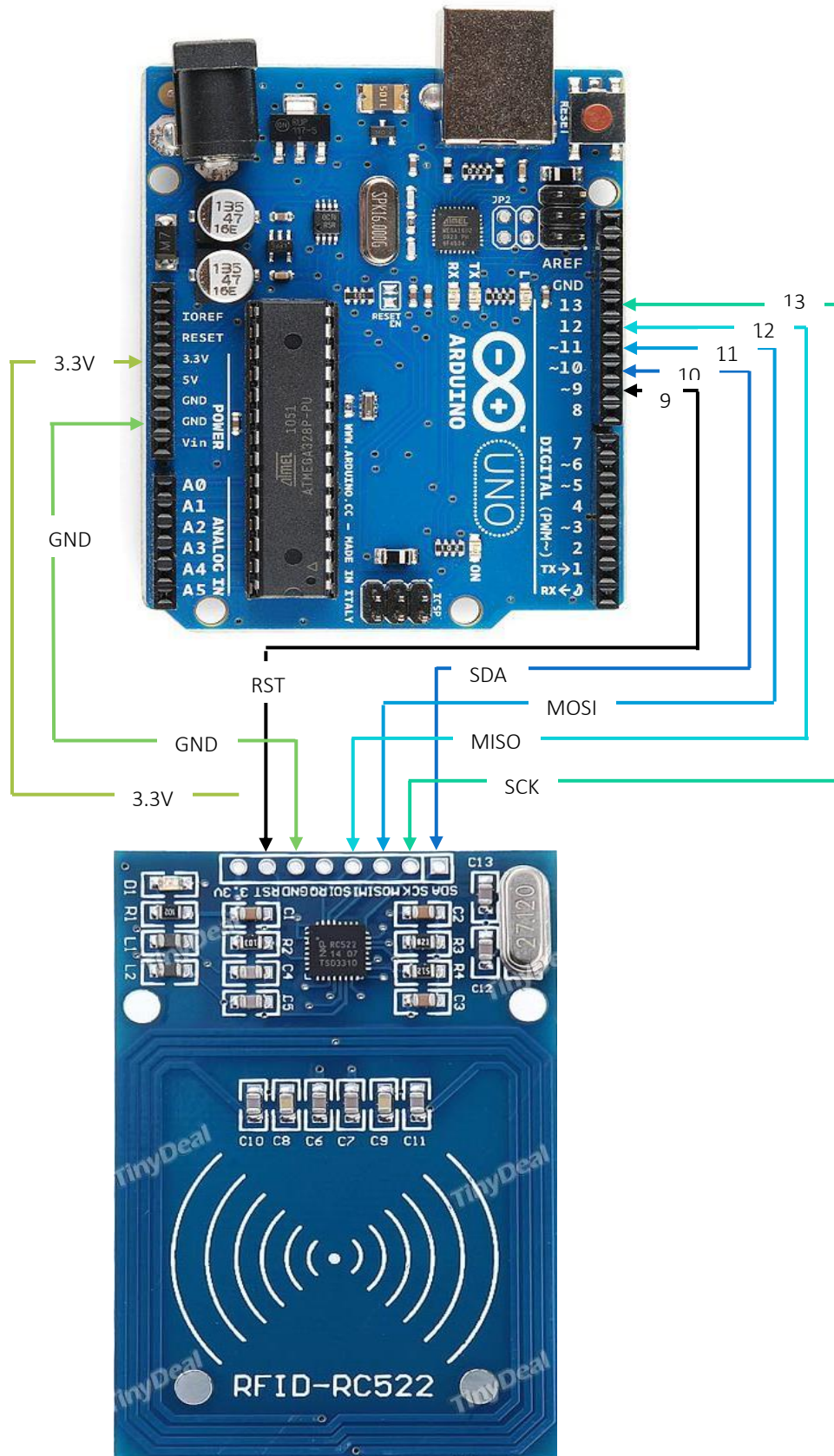


Figura 4.40 Diagrama de conexión módulo RFID con Arduino UNO
Fuente: Israel Cevallos

4.3.7. Características necesarias de los dispositivos móviles.

Tabla 4.4 Características de los Smartphone.

CARACTERÍSTICAS	Sony Xperia Z5 Compact	Samsung Galaxy S7	LG G4	Motorola X Play	Huawei Mate 8	Nokia Lumia 930
Sistema Operativo	Android OS, v5.1 Lollipop	Android OS, v6.0 Marshmallow	Android OS, v5.1 Lollipop	Android OS, v5.1 Lollipop	Android OS, v6.0 Marshmallow	Microsoft Windows Phone 8.1
Procesador	Qualcomm® Snapdragon™ 810	Snapdragon 820 o Exynos octa-core	Snapdragon 808	Snapdragon 615	Octacore 950	Snapdragon 800
Batería	2 días de duración	-	375 horas de duración	-	-	432 horas de duración
Dimensiones	4,6"	5,1"	5,5"	5,5"	6"	5"
Conectividad Bluetooth	*	*	*	*	*	*
Conectividad NFC	*	*	*	*	*	*

Fuente: Israel Cevallos

 Cumplimiento de conectividad NFC y Bluetooth

4.4. Sistema de arranque, apertura y cierre de puertas.

4.4.1. Bloqueo del volante Chevrolet Corsa Evolution,

El sistema de bloqueo del volante del Chevrolet Corsa Evolution, fue eliminado por completo, debido a que el vehículo poseía un sistema mecánico/manual que permitía este proceso; consecuentemente como se reemplazó el Switch de encendido por un botón, ya no hay manera de desactivar el mecanismo del bloqueo.

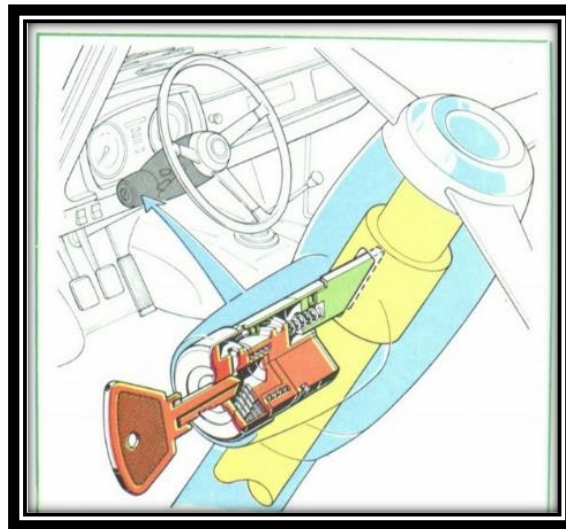


Figura 4.41 Bloqueo mecánico/manual del volante

Fuente: <https://motorgiga.com/fotos-de-coches/foto-anti-robo-antiguo-bloqueo-volante-tecnica-infografias-y-curiosidades/gmx-niv186-con347657.htm>

4.4.2. Diagrama de conexión botón de encendido.

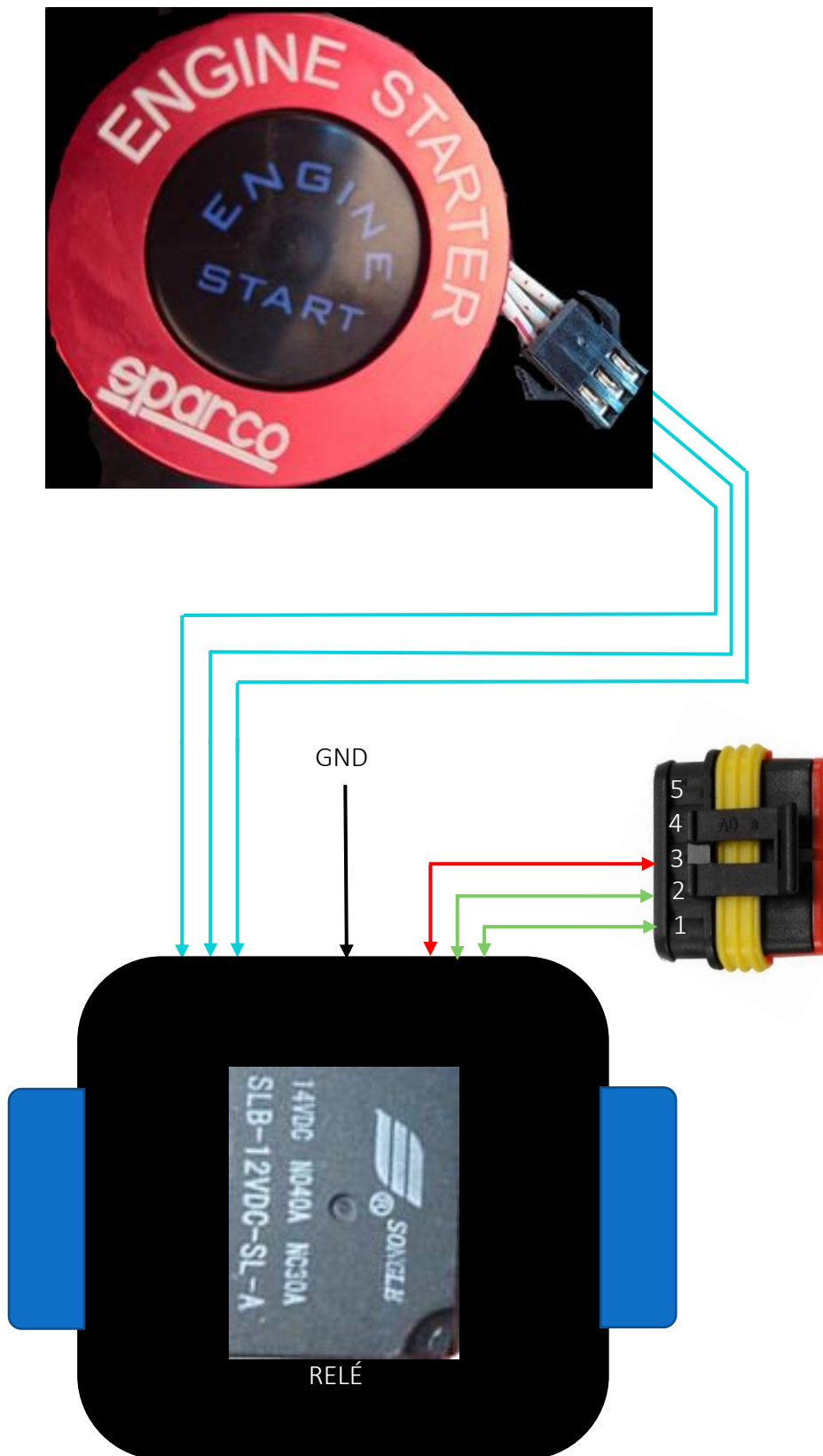


Figura 4.42 Diagrama de conexión botón de encendido
Fuente: Israel Cevallos

4.4.3. Diagrama de conexión actuadores de puertas con un módulo relé.

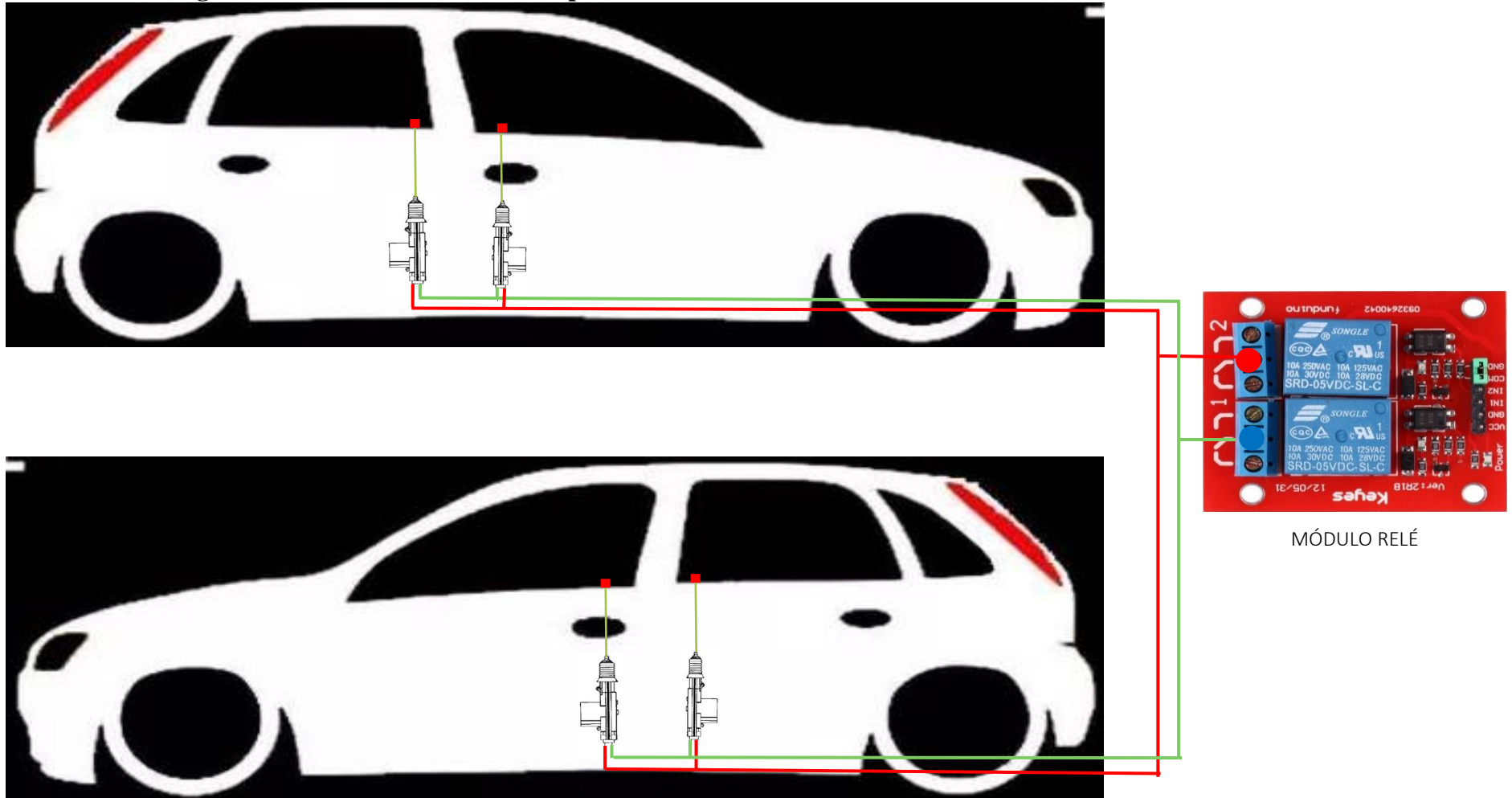
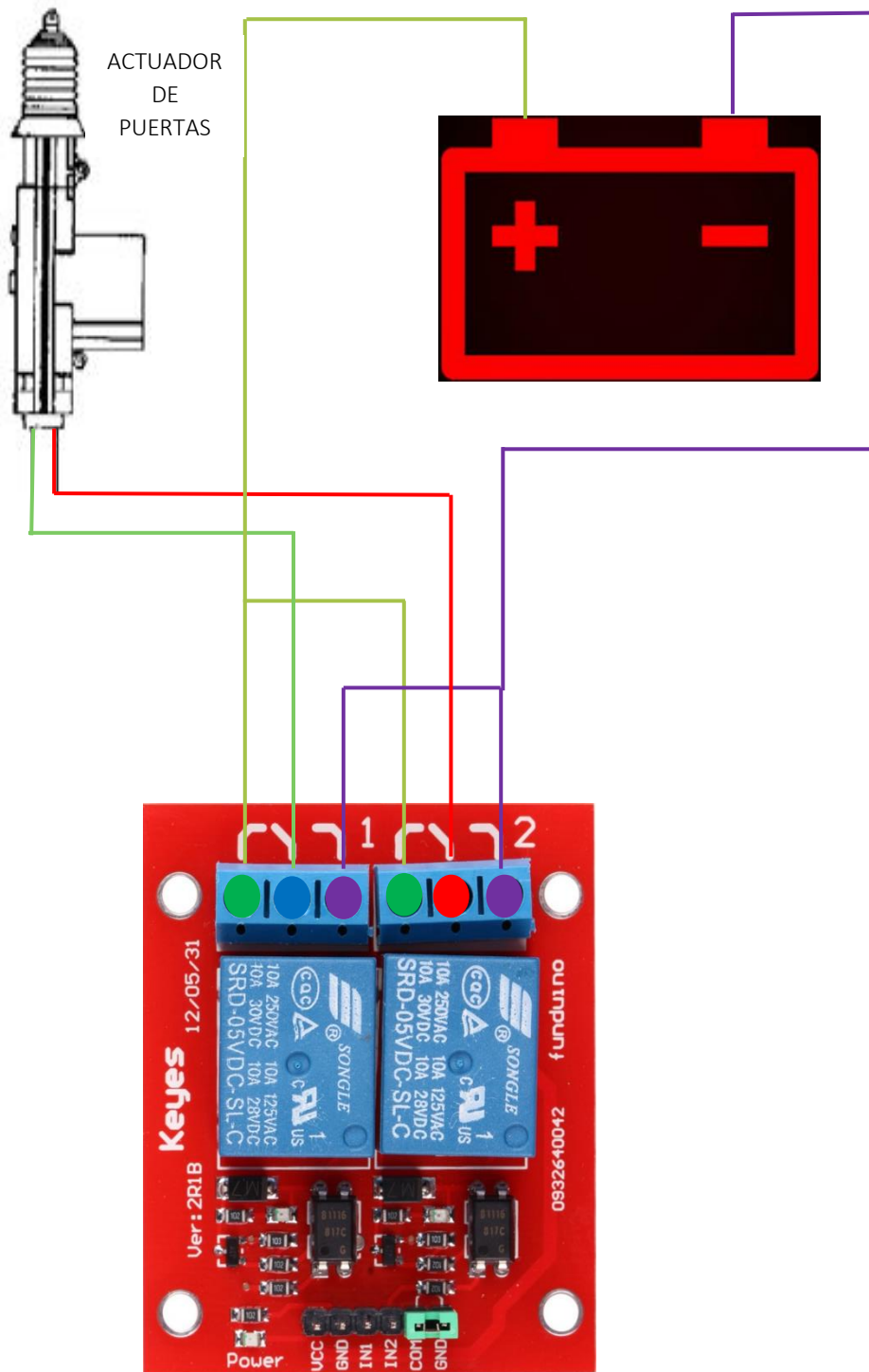


Figura 4.43 Diagrama de conexión actuadores de puertas con un módulo relé
Fuente: Israel Cevallos

4.4.4. Diagrama de alimentación del módulo relé y los actuadores de puertas.



MÓDULO RELÉ

Figura 4.44 Diagrama de alimentación del módulo relé y los actuadores de puertas
Fuente: Israel Cevallos

4.4.5. Diagrama de conexión del módulo relé con el Arduino UNO del contacto del vehículo.

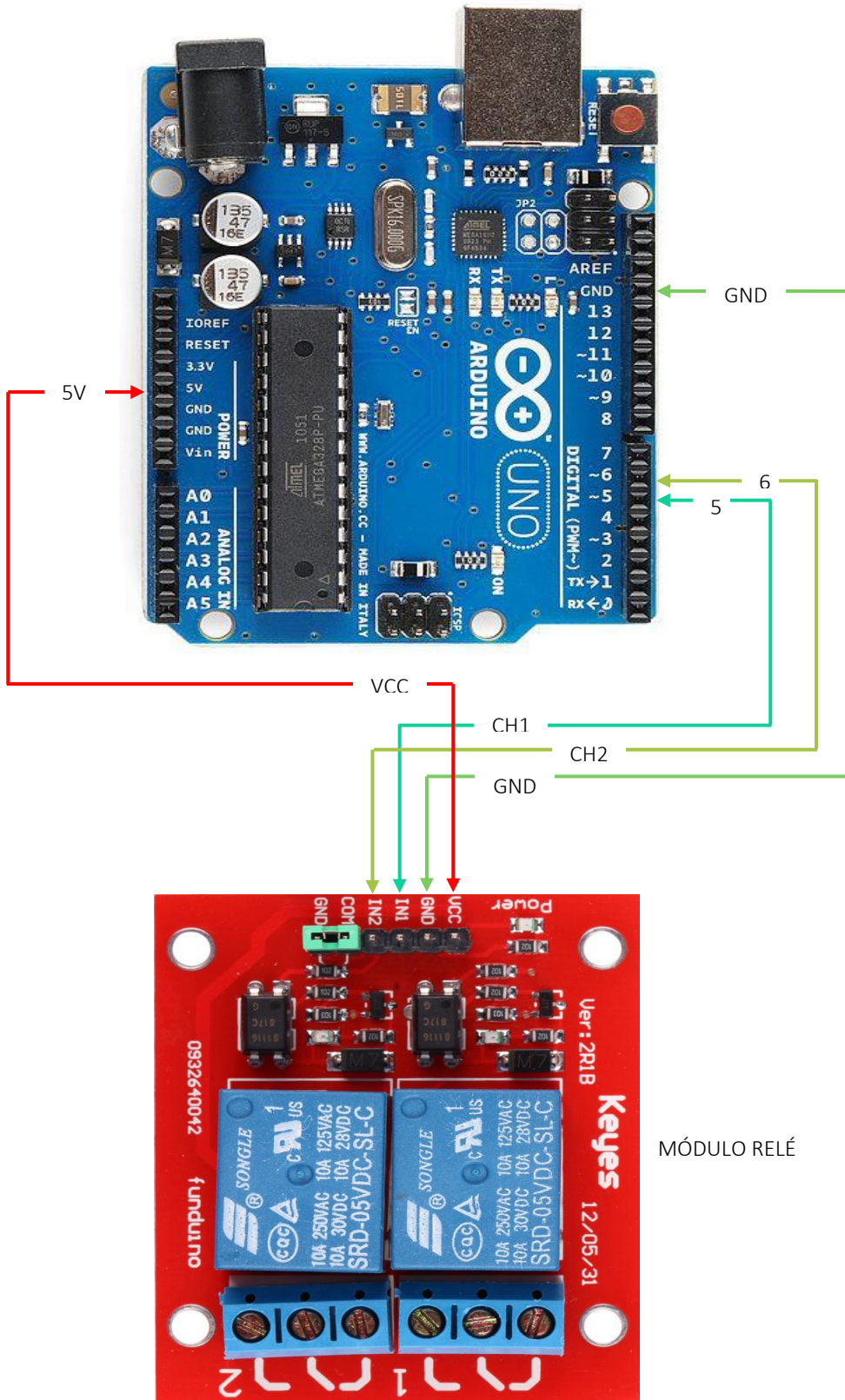


Figura 4.45 Diagrama de conexión del módulo relé con Arduino UNO del contacto del vehículo
Fuente: Israel Cevallos

4.4.6. Diagrama de conexión del contacto del vehículo con el módulo relé y la bocina.

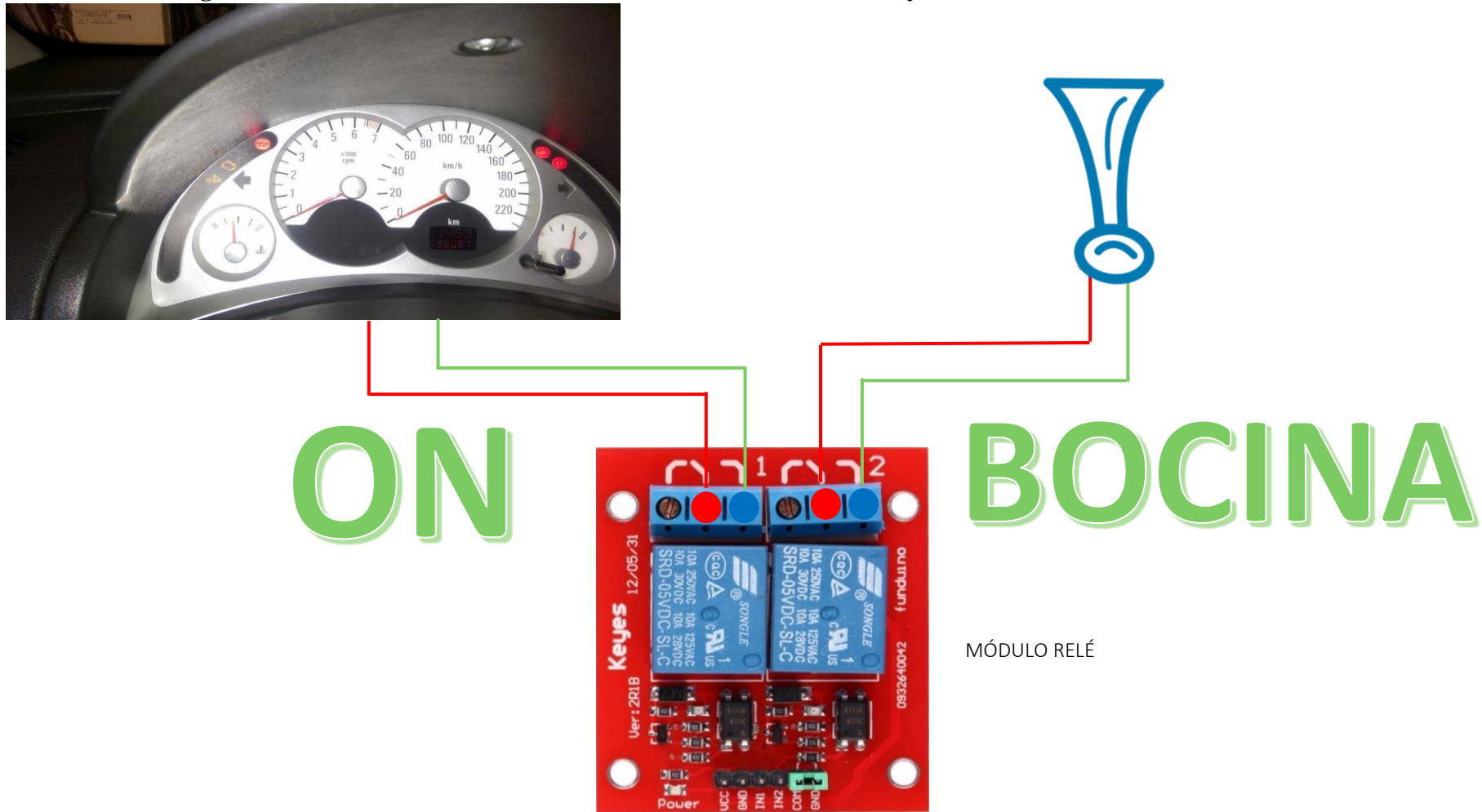


Figura 4.46 Diagrama de conexión del contacto del vehículo con el módulo relé y la bocina.

Fuente: Israel Cevallos

4.4.7. Diagrama de conexión y alimentación de la cajuela del vehículo con el módulo relé y el actuador.



Figura 4.47 Diagrama de conexión y alimentación de la cajuela del vehículo con el módulo relé y el actuador.
Fuente: Israel Cevallos

4.4.8. Diagrama de comunicación entre el Arduino UNO de las puertas con el Arduino UNO del contacto del vehículo.

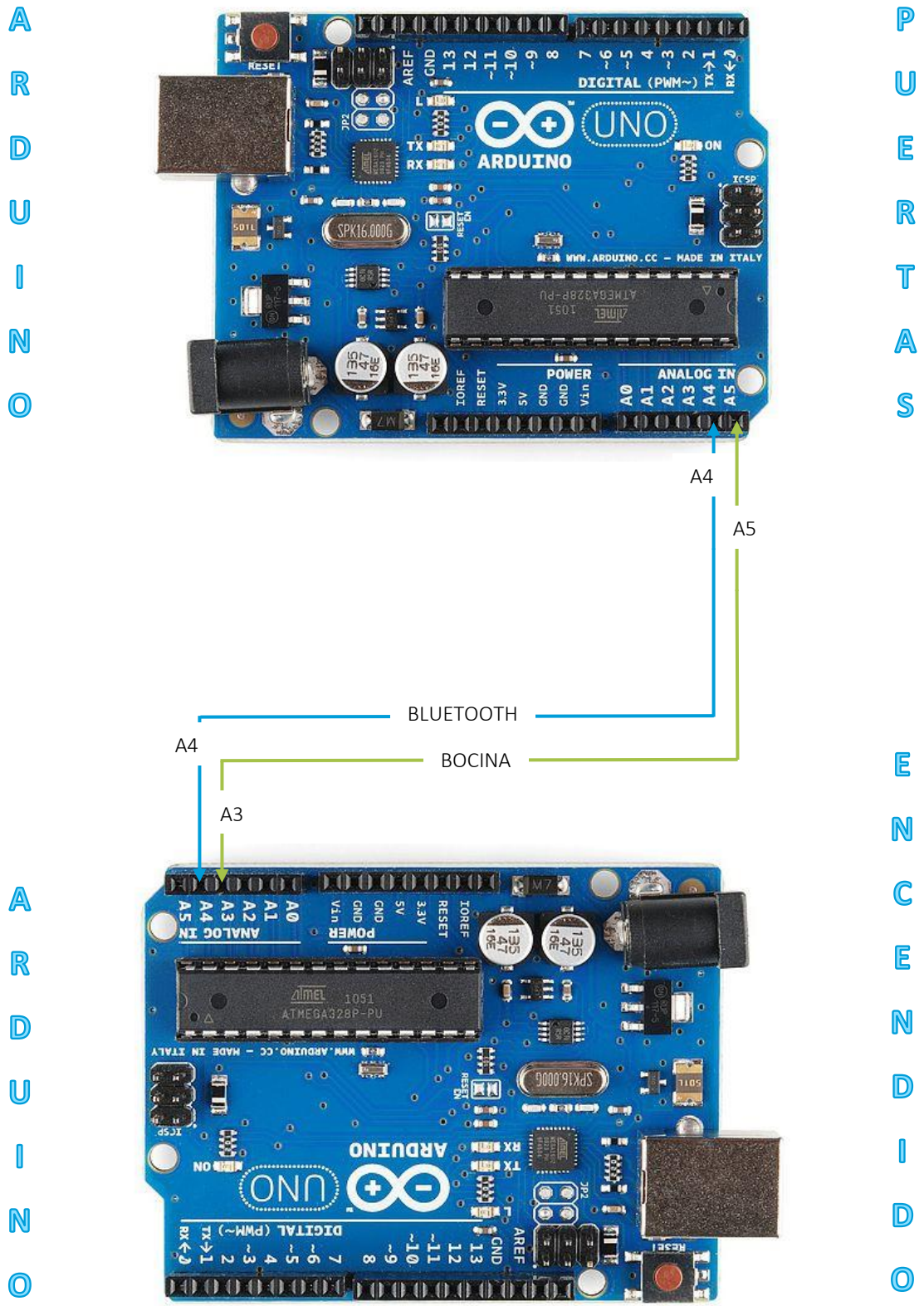


Figura 4.48 Diagrama de comunicación entre Arduino UNO de las puertas con el Arduino UNO del contacto del vehículo
Fuente: Israel Cevallos

4.4.9. Diagrama de comunicación entre el Arduino UNO de las puertas con el Arduino UNO de la cajuela.

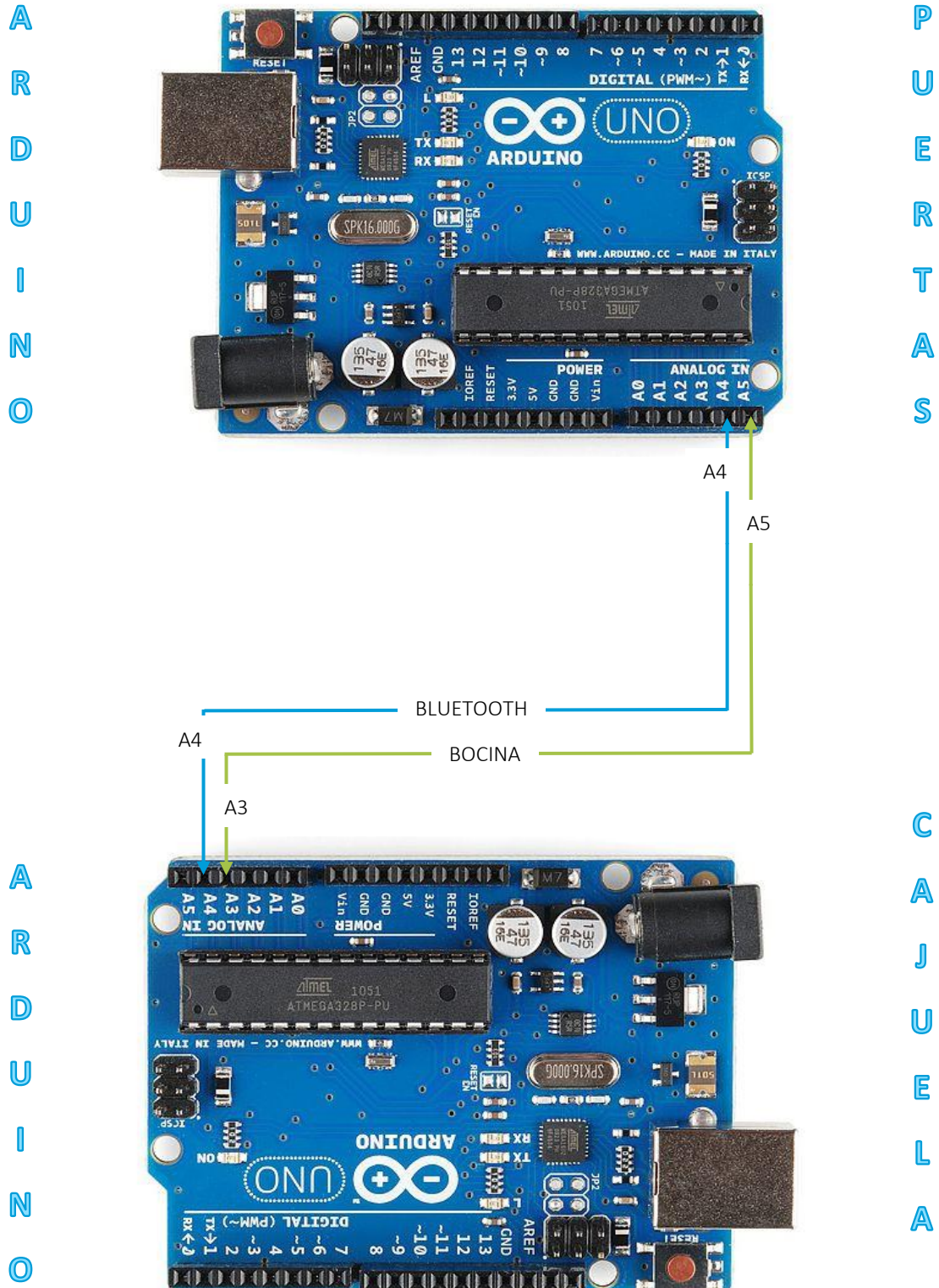


Figura 4.49 Diagrama de comunicación entre Arduino UNO de las puertas con el Arduino UNO de la cajuela
Fuente: Israel Cevallos

4.4.10. Diagrama de comunicación entre el Arduino UNO de la caja con el Arduino UNO del contacto del vehículo.

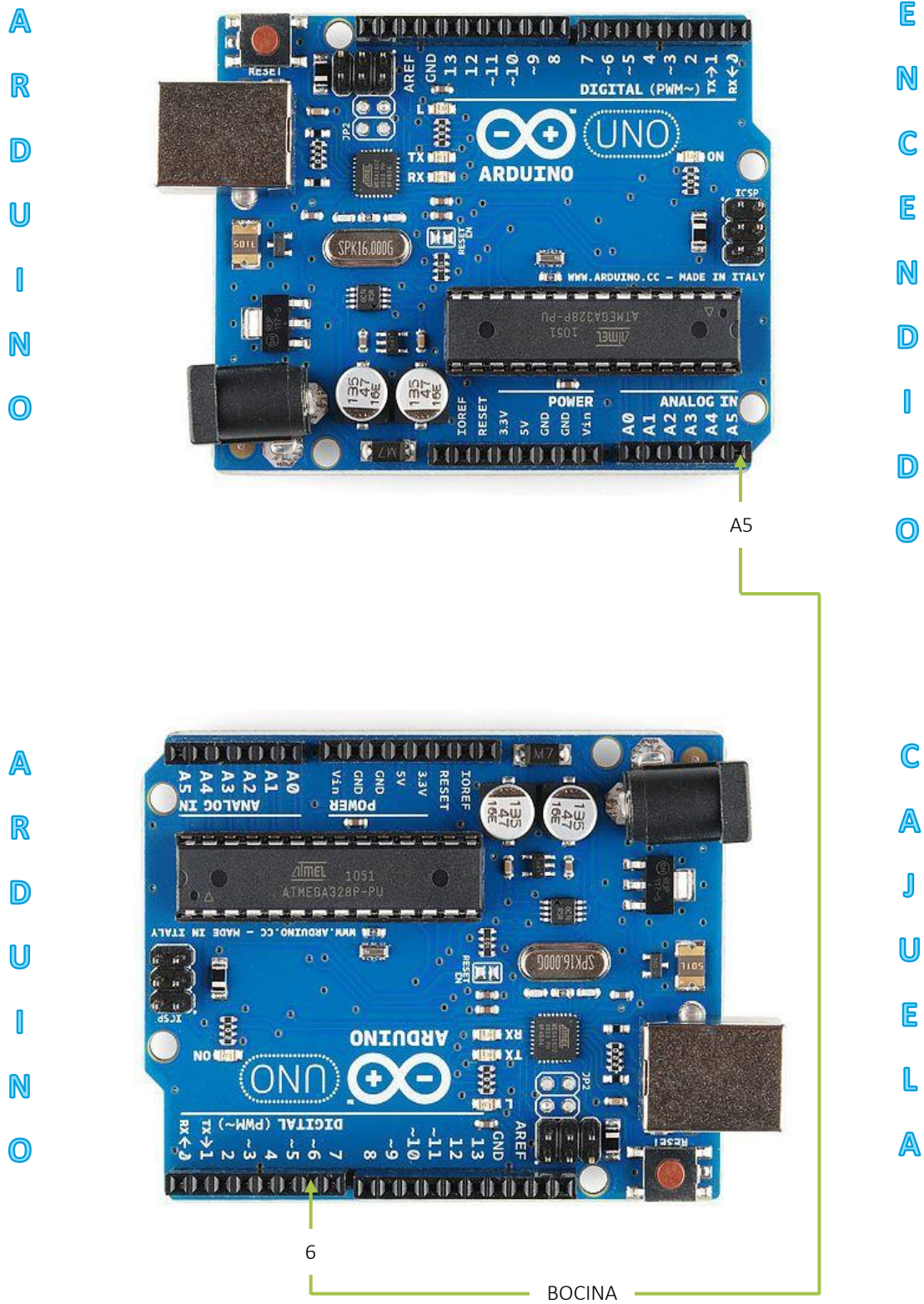


Figura 4.50 Diagrama de comunicación entre Arduino UNO de la caja con el Arduino UNO del contacto del vehículo.
Fuente: Israel Cevallos

4.5. Diseño de la aplicación móvil en App Inventor.

4.5.1. Introducción.

App Inventor es un entorno de programación en el cual se desarrollan aplicaciones móviles, para este proyecto, la aplicación desarrollada es CORSA_C". App Inventor permite desarrollar Apps para dispositivos Android.



Figura 4.51 Bienvenido a MIT App Inventor

Fuente: <http://www.conectados.blogsek.es/index.php/app-inventor-un-mundo-de-posibilidades/>

Es una innovadora herramienta para la creación de aplicaciones, que se encuentra en la nube y se puede ejecutar mediante cualquier navegador de internet, ya sea este Chrome, Firefox, Safari, que son los más comunes y recomendados; aquí se usa la construcción visual y ya no el lenguaje de programación estructurado como otras plataformas, mantiene una interfaz gráfica que facilita la navegación al usuario.

La principal característica de este entorno de creación de aplicaciones, es que se basa en un editor de bloques que maneja una librería Open Blocks de Java, y usa un lenguaje visual a partir de bloques; dichas librerías son

distribuidas por Massachusetts Institute of Technology (MIT) de licencia libre.

App Inventor cuenta con un “Entorno de desarrollo integrado (IDE)”, es decir que posee herramientas de diseño, desarrollo y una opción de prueba, que permite ejecutar la aplicación como si fuera un dispositivo Android.

Para iniciar la programación, lo primero que se debe hacer es entrar a la dirección Web de MIT App Inventor “<http://appinventor.mit.edu/explore/>”, tener una cuenta de Google ya que está sobre una plataforma de Google Labs, posterior al registro mostrará su página de inicio.

4.5.2. Inicio en App Inventor.

Entrar a su dirección web: <http://appinventor.mit.edu/explore/>

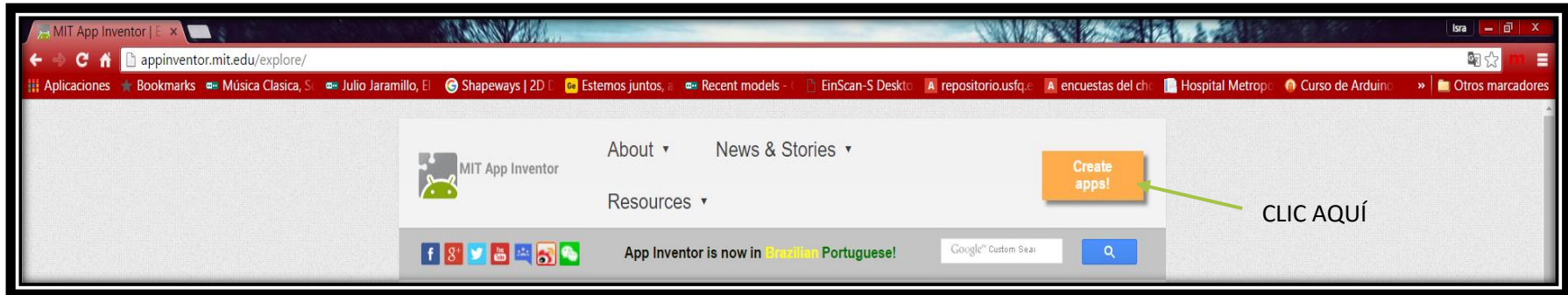


Figura 4.49 Página de inicio App Inventor

Fuente: Israel Cevallos

Ingresar con una cuenta de Google



Figura 4.52 Inicio de sesión o registro con una cuenta de Google

Fuente: Israel Cevallos

+ Página de inicio



Figura 4.51 Barra de herramientas de la página de inicio de App Inventor
Fuente: Israel Cevallos

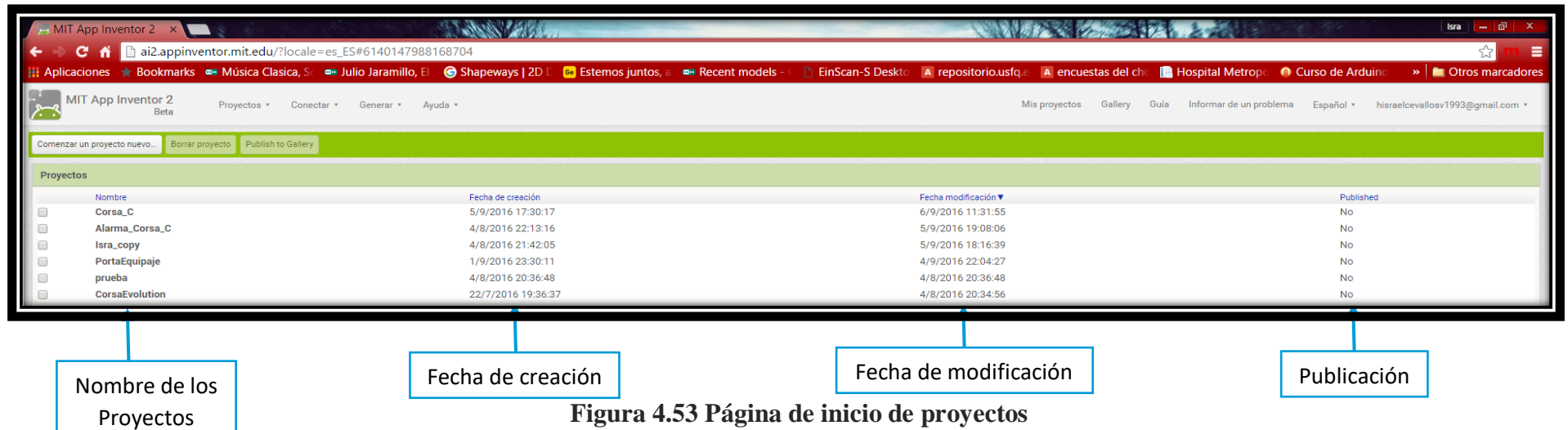


Figura 4.53 Página de inicio de proyectos
Fuente: Israel Cevallos

4.5.3. Entorno de programación.

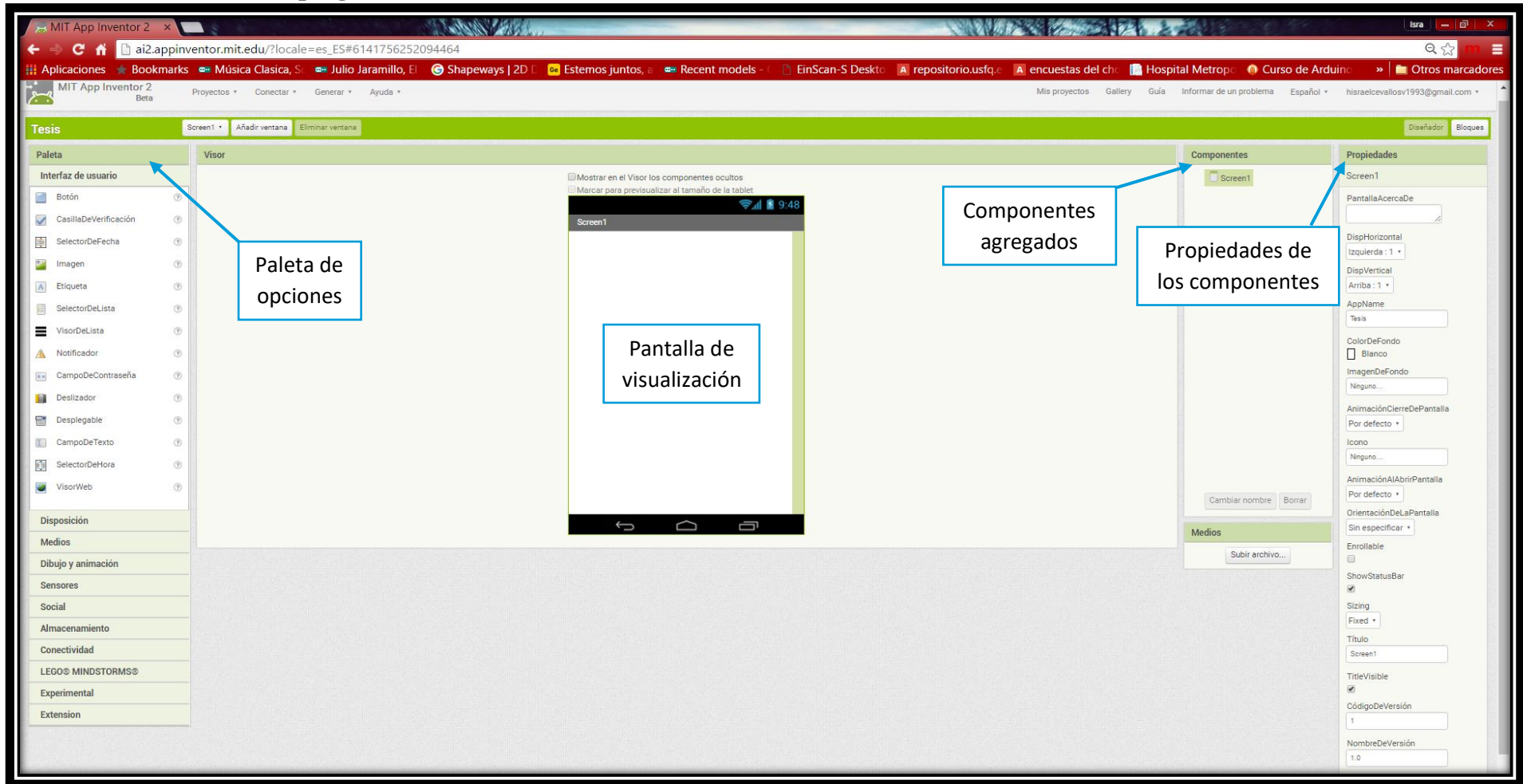


Figura 4.54 Entorno de programación
Fuente: Israel Cevallos

4.5.4. Programación visual.

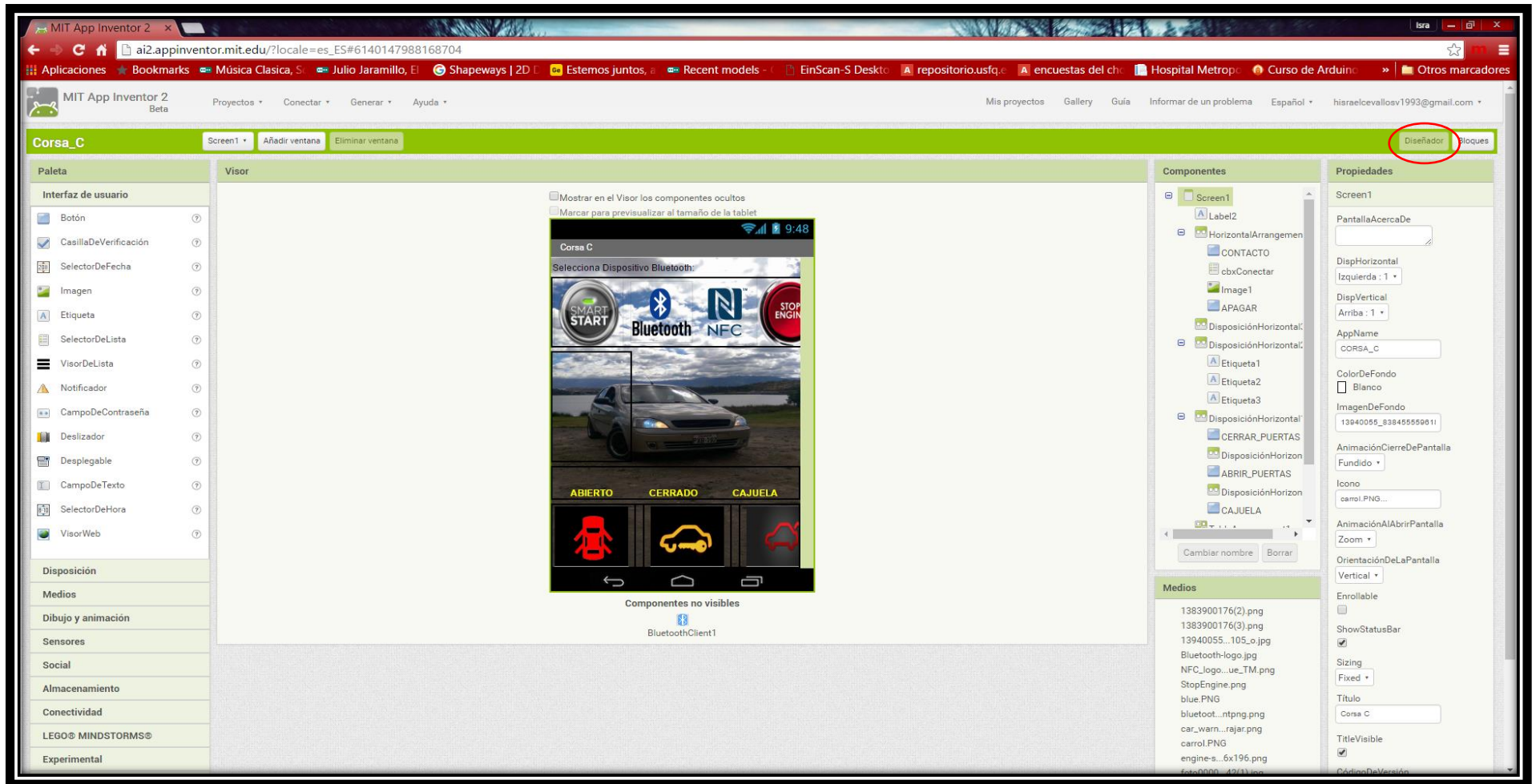


Figura 4.55 Programación visual

Fuente: Israel Cevallos

4.5.5. Programación en bloques.

Mochila/Aquí se alojan secciones de bloques temporalmente que serán usados posteriormente

Basurero/Aquí se eliminan secciones de bloques inservibles o incorrectos

Figura 4.56 Programación en bloques
Fuente: Israel Cevallos

4.5.6. Diseño de la App.

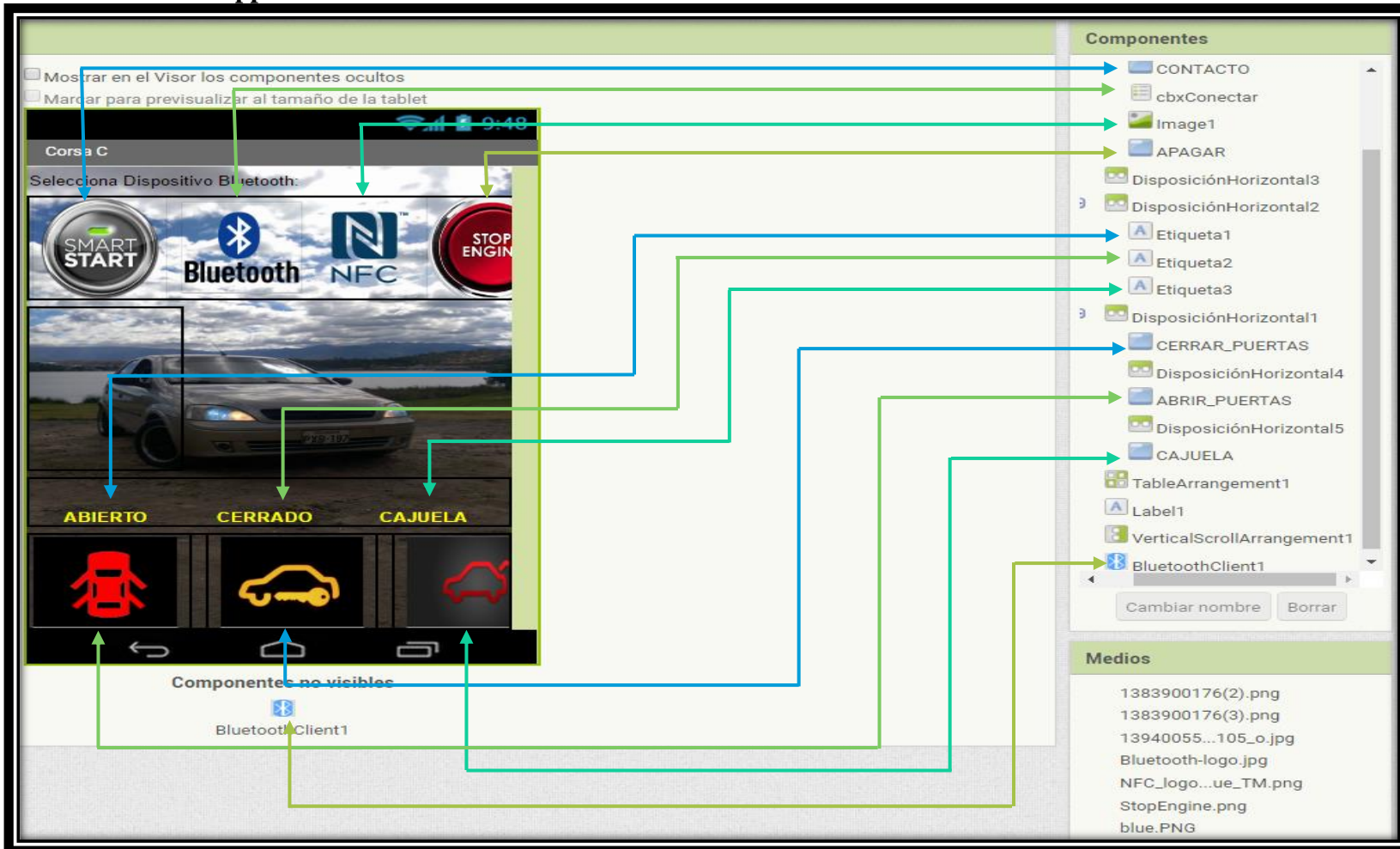


Figura 4.57 Diseño de la App
Fuente: Israel Cevallos

4.5.7. Pre-requisitos para el uso de la App “CORSA_C”.

✚ Pestaña Proyectos “Guardar proyecto como”



Figura 4.58 Guardar proyecto
Fuente: Israel Cevallos

✚ Pestaña Conectar “AI Companion”

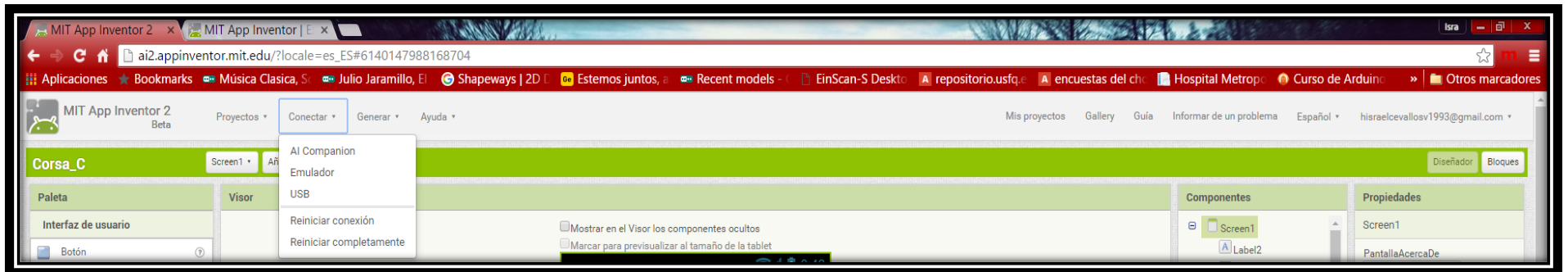


Figura 4.59 Conectar AI Companion
Fuente: Israel Cevallos

Descargar en el Smartphone la app “MIT AI2 Companion” desde Play Store de forma gratuita.



Figura 4.60 Descargar MIT AI2 Companion en el Smartphone
Fuente: Israel Cevallos

- ✚ Abrir MIT AI2 Companion y escanear el código QR o introducir el código emitido por AI Companion, para este proceso se requiere que ambos dispositivos, tanto el generador del código como el Smartphone estén conectados a la misma red.

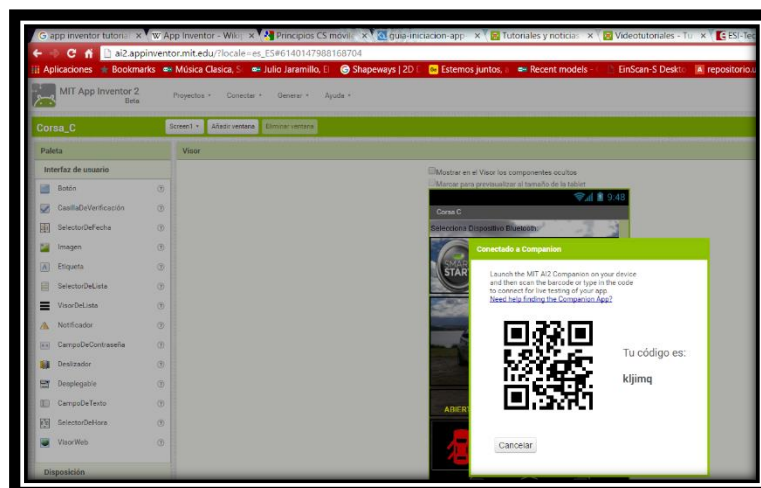


Figura 4.61 Escanear código QR emitido por AI Companion
Fuente: Israel Cevallos

- ✚ En la pantalla del Smartphone se emulará la aplicación y se podrá verificar su funcionamiento.



Figura 4.62 Emulación de la App CORSA_C
Fuente: Israel Cevallos

- ✚ Pestaña Generar “App (guardar archivo. apk en mi ordenador)”



Figura 4.63 Guardar App en formato. apk
Fuente: Israel Cevallos

4.5.8. Aplicación “CORSA_C”.



Figura 4.64 Aplicación CORSA_C
Fuente: Israel Cevallos

El primer paso para que la aplicación funcione es activar el Bluetooth en el menú de configuraciones del Smartphone Android y emparejar con la respectiva contraseña del módulo Bluetooth que posee el vehículo, posterior en la aplicación CORSA_C, presionar el botón Bluetooth que lo que permite es buscar los dispositivos cercanos para conectarse.

La sección superior consta de un botón de START, que permite activar el contacto del vehículo y funcionamiento de los accesorios como el radio, calefacción, etc., el siguiente botón es el de STOP que apagará el vehículo por completo al igual que el funcionamiento de los accesorios.

En la sección inferior se encuentra tres botones los cuales, el primero que es el botón abierto permite abrir los seguros de las cuatro puertas del vehículo, el botón cerrado asegura las cuatro puertas y por último el tercero que abre el seguro de la cajuela.

4.6. Diseño de las cajas de los módulos.

4.6.1. SolidWorks.

SolidWorks es un software CAD (diseño asistido por computadora) para modelado mecánico en 3D, desarrollado en la actualidad por SolidWorks Corp.

SolidWorks nos permite desarrollar productos de mejor calidad y más perfectos, dejándonos observar hasta el más mínimo detalle, que trabaja de manera más rápida y productiva.

Las herramientas de diseño de SOLIDWORKS le pueden ayudar a conceptualizar, visualizar, probar y documentar sus diseños en 3D.

Las herramientas de simulación de SOLIDWORKS le proporcionan un mayor control y una mejor perspectiva para verificar el rendimiento de su diseño.



Figura 4.65 SolidWorks

Fuente: <http://www.hawkridgesys.com/products/solidworks>

4.6.1.1. Diseño caja Arduino UNO.

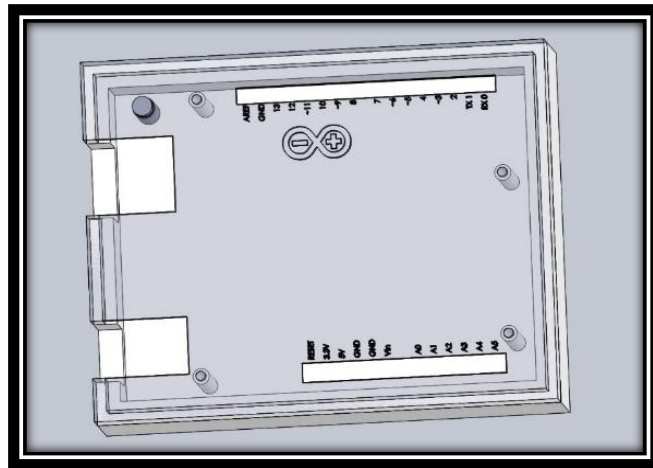


Figura 4.66 Diseño caja de Arduino UNO en SolidWorks
Fuente: Israel Cevallos

4.6.1.2. Diseño caja módulo RFID.

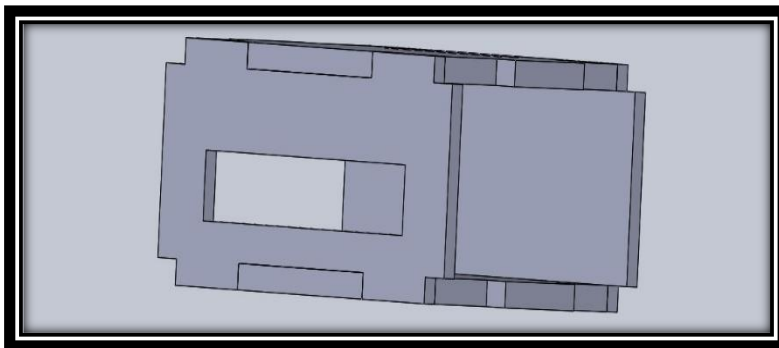


Figura 4.67 Diseño caja de módulo RFID en SolidWorks
Fuente: Israel Cevallos

4.6.1.3. Diseño caja módulo Bluetooth HC-05.

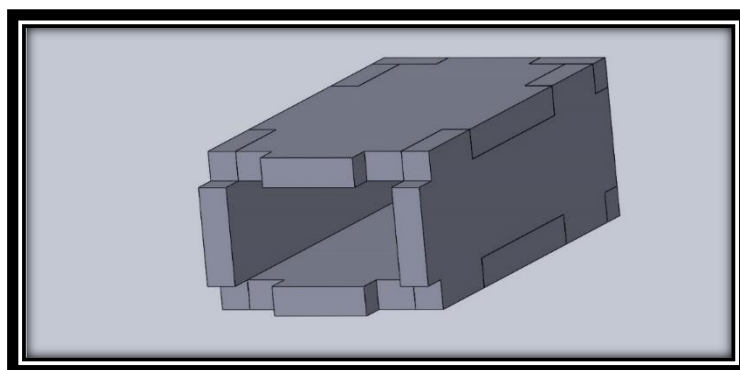


Figura 4.68 Diseño caja de módulo Bluetooth HC-05 en SolidWorks
Fuente: Israel Cevallos

4.6.1.4. Diseño caja módulo relé de 2 canales.

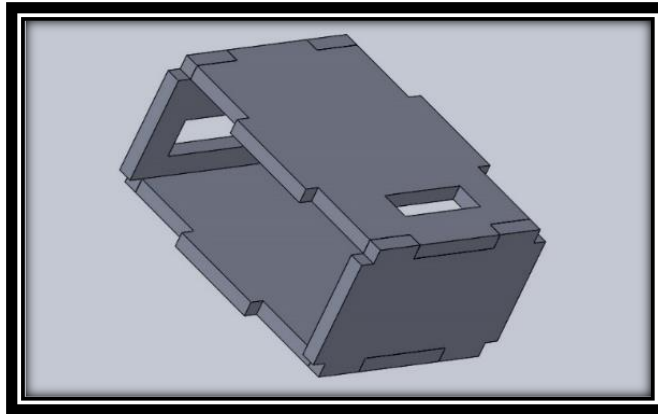


Figura 4.69 Diseño caja de módulo relé de 2 canales
Fuente: Israel Cevallos

4.6.2. Corte Laser.

La máquina de corte laser posee una estructura reforzada, tiene una potencia de aproximadamente 130 w, su tubo de CO2 de vidrio permite el corte preciso en materiales como acrílico, plástico y madera.



Figura 4.70 Corte laser
Fuente: <http://www.maquinadosymetalesserna.com/?opc=VGal&id=50>



Figura 4.71 Máquina de corte laser
Fuente: Israel Cevallos



Figura 4.72 Figuras listas en pantalla para la máquina de corte laser
Fuente: Israel Cevallos

4.6.3. Diseños finales.

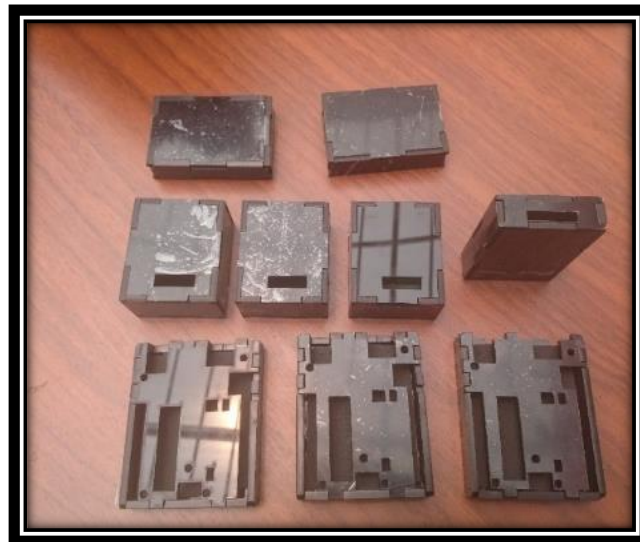


Figura 4.73 Cajas cortadas en laser y ensambladas
Fuente: Israel Cevallos

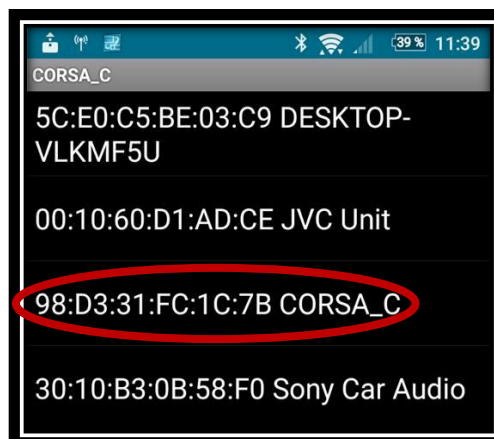
CAPÍTULO 5

5. Pruebas de funcionamiento.

5.1. Pruebas de conexión mediante Bluetooth.

5.1.1. Conexión al Bluetooth

El primer paso es encender el Bluetooth del dispositivo móvil, posteriormente en el ícono de Bluetooth de la App CORSA_C buscará los dispositivos disponibles para establecer una conexión, finalmente se selecciona CORSA_C que es el nombre con el cual está configurado el vehículo.



```
pinMode(bt, OUTPUT);  
// Initialize serial communication at 9600 bits per second:  
  
mySerial.begin(9600);  
mySerial.println("CORSA, conectado");
```

Figura 5.1 Prueba de conexión con Bluetooth CORSA_C

Fuente: Israel Cevallos

5.1.2. Prueba del contacto y apagado del vehículo

5.1.2.1. Vehículo en contacto

El contacto del vehículo se activa oprimiendo el ícono de SMART START, lo que permite es el uso de los accesorios como el radio, aire acondicionado, etc.

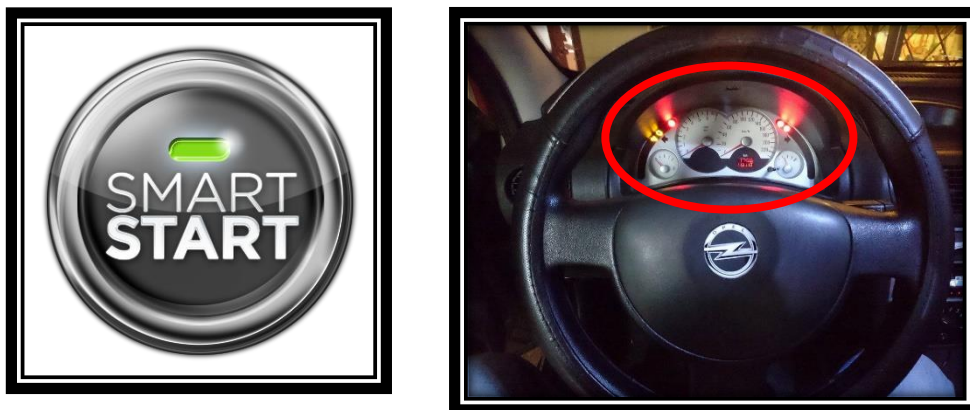


Figura 5.2 Prueba del contacto del vehículo mediante Bluetooth

Fuente: Israel Cevallos

5.1.2.2. Vehículo apagado

En el momento en que el vehículo se encuentre encendido, y se desee apagarlo hay que presionar el ícono de STOP ENGINE.

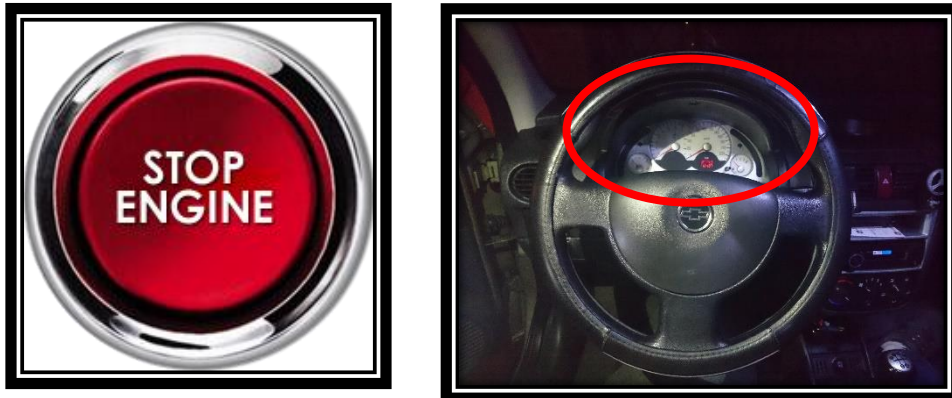


Figura 5.3 Prueba apagado del vehículo mediante Bluetooth

Fuente: Israel Cevallos

5.1.3. Prueba de la apertura y cierre de puertas

5.1.3.1. Apertura de puertas

Para abrir los seguros de las puertas, se oprime ícono de color rojo que indica seguro de puertas abiertas.



Figura 5.4 Prueba de apertura de puertas mediante Bluetooth

Fuente: Israel Cevallos

5.1.3.2. Cierre de puertas

Para cerrar los seguros del Chevrolet Corsa Evolution, se oprime el ícono de color amarillo que indica que el vehículo está asegurado.



Figura 5.5 Prueba de cierre de puertas mediante Bluetooth

Fuente: Israel Cevallos

5.1.4. Prueba de la cajuela

5.1.4.1. Apertura de la cajuela

Para la apertura de la cajuela, se oprime el ícono de color rojo, que permite el desbloqueo del seguro del porta equipaje, de esta manera se alza de manera manual el compartimento trasero.



Figura 5.6 Prueba de apertura de cajuela mediante Bluetooth

Fuente: Israel Cevallos

5.2. Pruebas de conexión mediante NFC/RFID.

5.2.1. Prueba del contacto y apagado del vehículo

El Chevrolet Corsa Evolution, consta con un lector RFID en la parte izquierda del tablero, que permite el contacto del y también el apagado del vehículo con tan solo acercarse el dispositivo móvil con tecnología NFC o a su vez una Tarjeta o Tag RFID.

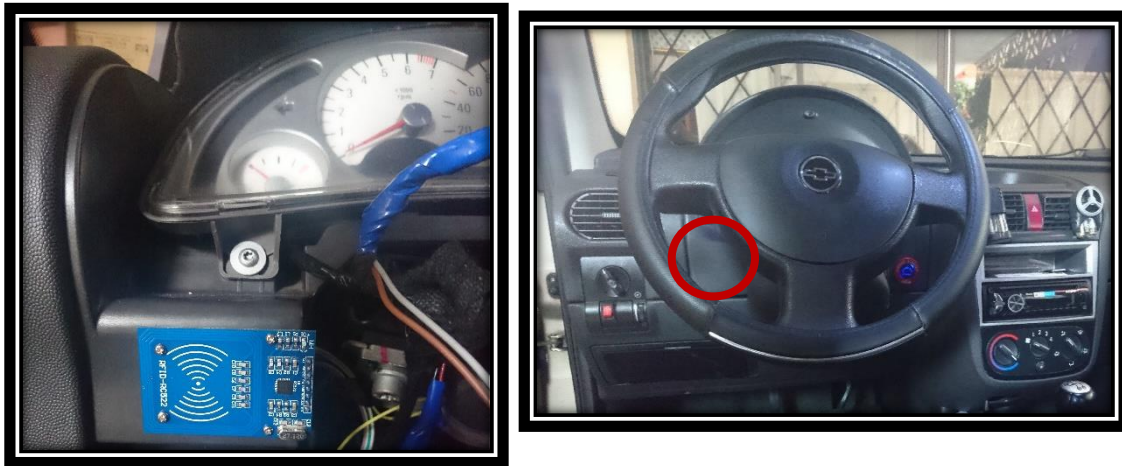
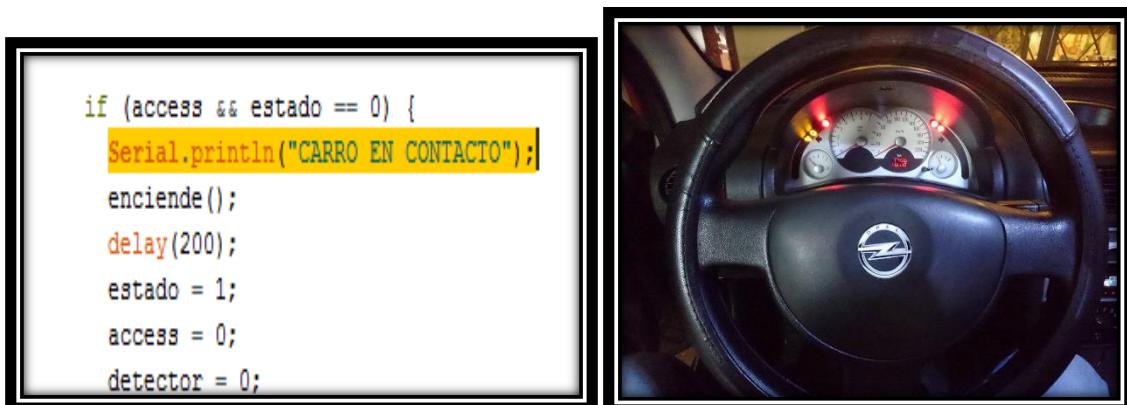


Figura 5.7 Prueba de contacto y apagado del vehículo mediante NFC/RFID

Fuente: Israel Cevallos

5.2.1.1. Vehículo en contacto

El contacto del vehículo está correcto, permite hacer uso de sus accesorios para posteriormente dar arranque por medio del botón de encendido, se puede verificar en el MONITOR SERIE de Arduino el mensaje que muestra, corroborando la funcionalidad del mecanismo.



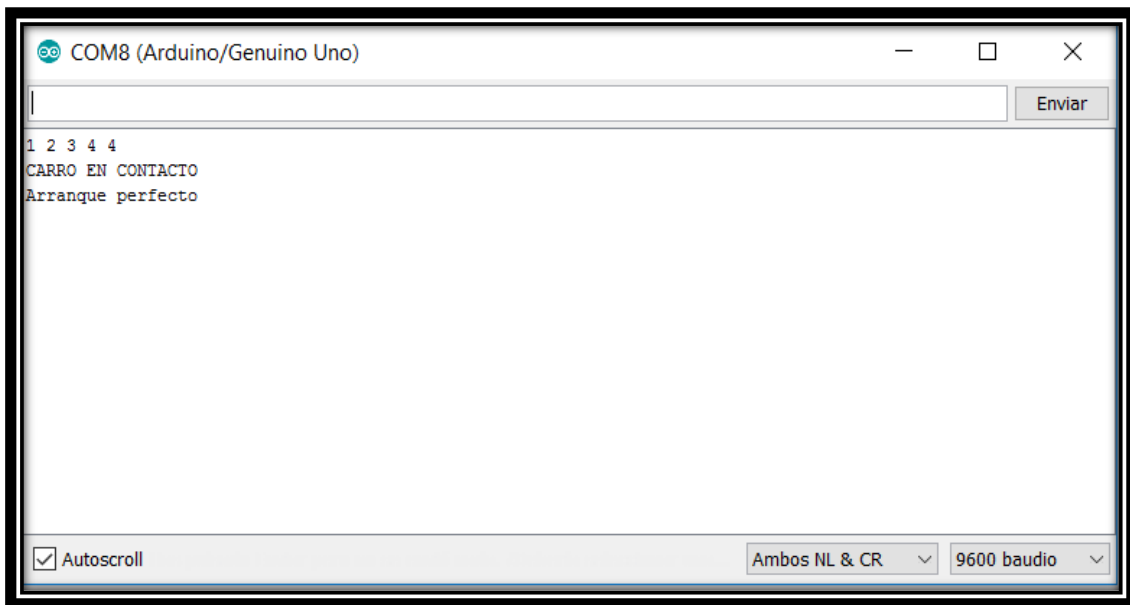


Figura 5.8 Prueba de vehículo en contacto mediante NFC/RFID

Fuente: Israel Cevallos

5.2.1.2. Vehículo apagado

El lector RFID ubicado en la parte izquierda del volante antes descrito, detecta el dispositivo móvil con NFC o su Tarjeta o Tag RFID, antes codificados y permite que el vehículo se apague por completo, y el MONITOR SERIE informa que el carro esta OFF.

```
} else if (access == estado == 1) {  
  Serial.println("APAGADO");  
  apaga();  
  delay(200);  
  estado = 0;  
  access = 0;  
  detector = 0;
```



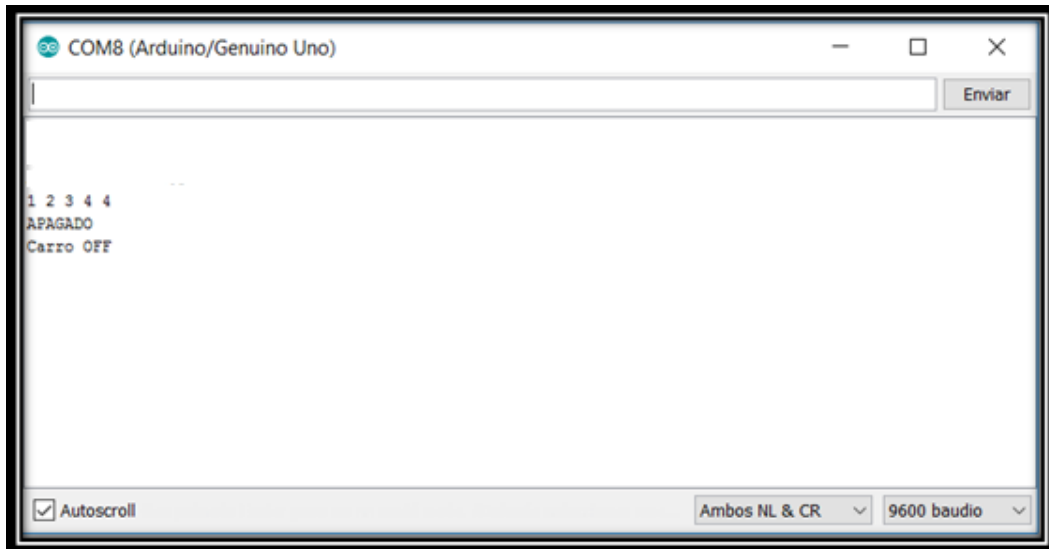


Figura 5.9 Prueba de apagado del vehículo mediante NFC/RFID

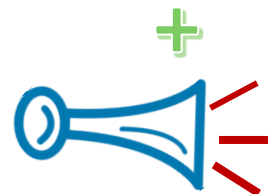
Fuente: Israel Cevallos

5.2.1.3. Intruso

Cuando el lector RFID implementado en el Chevrolet Corsa Evolution, detecta una Tarjeta, Tag o dispositivo móvil no registrado, la bocina del vehículo empieza a sonar, y da una alerta de intruso.

```
mySerial.println("CORSA, USTED ES UN INTRUSO");
Serial.println("CORSA, USTED ES UN INTRUSO");
}

if (detector == 1)
{
  alarma();
}
}
if (detector == 1)
{
  alarma();
}
```





**Figura 5.10 Prueba de intruso en el contacto del vehículo mediante
NFC/RFID**

Fuente: Israel Cevallos

5.2.2. Prueba de apertura y cierre de puertas

El vehículo cuenta con un lector RFID en el extremo superior del parabrisas delantero del lado del conductor, lo que permite la facilidad de acercamiento de la Tarjeta, Tag o dispositivo móvil previamente configurados para que de esta manera permita la apertura y cierre de puertas del Chevrolet Corsa Evolution.



Figura 5.11 Prueba de apertura y cierre de puertas mediante NFC/RFID

Fuente: Israel Cevallos

5.2.2.1. Apertura de puertas

El lector RFID detecta el dispositivo correcto que permita la apertura de puertas, en el MONITOR SERIE de Arduino muestra un mensaje de CORSA, CARRO ABIERTO.

```
if (access == estado == 0) {  
  mySerial.println("CORSA, CARRO ABIERTO");  
  Serial.println("CORSA,CARRO ABIERTO");  
  abre();  
  estado = 1;  
  access = 0;  
  detector = 0;  
}
```

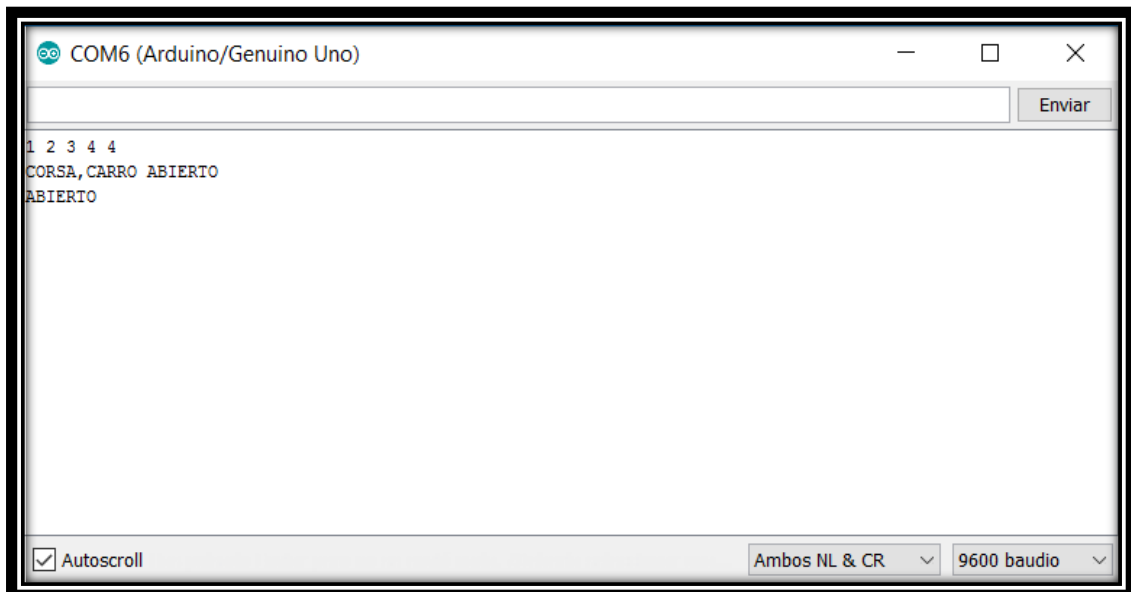


Figura 5.12 Prueba de apertura puertas mediante NFC/RFID

Fuente: Israel Cevallos

5.2.2.2. Cierre de puertas

Con la Tarjeta, Tag y dispositivo móvil NFC se puede asegurar el Chevrolet Corsa Evolution.

```
    } else if (access && estado == 1) {  
        mySerial.println("CORSA, CARRO CERRADO");  
        Serial.println("CORSA, CARRO CERRADO");  
        cierra();  
        estado = 0;  
        access = 0;  
        detector = 0;  
    } else {  
        detector = 1;
```

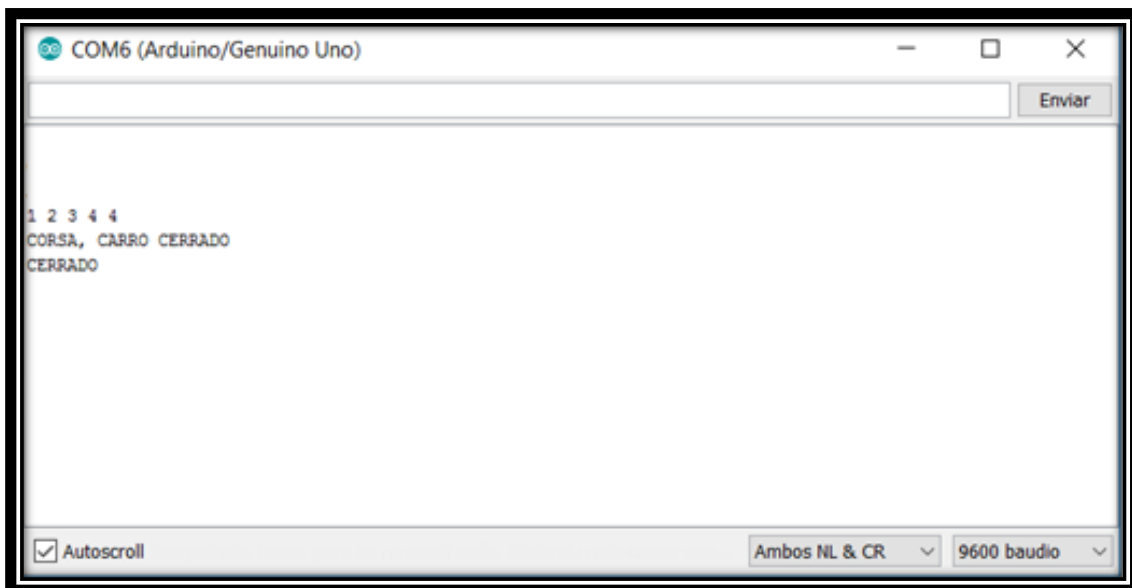


Figura 5.13 Prueba de cierre puertas mediante NFC/RFID

Fuente: Israel Cevallos

5.2.2.3. Intruso

Al momento que se detecte un dispositivo desconocido acercarse al lector RFID ubicado en el parabrisas delantero, la bocina del vehículo comienza a sonar, emitiendo un mensaje de CORSA, USTED ES UN INTRUSO.

```
mySerial.println("CORSA, USTED ES UN INTRUSO");
Serial.println("CORSA, USTED ES UN INTRUSO");
}
if (detector == 1)
{
  alarma();
}
}
if (detector == 1)
{
  alarma();
}
```

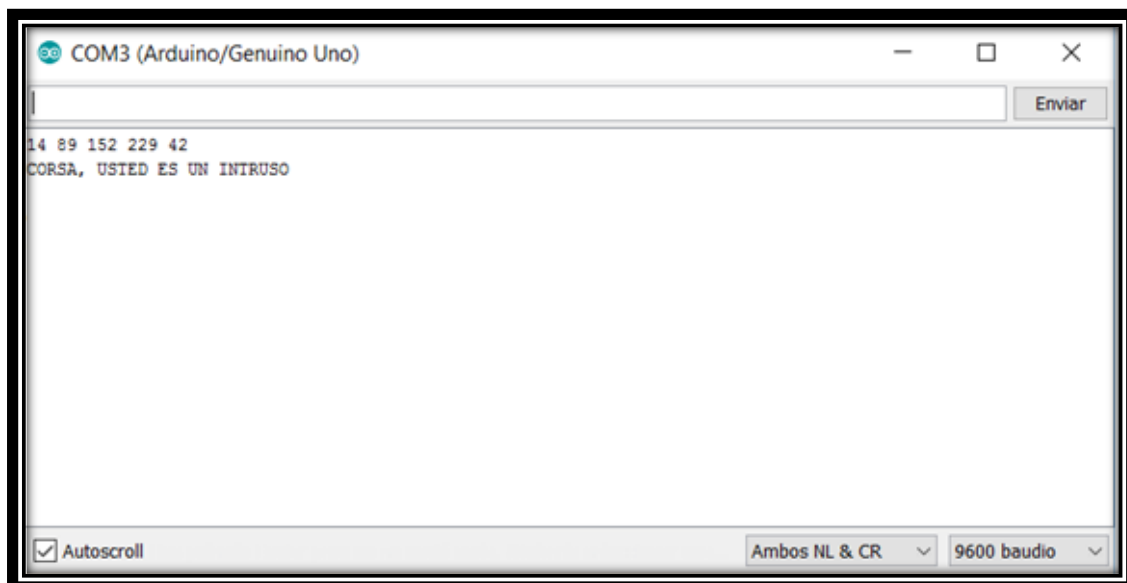


Figura 5.14 Prueba de intruso en puertas mediante NFC/RFID

Fuente: Israel Cevallos

5.2.3. Prueba de la cajuela

Al ser un vehículo Hatchback, el lector RFID está colocado en el parabrisas posterior, lo que permite la maniobrabilidad para acercar el dispositivo vinculado y abrir la cajuela de forma manual.



Figura 5.15 Prueba de la cajuela mediante NFC/RFID

Fuente: Israel Cevallos

5.2.3.1. Cajuela abierta

La cajuela se abre, siempre y cuando el dispositivo de proximidad sea compatible con el lector RFID.

```
if (access == estado == 0) {  
  mySerial.println("CORSA, CAJUELA ABIERTA");  
  Serial.println("CORSA, CAJUELA ABIERTA");  
  abre();  
  estado = 1;  
  access = 0;  
  detector = 0;  
  delay(400);  
  cierra();  
}
```



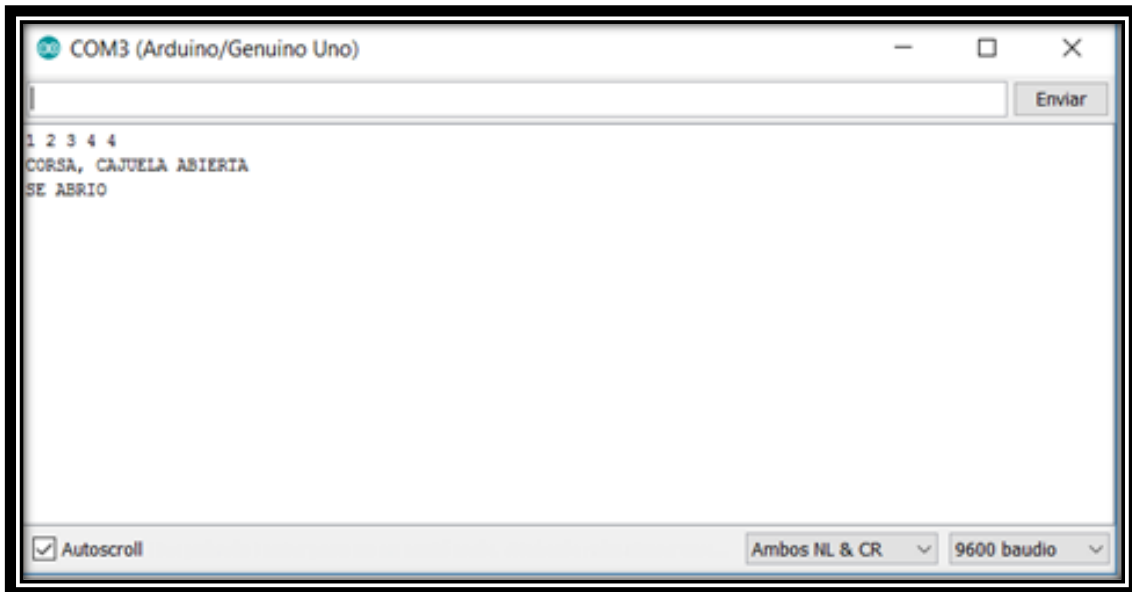


Figura 5.16 Prueba de la cajuela abierta mediante NFC/RFID

Fuente: Israel Cevallos

5.2.3.2. Cajuela cerrada

La cajuela se asegura de manera automática, ya que el actuador instalado regresa por la inercia del mecanismo de la chapa de la cajuela, y al momento emite un mensaje en su MONITOR SERIE de CORSA, CAJUELA CERRADA.

```
    } else if (access == estado == 1) {  
        mySerial.println("CORSA, CAJUELA CERRADA");  
        Serial.println("CORSA, CAJUELA CERRADA");  
        abre();  
        estado = 0;  
        access = 0;  
        detector = 0;  
        delay(400);  
        cierra();  
    } else {  
        detector = 1;
```



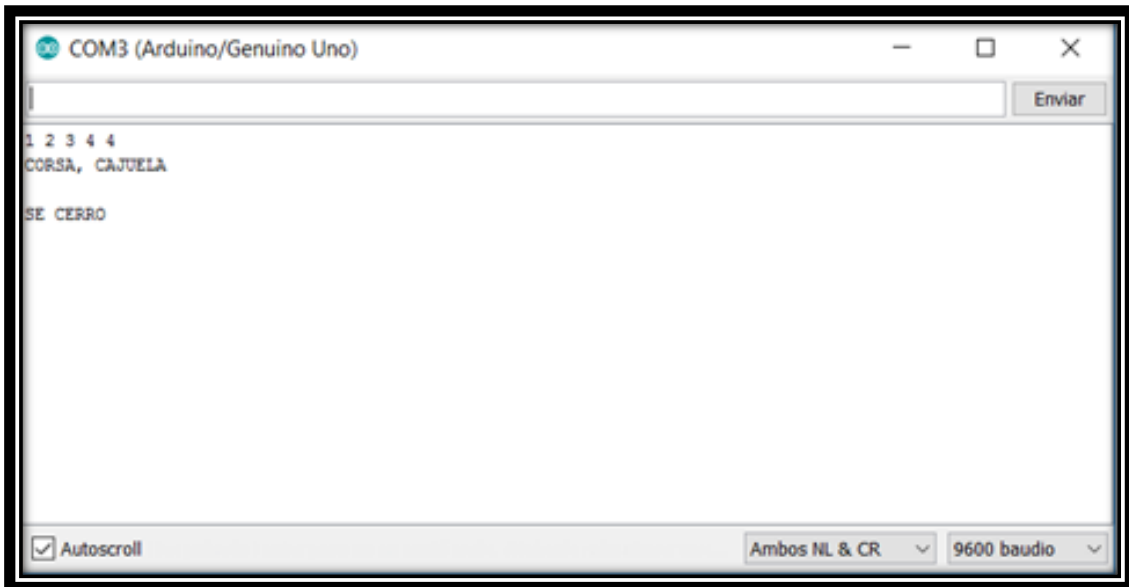


Figura 5.17 Prueba de la cajuela cerrada

Fuente: Israel Cevallos

5.2.3.3. Intruso

La alerta sonora de intruso funciona igual que en los anteriores mecanismos antes descritos, al no detectar un dispositivo codificado con el lector RFID, la bocina del vehículo suena y alerta que un intruso intenta abrir la cajuela.

```
    } else {  
        detector = 1;  
  
        mySerial.println("CORSA, USTED ES UN INTRUSO");  
        Serial.println("CORSA, USTED ES UN INTRUSO");  
    }  
  
    if (detector == 1)  
    {  
        alarma();  
    }  
}  
  
if (detector == 1)  
{  
    alarma();  
}
```



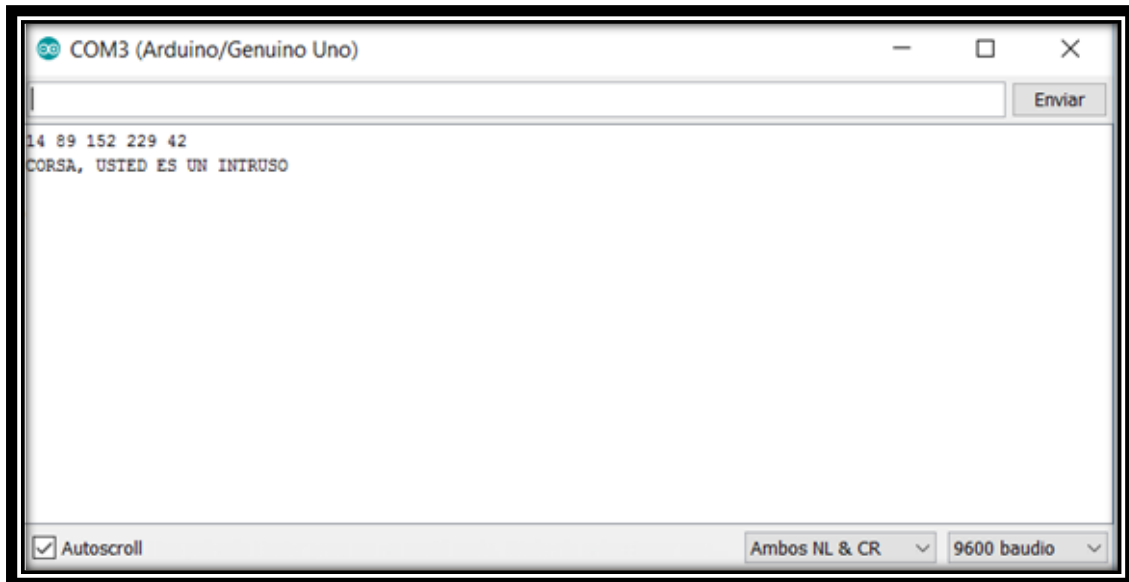


Figura 5.18 Prueba de intruso en la cajuela mediante NFC/RFID

Fuente: Israel Cevallos

CONCLUSIONES

- ✚ Arduino posee mayores beneficios y ventajas respecto a las demás placas programables, no solamente por su costo inferior, sino también, gracias a la multiplataforma en la cual se ejecuta como el 100% de compatibilidad con todos los sistemas operativos que se encuentran disponibles hoy en el mercado, del mismo modo su software de programación está disponible para descargarse gratuitamente desde la Página Web Oficial de Arduino.
- ✚ La correcta combinación de la programación en Arduino con el Código ASCII, permite crear accesos seguros para el modelo implementado de apertura y cierre de puertas, así como el contacto y apagado del Chevrolet Corsa Evolution.
- ✚ NFC es una rama de la tecnología de alta frecuencia de RFID y ambos operan en la misma frecuencia de 13.56MHz, que permite una conexión instantánea para facilitar la aplicación en dispositivos móviles o tarjetas y tags RFID; estos últimos programados para otorgar la misma funcionalidad del dispositivo móvil con tecnología NFC; que facilita la alternabilidad de uso.
- ✚ App inventor es una innovadora herramienta para la creación de aplicaciones móviles que usa la construcción visual y ya no el lenguaje de programación estructurado. Mantiene una interfaz gráfica que facilita la navegación dentro del desarrollo de la misma, para identificar oportunamente el dimensionamiento de los elementos que contiene.

RECOMENDACIONES

- ✚ En la actualidad, la era digital toma cada vez más espacio en aplicaciones para diferentes industrias, por lo que el sector automotriz, debería tomar ventaja de estas iniciativas para entregar a sus clientes productos y servicios que combinen la versatilidad, seguridad y facilidad de uso, como se ha mostrado en este proyecto, a través de la implementación de NFC/RFID y Bluetooth como medio de apertura y cierre de puertas, así como el contacto y apagado del vehículo Chevrolet Corsa Evolution.

- ✚ A fin de que el dispositivo Bluetooth pueda ser asociado únicamente con usuarios específicos, se recomienda sustituir el nombre y contraseña pre-definidos de fábrica, por unos propios, para evitar la vulnerabilidad de emparejamiento con usuarios desconocidos.

- ✚ Con la realización de este proyecto, se recomienda cada vez más la incorporación de llaves digitales que minimizan la vulnerabilidad de llaves convencionales para los vehículos.

- ✚ La era digital nos permite introducirnos en el mundo de la programación y de la diversidad de idiomas, para lo cual se recomienda que los centros de formación incorporen a su curricula este tipo de contenidos que facilita a la población el conocimiento y la aplicación temprana de los mismos.

BIBLIOGRAFÍA

- ✚ Artero, O. T. (2013). ARDUINO curso práctico de formación. En O. T. Artero, *ARDUINO curso práctico de formación* (pág. 3). Madrid: RC Libros.
- ✚ Sabella, R. P. (2016). NFC FOR DUMMIES. En R. P. Sabella, *NFC FOR DUMMIES* (pág. 88). New Jersey: John Wiley & Sons, Inc.
- ✚ Sabella, R. P. (2016). NFC FOR DUMMIES. En R. P. Sabella, *NFC FOR DUMMIES* (pág. 90). New Jersey: John Wiley & Sons, Inc.
- ✚ Boylestad, R. L. (2011). Campo Magnético. En R. L. Boylestad, *INTRODUCCIÓN AL ANÁLISIS DE CIRCUITOS* (pág. 252). México: Pearson.
- ✚ http://www.uco.es/aulasoftwarelibre/wpcontent/uploads/2010/05/Arduino_user_manual_es.pdf
- ✚ Bejarano, C. C. (Diciembre de 2014). Obtenido de <http://docplayer.es/8807230-Desarrollo-de-un-prototipo-rfid-con-alarma-remota-para-monitorear-puertas-de-acceso.html>
- ✚ Cazorla, J. A. (Diciembre de 2015). *Mecánicos*. Obtenido de El blog del taller de reparación del automóvil: <http://www.blogmecanicos.com/2015/12/fallo-en-el-bloqueo-electrico-de-la.html>
- ✚ Dani. (2014). *Aficionados a la Mecánica*. Obtenido de Curso rápido de electricidad del automóvil: http://www.aficionadosalamecanica.net/curso_alternador.htm
- ✚ *ELECTRONICA ESTUDIO.COM*. (s.f.). Obtenido de Ingeniería Electrónica y Proyectos PICmicro: <http://www.electronicaestudio.com/microcontrolador.htm>

- ✚ *FQ Ingeniería Electrónica.* (s.f.). Obtenido de <http://www.fqingenieria.com/es/conocimiento/estandares-y-regularizaciones-para-rfid-36>
- ✚ Marticorena, J. L. (2008). Obtenido de <http://www.profesormolina.com.ar/mismaterias/instalaciones/miniproyectos/minipro2/diodos.pdf>
- ✚ *MASTER MAGAZINE.* (s.f.). Obtenido de <http://www.mastermagazine.info/termino/4020.php>
- ✚ *MCI Capacitación.* (2015). Obtenido de <http://cursos.olimex.cl/rfid-vs-nfc-cuales-la-diferencia/>
- ✚ *MIT App Inventor.* (2015). Obtenido de <http://appinventor.mit.edu/explore/about-us.html>
- ✚ <http://www.forosdeelectronica.com/f24/significa-micocontrolador-sea-8-bits-16-bit-32-a-17930/>
- ✚ <http://electronicaprofacil.blogspot.com/2014/05/arduino-uno-atmega328p-simulacion-con.html>
- ✚ <http://microcontroladores-mrelberni.com/temporizador-contador-pic/>
- ✚ <http://playground.arduino.cc/ArduinoNotebookTraduccion/Structure>
- ✚ <http://es.ccm.net/contents/55-codigo-ascii>
- ✚ <http://docplayer.es/8807230-Desarrollo-de-un-prototipo-rfid-con-alarma-remota-para-monitorear-puertas-de-acceso.html>
- ✚ <http://es.ccm.net/contents/55-codigo-ascii>

ANEXOS

CÓDIGOS DE PROGRAMACIÓN DEL CHEVROLET CORSA EVOLUTION

✚ Anexo en medio magnético (CD)

IMPLEMENTACIÓN EN PROCESO



BOTÓN DE ENCENDIDO SPARCO



ELIMINACIÓN SWITCH DE
ENCENDIDO



IDENTIFICACIÓN DE LOS CABLES
DE IGNICIÓN



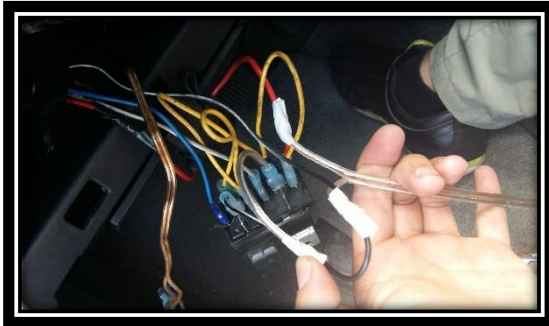
PERFORACIÓN DEL TABLERO DE
INSTRUMENTOS PARA COLOCAR
EL BOTÓN DE ENCENDIDO



INSTALACIÓN Y PRUEBA DE
BOTÓN DE ENCENDIDO



DESARMADO DE PUERTAS



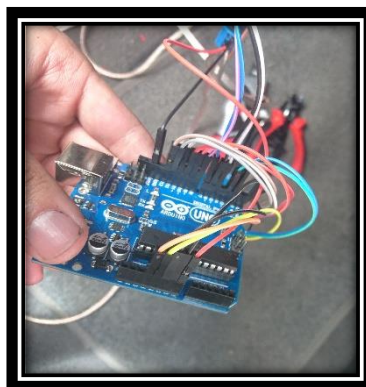
DESCONECCIÓN ALARMA
CHEVROLET



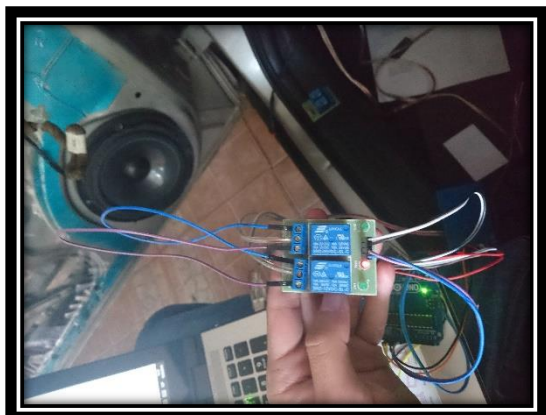
IDENTIFICACIÓN CABLES DE
FUENTE 12 V



PARTES INTERNA DE MÓDULO DE ALARMA CHEVROLET



CONEXIÓN ARDUINO UNO EN PUERTAS



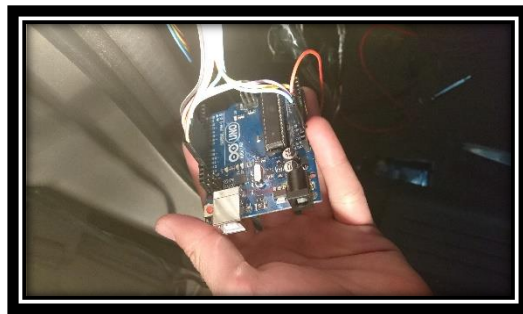
CONEXIÓN MÓDULO Y RELÉ Y ARDUINO UNO EN PUERTAS



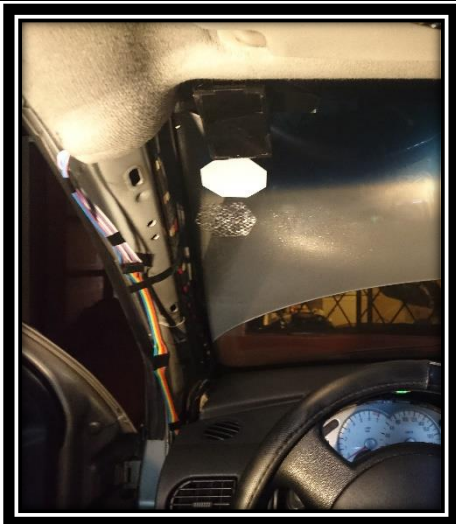
CONEXIÓN MÓDULO RFID EN PUERTAS



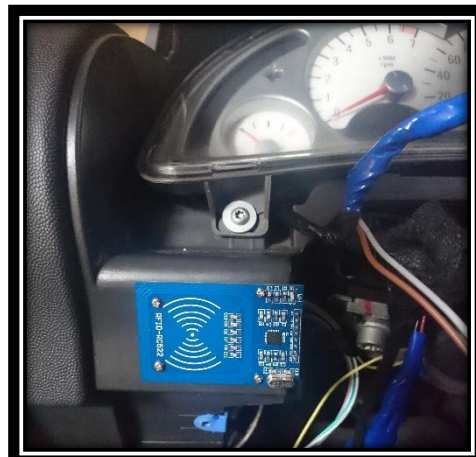
IDENTIFICACIÓN DE LOS CABLES
DE LA BOCINA



CONEXIÓN DE ARDUINO UNO
PARA EL SISTEMA DE CONTACTO
DEL VEHÍCULO



COLOCACIÓN LECTOR RFID EN
PARABRISAS DELANTERO



COLOCACIÓN LECTOR RFID EN
TABLERO DE INSTRUMENTOS



CONEXIÓN MÓDULO BLUETOOTH
HC-05 A ARDUINO UNO DE
PUERTAS



CONEXIÓN ARDUINO UNO EN LA
CAJUELA



CONEXIÓN MÓDULO RFID CON
ARDUINO UNO EN CAJUELA



DESARMADO DE CHAPA DE
CAJUELA



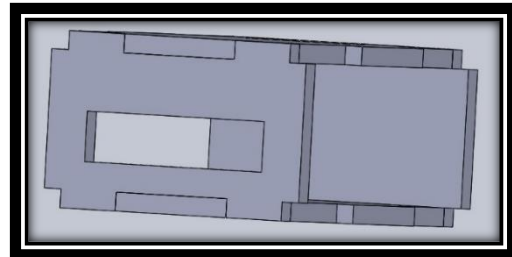
IMPLEMENTACIÓN DE UN
ACTUADOR PARA APERTURA DE
CAJUELA



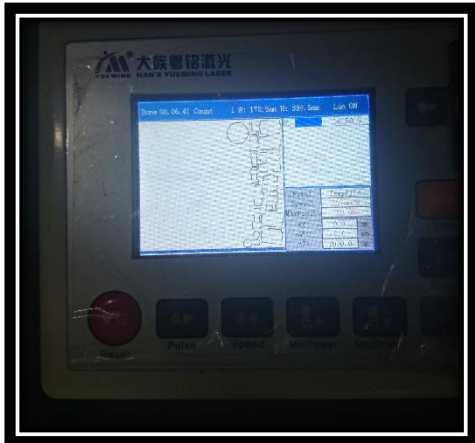
ELIMINACIÓN DE CILINDRO DE
CHAPA EN PUERTAS DELANTERAS



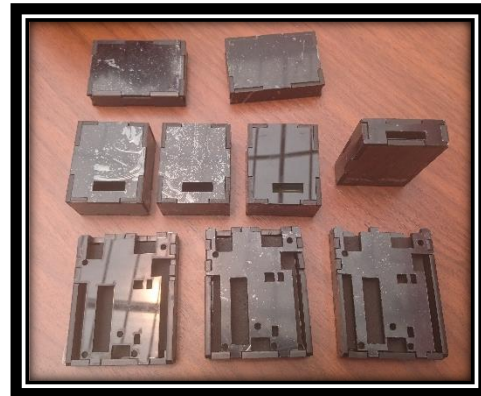
CILINDRO DE CHAPA DE PUERTAS
DELANTERAS



DISEÑO DE CAJAS PARA MÓDULOS
ARDUINO UNO, RFID, RELÉ DE 2
CANALES



CORTE LASER DE CAJAS PARA
MÓDULOS EN ACRÍLICO NEGRO



ENSAMBLAJE DE CAJAS PARA LOS
MÓDULOS EN ACRÍLICO



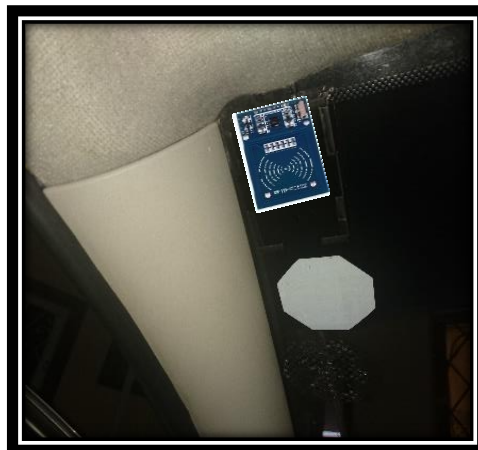
SUJECIÓN DE LAS CAJAS CON SUS
RESPECTIVOS MÓDULOS EN
GABETA DEL VEHÍCULO



PRUEBA DEL CONTACTO DEL
VEHÍCULO



PRUEBA DE FUNCIONAMIENTO
DEL BOTÓN DE ENCENDIDO
SPARCO



PRUEBA DE MÓDULO RFID DE
PUERTAS



PRUEBA DE MÓDULO RFID DE
CAJUELA



CHEVROLET CORSA EVOLUTION
TERMINADO

PRESUPUESTO.

Arduino UNO puertas	\$50
Arduino UNO encendido	\$50
Arduino UNO cajuela	\$50
Módulo relé puertas	\$5,5
Módulo relé encendido	\$5,5
Módulo relé cajuela	\$5,5
Módulo RFID puertas	\$22,5
Módulo RFID encendido	\$22,5
Módulo RFID cajuela	\$22,5
Módulo Bluetooth	\$10
Botón de encendido	\$60
10 m cable #16	\$6
Cables para Arduino	\$10
Automatización cajuela	\$85
Vinchas para puertas	\$10
Corte laser y diseño de cajas	\$70
Fibra de carbono en chapas	\$25
Módulo Bluetooth quemado	\$10
Arduino UNO quemado	\$50
Implementos varios	\$16
Celular	\$500
Publicidad del Vehículo	\$60
Movilización	\$135
Modelado 1	\$550
Ejemplares y suministros de oficina	\$120
TOTAL	\$1951,00