



UNIVERSIDAD UTE

**FACULTAD DE CIENCIAS DE LA INGENIERÍA E
INDUSTRIAS**

CARRERA DE INGENIERÍA AUTOMOTRIZ

**IMPLEMENTACIÓN DE UN SISTEMA DE ENCENDIDO Y
BLOQUEO CENTRAL COMANDADO POR UNA APLICACIÓN
DE TELÉFONO CELULAR**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO AUTOMOTRIZ**

LUIS STEEVEN VELOZ SALAZAR

**DIRECTOR: ING. ELVIS PATRICIO GUALOTUÑA, MSC.
Santo Domingo, Enero 2019**

© Universidad UTE. 2019

Reservados todos los derechos de reproducción

FORMULARIO DE REGISTRO BIBLIOGRÁFICO PROYECTO DE TITULACIÓN

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	171789964-3
APELLIDOS Y NOMBRES:	Veloz Salazar Luis Steeven
DIRECCIÓN:	Coop. Liberación Popular, Ayacucho y Latacunga
EMAIL:	steev_max_veloz@hotmail.es
TELÉFONO FIJO:	---
TELÉFONO MOVIL:	0969993285

DATOS DE LA OBRA	
TITULO:	Implementación de un sistema de encendido y bloqueo central comandado por una aplicación de teléfono celular
AUTOR O AUTORES:	Veloz Salazar Luis Steeven
FECHA DE ENTREGA DEL PROYECTO TÉCNICO:	24 de enero, 2019
DIRECTOR DEL PROYECTO TÉCNICO:	Ing. Elvis Patricio Gualotuña, MsC.
PROGRAMA	PREGRADO <input checked="" type="checkbox"/> POSGRADO <input type="checkbox"/>
TÍTULO POR EL QUE OPTA:	Ingeniero automotriz
RESUMEN:	Dentro del mundo tecnológico, el campo de la programación y automatización crece de manera vertiginosa, los dispositivos electrónicos toman, cada día, más territorio en nuestras vidas, al punto de que casi dependemos de ellos para vivir, véase por ejemplo el uso de los smartphones, computadores personales o smartwatches. Siendo así que la importancia de un sistema de encendido y bloqueo central comandados por una aplicación de teléfono celular es totalmente clara y justificada. Dicho sistema se implementó en un banco de pruebas de un motor diésel tipo DIS con inyección CRDI para el encendido y apagado del motor, y en un conjunto

de cuatro actuadores de un sistema de bloqueo central genérico para el control simultáneo e individual de cada actuador.

En el Marco Teórico se detalló y describió todo lo relacionado con el sistema de encendido de un motor de tipo DIS a Diésel, el sistema de bloqueo central, el ambiente de programación Arduino, y otros componentes electrónicos que permiten el control de estos sistemas, tales como el módulo L298N para control de motores DC y el módulo Bluetooth HC-06 el cual permite establecer una comunicación inalámbrica con el sistema. Partiendo de los conocimientos adquiridos durante el proceso de estudio en horas de clase y la investigación teórica para el desarrollo de este proyecto se realizó un diseño de la estructura de la maqueta tomando en cuenta primeramente el circuito de control y todos sus componentes, de esta manera fue posible diseñar el dimensionado de una cubierta y soporte para este sistema. Una vez culminada la construcción del sistema se desarrolló el programa *.ino* en ARDUINO el cual dirige las acciones a realizar en el funcionamiento del sistema dependiendo de los parámetros y condiciones establecidos en lenguaje C. Finalmente se desarrolló una aplicación para móviles con sistema operativo Android la cual permite al usuario comandar a voluntad el encendido y apagado del MCI como también la apertura y cierre de los actuadores del bloqueo central.

PALABRAS CLAVES:

Innovación, programación,
componentes, Arduino, comunicación
inalámbrica, lenguaje C.

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.



f: _____

VELOZ SALAZAR LUIS STEEVEN
C. I. 171789964-3

DECLARACIÓN Y AUTORIZACIÓN

Yo, **VELOZ SALAZAR LUIS STEEVEN** CI: **171789964-3** autor del proyecto titulado **IMPLEMENTACIÓN DE UN SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL COMANDADO POR UNA APLICACIÓN DE TELÉFONO CELULAR** previo a la obtención del título de **INGENIERO AUTOMOTRIZ** en la Universidad UTE.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación de grado para que sea integrado al Sistema Nacional de información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la BIBLIOTECA de la Universidad UTE a tener una copia del referido trabajo de titulación de grado con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Santo Domingo, 24 de enero de 2019




f: _____

VELOZ SALAZAR LUIS STEEVEN
C. I. 171789964-3

CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor, certifico que el presente trabajo de titulación que lleva por título **IMPLEMENTACIÓN DE UN SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL COMANDADO POR UNA APLICACIÓN DE TELÉFONO CELULAR** para aspirar al título de **INGENIERO AUTOMOTRIZ** fue desarrollado por **LUIS STEEVEN VELOZ SALAZAR**, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería e Industrias; y que dicho trabajo cumple con las condiciones requeridas para ser sometido a las evaluaciones respectivas de acuerdo a la normativa interna de la Universidad UTE.



Ing. Elvis Patricio Gualotuña, MsC.

DIRECTOR DEL TRABAJO

C.I. 171687660-0

DECLARACIÓN

Yo **LUIS STEEVEN VELOZ SALAZAR**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad UTE puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.



Luis Steeven Veloz Salazar
C. I. 171789964-3

DEDICATORIA

Este proyecto lo dedico principalmente a Dios ya que Él es el dador de la vida, la sabiduría y las bendiciones que cada día recibo, a mis padres ya que han sido y serán siempre quienes me enseñen a ser una persona correcta con muy buena moral y principios, son quienes me han dado todo con mucho esfuerzo incondicionalmente y a quienes les debo toda su paciencia y amor.

A mis hermanos y toda mi familia ya que siempre han estado pendientes de mi confiando en que yo lograría terminar mis estudios superiores.

A mi novia, quien siempre me motiva, me apoya y me enseña que los más grandes sueños no se los construye solo.

Luis Steeven Veloz Salazar

AGRADECIMIENTO

Agradezco a Dios primeramente por la vida y la oportunidad que me da de culminar mis estudios universitarios para poder graduarme, cumpliendo así una meta muy importante en mi vida.

A mis padres por ser un muy buen ejemplo de vida, por ser quienes me han guiado por un camino de buenas costumbres y principios éticos, les debo demasiado y espero algún día pagarles todo el sacrificio y amor que me han dado.

A mis hermanos, mi familia, mis amigos y mi novia ya que siempre están conmigo apoyándome y motivándome.

A mis compañeros de universidad por estar estos años conmigo compartiendo conocimientos y aprendiendo juntos.

A mis docentes ya que fueron quienes impartieron sus conocimientos con mucha paciencia, además de que nos permitieron siempre desarrollar una amistad con ellos dentro y fuera de clases.

A mi Director de Tesis Ing. Elvis Gualotuña por dedicar su tiempo para la supervisión, corrección y aporte intelectual durante la dirección de este proyecto.

Luis Steeven Veloz Salazar

ÍNDICE DE CONTENIDO

	Página
RESUMEN	1
1. INTRODUCCIÓN	2
1.1. JUSTIFICACIÓN	2
1.2. MARCO TEÓRICO	3
1.2.1. SISTEMA DE ENCENDIDO	3
1.2.1.1. Sistema de encendido tipo DIS	4
1.2.1.2. Encendido por compresión	4
1.2.2. BLOQUEO CENTRAL O CIERRE CENTRALIZADO	5
1.2.3. ARDUINO	6
1.2.3.1. Introducción a Arduino	6
1.2.3.2. Sistemas empotrados	7
1.2.3.3. Librerías Arduino	7
1.2.3.4. Ejemplo de programa en lenguaje C	8
1.2.4. MOTORES DE CORRIENTE DIRECTA (DC)	10
1.2.5. MÓDULO L298N	11
1.2.6. BLUETOOTH	12
1.2.6.1. Breve historia del Bluetooth	13
1.2.7. CONCEPTOS DE ELECTRÓNICA BÁSICA	14
1.2.7.1. El Transistor y reguladores	14
1.2.8. RELÉS Y ELEMENTOS DE PROTECCIÓN	15
2. METODOLOGÍA	18
2.1. ALIMENTACIÓN	18
2.2. SEÑALES DE RESPUESTA	18
2.3. PROGRAMACIÓN	19
2.4. DISEÑO Y SIMULACIÓN DEL CIRCUITO DE CONTROL	19
2.4.1. FUENTE DE ALIMENTACIÓN	21
2.4.2. CIRCUITO REGULADOR	22
2.4.3. MÓDULO CENTRAL	22
2.4.4. DISPOSITIVO DE COMUNICACIÓN INALÁMBRICA	23
2.4.5. MÓDULOS DE CONTROL DE ACTUADORES	24
2.4.6. CIRCUITOS OPTOACOPLADORES	25

2.5. PROGRAMACIÓN EN ARDUINO	27
2.5.1. DECLARACIÓN DE VARIABLES	27
2.5.2. SECCIÓN SETUP	28
2.5.3. SECCIÓN LOOP	28
2.6. PRUEBAS DE MEDICIÓN EN LA SIMULACIÓN	29
2.7. CONSTRUCCIÓN DEL CIRCUITO DE CONTROL	34
2.7.1. CUBIERTA Y SOPORTE	44
2.8. DISEÑO Y PROGRAMACIÓN DE LA APLICACIÓN DE CELULAR	47
2.8.1. DISEÑO DE LA APLICACIÓN	47
2.8.2. PROGRAMACIÓN DE LA APLICACIÓN	51
2.9. PRUEBAS DE MEDICIÓN EN LA PRÁCTICA	55
2.9.1. VOLTAJES (VOLTÍMETRO)	56
2.9.2. AMPERAJES (AMPERÍMETRO)	58
2.9.3. ONDAS DIGITALES (OSCILOSCOPIO)	59
3. RESULTADOS Y DISCUSIÓN	61
3.1. PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA	61
3.1.1. FUNCIONAMIENTO DE LA APLICACIÓN	62
3.1.2. CONEXIÓN VÍA BLUETOOTH	63
3.1.3. ENCENDIDO Y APAGADO DEL MCI	66
3.1.4. CONTROL DE ACTUADORES	68
4. CONCLUSIONES Y RECOMENDACIONES	69
4.1. CONCLUSIONES	69
4.2. RECOMENDACIONES	70
BIBLIOGRAFÍA	71
ANEXOS	74

ÍNDICE DE TABLAS

	Página
Tabla 1. Librerías del lenguaje C y del sistema Arduino (Reyes Cortés & Cid Monjaraz, 2015)	8
Tabla 2. Dispositivos Bluetooth de acuerdo a la potencia de transmisión (Reyes Cortés & Cid Monjaraz, 2015)	12
Tabla 3. Dispositivos Bluetooth de acuerdo al ancho de banda (Reyes Cortés & Cid Monjaraz, 2015)	13
Tabla 4. Comparación Simulación-Práctica de fuente 12V	56
Tabla 5. Comparación Simulación-Práctica de fuente Arduino	56
Tabla 6. Comparación Simulación-Práctica de señal emitida por Arduino	57
Tabla 7. Comparación Simulación-Práctica de señal de relés	57
Tabla 8. Comparación Simulación-Práctica de señal de módulos L298N	58
Tabla 9. Comparación Simulación-Práctica de amperaje de fuente	58
Tabla 10. Comparación Simulación-Práctica de amperaje de relés	59

ÍNDICE DE FIGURAS

	Página
Figura 1. Diagrama sistema de encendido DIS (Daniel, 2014)	4
Figura 2. Sistema de cierre eléctrico-neumático (CESVIMAP, 2007)	5
Figura 3. Sistema de cierre eléctrico (CESVIMAP, 2007)	6
Figura 4. Cerradura abierta y cerrada (CESVIMAP, 2007)	6
Figura 5. Arduino UNO y cable estándar USB (Amazon, 2016)	7
Figura 6. Funcionamiento de un motor eléctrico (Reyes Cortés & Cid Monjaraz, 2015)	10
Figura 7. Tipos de motores de corriente directa (Reyes Cortés & Cid Monjaraz, 2015)	10
Figura 8. Esquema de un circuito de puente H (Frank, 2017)	11
Figura 9. Módulo L298N (Naylamp Mechatronics SAC., 2016)	12
Figura 10. Esquema de los transistores PNP (Llamas, 2016)	14
Figura 11. Simbología de un transistor (Cinjordiz, Infootec, 2018)	14
Figura 12. Relé de armaduras (Cinjordiz, Infootec, 2018)	15
Figura 13. Partes de un relé de armaduras (Cinjordiz, Infootec, 2018)	15
Figura 14. Conexionado relé de protección (Oñós Prados, Ramírez Vázquez, Ruiz Vasallo, & Pecanins Vallés, 1996)	16
Figura 15. Conexionado relé de mando (Oñós Prados, Ramírez Vázquez, Ruiz Vasallo, & Pecanins Vallés, 1996)	16
Figura 16. Simulación del circuito de control	20
Figura 17. Simulación de la fuente de alimentación	21
Figura 18. Simulación del circuito regulador de voltaje	22
Figura 19. Implementación de la tarjeta Arduino en la simulación	23
Figura 20. Implementación del módulo Bluetooth en la simulación	24
Figura 21. Implementación de los módulos L298N en la simulación	25
Figura 22. Diagrama del switch de encendido (Anónimo, 2016)	26
Figura 23. Simulación de los circuitos optoacopladores	26
Figura 24. Declaración de variables del programa en Arduino	27
Figura 25. Sección Setup del programa en Arduino	28
Figura 26. Parte de la sección Loop del programa en Arduino	29

Figura 27.	Verificación del valor de la fuente de alimentación	30
Figura 28.	Obtención de amperaje emitido por la fuente de alimentación	30
Figura 29.	Verificación del funcionamiento del circuito regulador	31
Figura 30.	Verificación del valor de señal emitido por el módulo Arduino	31
Figura 31.	Verificación del funcionamiento de los circuitos optoacopladores	32
Figura 32.	Obtención de amperaje de los circuitos optoacopladores	32
Figura 33.	Obtención de señal de alimentación de actuadores	33
Figura 34.	Onda de señal digital del pin 12 obtenida en la simulación	33
Figura 35.	Onda de señal digital del pin 13 obtenida en la simulación	34
Figura 36.	Cables flexibles N° 10,12 y 14	35
Figura 37.	Cables de protoboard	35
Figura 38.	Cable común USB	35
Figura 39.	Adaptador USB	36
Figura 40.	Batería de 12 V	36
Figura 41.	Bornes de conexión	37
Figura 42.	Porta fusible y fusible	37
Figura 43.	Botón interruptor	37
Figura 44.	Barra metálica de alimentación común	38
Figura 45.	Regulador LM7805	38
Figura 46.	Circuito regulador de voltaje	38
Figura 47.	Esquema Arduino Leonardo (Aqeel, 2018)	39
Figura 48.	Módulo Bluetooth HC-06	40
Figura 49.	Circuito optoacoplador	40
Figura 50.	Relé 904	41
Figura 51.	Instalación eléctrica del banco de pruebas	41
Figura 52.	Conexión al switch de encendido en ignición y arranque	41
Figura 53.	Soquets macho-hembra de dos pines	42
Figura 54.	Soquets instalados en el banco de pruebas	42
Figura 55.	Motor CRDI del banco de pruebas	42
Figura 56.	Módulo L298N	43
Figura 57.	Bornes de altavoz	43

Figura 58.	Conjunto de actuadores a implementar en el sistema	44
Figura 59.	Alto, ancho y profundidad de la cubierta en AI	45
Figura 60.	Dimensiones internas de las caras C1 y C2 en AI	45
Figura 61.	Dimensiones internas de las caras D1 y D2 en AI	46
Figura 62.	Ancho y largo de las tablas de soporte en AI	47
Figura 63.	Screen "1" de la aplicación de celular	48
Figura 64.	Screen "3" de la aplicación de celular	48
Figura 65.	Screen "4" de la aplicación de celular	49
Figura 66.	Fondo para el icono de la aplicación (Keo, 2018)	49
Figura 67.	Foto de Sportage Diésel 2010 para el ícono (Microsoft, 2010)	50
Figura 68.	Símbolo identificativo de Bluetooth para el ícono (Raya, 2015)	50
Figura 69.	Diseño final del ícono de la aplicación	50
Figura 70.	Bloque de programación Screen "1"	51
Figura 71.	Primera parte del bloque de programación Screen "3"	52
Figura 72.	Segunda parte del bloque de programación Screen "3"	52
Figura 73.	Tercera parte del bloque de programación Screen "3"	53
Figura 74.	Cuarta parte del bloque de programación Screen "3"	54
Figura 75.	Bloque de programación Screen "4"	54
Figura 76.	Multímetro automotriz utilizado en las pruebas de medición	55
Figura 77.	Osciloscopio automotriz utilizado en las pruebas de medición	55
Figura 78.	Comparación Simulación-Práctica de fuente 12V	56
Figura 79.	Comparación Simulación-Práctica de fuente Arduino	56
Figura 80.	Comparación Simulación-Práctica de señal emitida por Arduino	57
Figura 81.	Comparación Simulación-Práctica de señal de relés	57
Figura 82.	Comparación Simulación-Práctica de señal de módulos L298N	58
Figura 83.	Comparación Simulación-Práctica de amperaje de fuente	58
Figura 84.	Comparación Simulación-Práctica de amperaje de relés	59

Figura 85.	Comparación Simulación-Práctica onda de señal digital pin 13	59
Figura 86.	Comparación Simulación-Práctica onda de señal digital pin 12	60
Figura 87.	Conexión del sistema a la fuente 12V para las pruebas	61
Figura 88.	Conexión del sistema al banco de pruebas	62
Figura 89.	Conexión entre circuito de control y banco de pruebas	62
Figura 90.	Funcionamiento de la aplicación en tres modelos de celulares	63
Figura 91.	Enlazando móvil a dispositivo Bluetooth	63
Figura 92.	Ícono identificativo de la aplicación	64
Figura 93.	Pantalla de inicio de la aplicación	64
Figura 94.	Mensaje de advertencia de comunicación Bluetooth	65
Figura 95.	Enlazando aplicación a dispositivo Bluetooth	65
Figura 96.	Conexión establecida entre aplicación y dispositivo Bluetooth	66
Figura 97.	Encendido del banco de pruebas por medio de la aplicación	66
Figura 98.	Encendido del MCI por medio de la aplicación	67
Figura 99.	Apagado del MCI por medio de la aplicación	67
Figura 100.	Control individual de actuadores por medio de la aplicación	68
Figura 101.	Control simultáneo de actuadores por medio de la aplicación	68

ÍNDICE DE ANEXOS

	Página
ANEXO 1. GUÍA DE PRÁCTICA PARA LA CONEXIÓN DEL SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH AL BANCO DE PRUEBAS	74
ANEXO 2. GUÍA DE PRÁCTICA PARA LA PROGRAMACIÓN DEL SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH	91
ANEXO 3. BOCETO INICIAL DE LA MAQUETA	105
ANEXO 4. DISEÑO Y CONTRUCCIÓN DE LA CUBIERTA EN MADERA	105
ANEXO 5. DISTRIBUCIÓN DE LOS DISPOSITIVOS PARA TOMA DE MEDIDAS	106
ANEXO 6. CONSTRUCCIÓN DE LA CUBIERTA DE ACRÍLICO	106
ANEXO 7. PERFORADO EN LA CUBIERTA DE ACRÍLICO	107
ANEXO 8. MONTAJE DE LOS DISPOSITIVOS EN LA CUBIERTA	107
ANEXO 9. CONEXIÓN Y PRUEBA DE FUNCIONAMIENTO EN EL BANCO DE PRUEBAS	108
ANEXO 10. INSTALACIÓN DE SOQUET EN EL BANCO DE PRUEBAS	108
ANEXO 11. SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH TERMINADO	109

RESUMEN

Dentro del mundo tecnológico, el campo de la programación y automatización crece de manera vertiginosa, los dispositivos electrónicos toman, cada día, más territorio en nuestras vidas, al punto de que casi dependemos de ellos para vivir, véase por ejemplo el uso de los smartphones, computadores personales o smartwatches. Siendo así que la importancia de un sistema de encendido y bloqueo central comandados por una aplicación de teléfono celular es totalmente clara y justificada. Dicho sistema se implementó en un banco de pruebas de un motor diésel tipo DIS con inyección CRDI para el encendido y apagado del motor, y en un conjunto de cuatro actuadores de un sistema de bloqueo central genérico para el control simultáneo e individual de estos.

En el Marco Teórico se detalló y describió todo lo relacionado con el sistema de encendido de un motor de tipo DIS a Diésel, el sistema de bloqueo central, el ambiente de programación Arduino, y otros componentes electrónicos que permiten el control de estos sistemas, tales como el módulo L298N para control de motores DC y el módulo Bluetooth HC-06 el cual permite establecer una comunicación inalámbrica con el sistema. Partiendo de los conocimientos adquiridos durante el proceso de estudio en horas de clase y la investigación teórica para el desarrollo de este proyecto se realizó un diseño de la estructura de la maqueta tomando en cuenta primeramente el circuito de control y todos sus componentes, para lo cual se realizaron pruebas de funcionamiento en PROTEUS, de esta manera fue posible diseñar el dimensionado de una cubierta y soporte para este sistema utilizando el software de diseño en 2D ADOBE ILLUSTRATOR. Una vez culminada la construcción del sistema se desarrolló el programa *.ino* en ARDUINO el cual dirige las acciones a realizar en el funcionamiento del sistema dependiendo de los parámetros y condiciones establecidos en lenguaje C. Finalmente se desarrolló una aplicación en el software online APP INVENTOR para móviles con sistema operativo Android la cual permite al usuario comandar a voluntad el encendido y apagado del MCI como también la apertura y cierre de los actuadores del bloqueo central.

En Resultados y Discusión se demostró el correcto funcionamiento del sistema implementado mediante las pruebas realizadas, para lo cual se hicieron mediciones de voltajes, corriente y obtención de ondas digitales emitidas por el módulo Arduino en una simulación en Proteus y una práctica real.

Palabras claves: Innovación, programación, componentes, Arduino, comunicación inalámbrica, lenguaje C.

1. INTRODUCCIÓN

1.1. JUSTIFICACIÓN

La innovación en el ámbito tecnológico del sector automotriz está tomando mucha fuerza en los últimos años, véase por ejemplo los sistemas electrónicos cada vez más completos y sofisticados en los vehículos con múltiples funciones adecuados para reducir las emisiones contaminantes, la seguridad y el confort de los pasajeros, de manera que la realización de un proyecto enfocado en este tema tiene muy buena aceptación, por lo que un sistema de encendido en un MCI y bloqueo central comandados por una aplicación de teléfono celular con sus respectivas guías de montaje, guías de conexión y guías de programación complementarían los conocimientos teóricos recibidos en asignaturas de carácter técnico mediante prácticas en las áreas de talleres de la Universidad Tecnológica Equinoccial, buscando el beneficio y la protección del usuario o propietario de un vehículo automotor demostrando la credibilidad de este sistema para así ganar la confianza de un posible cliente, además de fomentar y promover la investigación permitiendo fortalecer la experiencia en el manejo de éste tipo sistemas innovadores inclinados hacia las nuevas tecnologías para los estudiantes de la carrera de ingeniería automotriz. (Artero, 2013) (Bosch, 2000)

Mediante la investigación se realiza y se pone en marcha un sistema de encendido y bloqueo central comandado por un teléfono móvil dentro de las instalaciones de la institución, de manera que la implementación de un sistema de este tipo no solo permite un valor agregado al sistema de control de un vehículo moderno, sino que ofrece la oportunidad de ser estudiado mediante guías de práctica enfocadas en la conexión del circuito y la programación del módulo principal del sistema, ya que éste cuenta con un módulo Arduino siendo el módulo principal para el funcionamiento de este proyecto, el cual es estudiado tanto externa (conexiones) como internamente (lenguajes de programación). (Banzi & Shiloh, 2014)

En medio de una competencia en el campo de la innovación tan fuerte dentro del mercado, un proyecto innovador de este tipo se ve en la necesidad de proponer una idea que beneficie a los propietarios y futuros propietarios de un vehículo moderno, brindando un sistema de encendido y apertura de puertas mediante la manipulación por celular, dispositivo electrónico del cual las personas del nuevo siglo dependen cada vez más, protegiendo tanto el vehículo como el sentido de pertenencia y confianza del propietario. De esta manera se logra que el cuidado y la protección del vehículo estén al alcance de sus dueños, mediante la manipulación por teléfono celular, volviéndose más sofisticado y personal. (Bosch, 2000)

El impacto que genera este proyecto en la sociedad se puede apreciar en la ayuda y facilidad para los estudiantes de la carrera de ingeniería automotriz

al brindarles un equipo de pruebas de conexión y programación basado en el sistema Arduino ya que no van a tener inconvenientes ni impedimentos al estudiar y poner en práctica temas vistos solo de manera teórica, esto es algo muy positivo para la sociedad ya que los estudiantes de dicha carrera complementan sus conocimientos en muchos más campos de la electrónica y programación para así ofrecer un mejor servicio en el ámbito laboral.

El impacto teórico que genera este proyecto beneficia y complementa mucho los conocimientos adquiridos en clase por parte de los estudiantes de ingeniería automotriz ya que les permite aplicar lo aprendido mediante la práctica y ensayo.

Este proyecto permite estudiar el ambiente de programación Arduino enfocado en la interacción con otros módulos de control electrónicos, así como también el funcionamiento del sistema en tiempo real mediante las señales de entrada y salidas (voltaje, amperaje, curvas digitales). Este sistema es de uso didáctico para los estudiantes de la carrera de ingeniería automotriz ya que cuenta con guías de conexión y programación reforzando así temas referentes a la programación y electricidad automotriz.

1.2. MARCO TEÓRICO

En este capítulo veremos los conceptos básicos referentes a todo lo empleado en la realización de este proyecto de tesis.

1.2.1. SISTEMA DE ENCENDIDO

Este sistema tiene como fin otorgar la energía necesaria al motor de manera periódica para mantener los ciclos de funcionamiento (admisión, compresión, combustión, escape). El encendido del combustible se producirá después de haberse realizado las correspondientes fases, véase el caso del motor de combustión interna ciclo Otto, el cual genera una chispa en la cámara de combustión o el de ciclo Diésel, que a su vez envía combustible mediante las bombas de inyección. El sistema de encendido se ha clasificado en los siguientes tipos:

- Encendido convencional (ruptor)
- Encendido electrónico por descarga de condensador
- Encendido electrónico sin contactos (transistorizado)
- Encendido electrónico integral
- Sistema de encendido DIS (Direct Ignition System)

En este caso nos enfocaremos en el funcionamiento del sistema de encendido tipo DIS.

1.2.1.1. Sistema de encendido tipo DIS

Este tipo de sistema no depende del distribuidor como en la mayoría de los casos para lo cual se envía la chispa a dos de los cilindros, si no que basa su funcionamiento en la recepción de datos y condiciones enviadas por la computadora mediante sensores (posición, temperatura, revoluciones) para saber en qué momento lanzar la chispa. Este sistema suprime ciertos elementos mecánicos los cuales sufrirían daños y averías.

En el sistema de encendido tipo DIS se gana tiempo potencia y fiabilidad al momento de generar la chispa lo cual previene inconvenientes a altas revoluciones, debido al acercamiento de las bobinas a las bujías suprimiendo en algunos casos los cables de alta tensión.

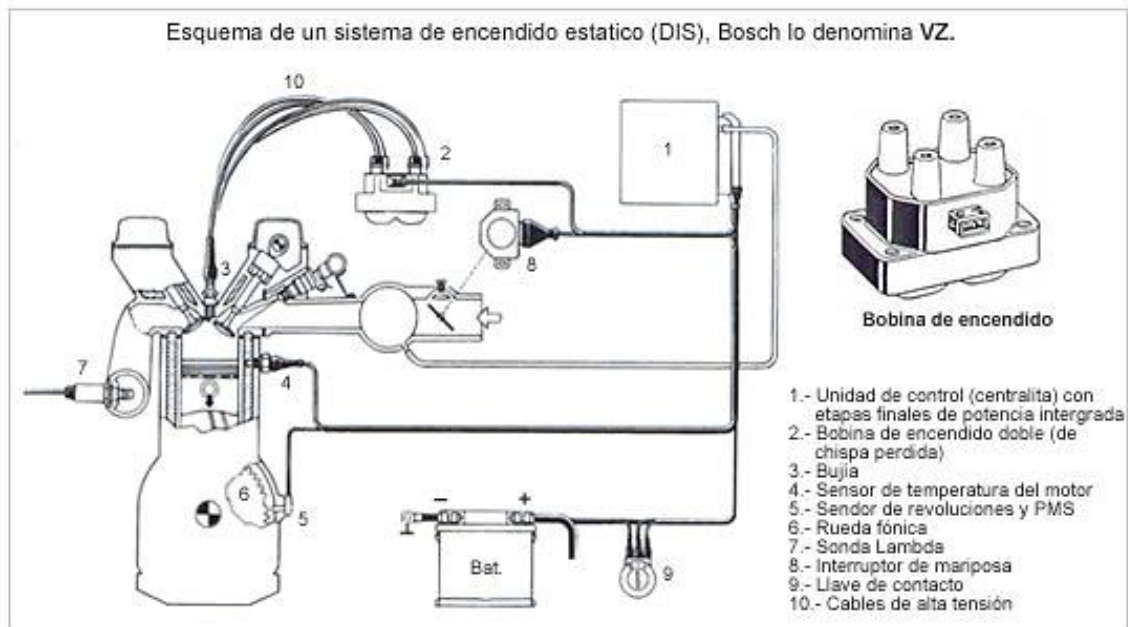


Figura 1. Diagrama sistema de encendido DIS (Daniel, 2014)

1.2.1.2. Encendido por compresión

Los motores cuyo sistema de encendido es por compresión tienen tres fases:

1-2: Retardo de encendido (se produce calor por medio del combustible pero no lo suficiente para que la presión aumente).

2-3: Aumento fuerte de presión (el combustible inyectado combustiona rápidamente).

3-4: Combustión lenta del combustible no quemado.

El rendimiento máximo aún no puede lograrse a pesar de completarse estas tres fases ya que la mezcla aire-combustible no se ha realizado completamente.

1.2.2. BLOQUEO CENTRAL O CIERRE CENTRALIZADO

En la actualidad la gran mayoría de los vehículos han adoptado la evolución del bloqueo central, cuyo principio de funcionamiento se enfoca en la apertura y el cierre de las puertas de un vehículo simultáneamente, proporcionando comodidad al evitar la apertura y cierre de todas las puertas de manera individual.

Este sistema además se enfoca en brindar seguridad, evitando que alguna puerta quede abierta por error o descuido. Generalmente no es necesario el contacto entre las llaves y el bombín para que se lleve a cabo esta operación, se lo hace a distancia.

Cuando el sistema actúa, un sensor se encarga de enviar una señal hacia la centralita la cual la distribuye a cada uno de los actuadores de las cerraduras para así bloquearlas o desbloquearlas a la vez.

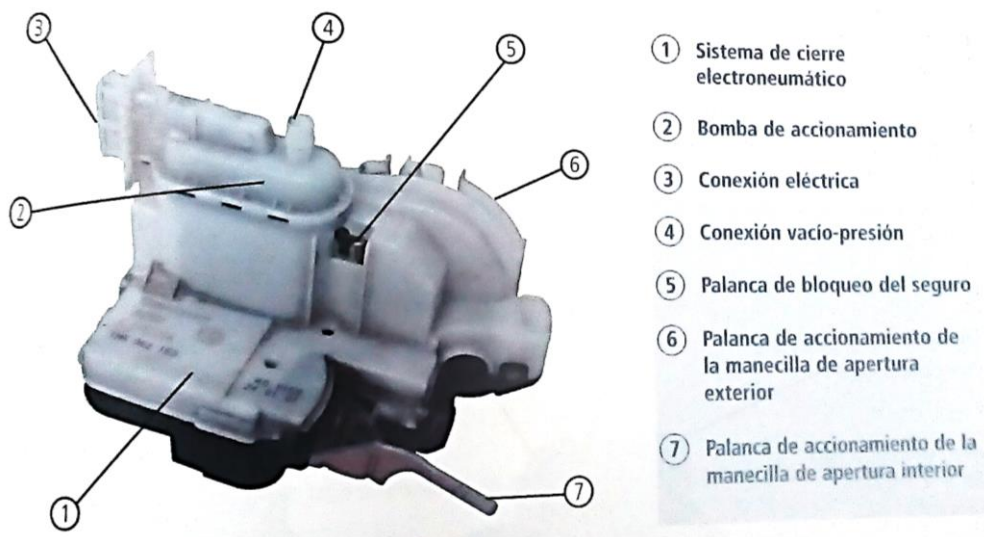


Figura 2. Sistema de cierre eléctrico-neumático (CESVIMAP, 2007)

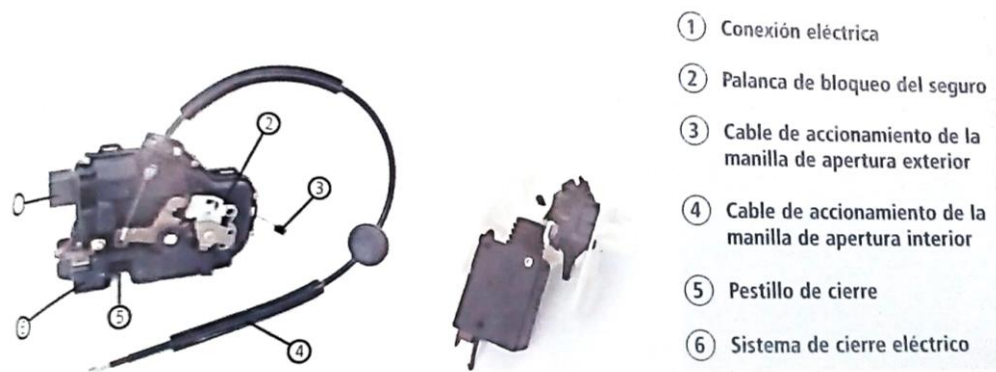


Figura 3. Sistema de cierre eléctrico (CESVIMAP, 2007)

El sistema de bloqueo central se compone principalmente del botón de seguridad y los motores o actuadores.

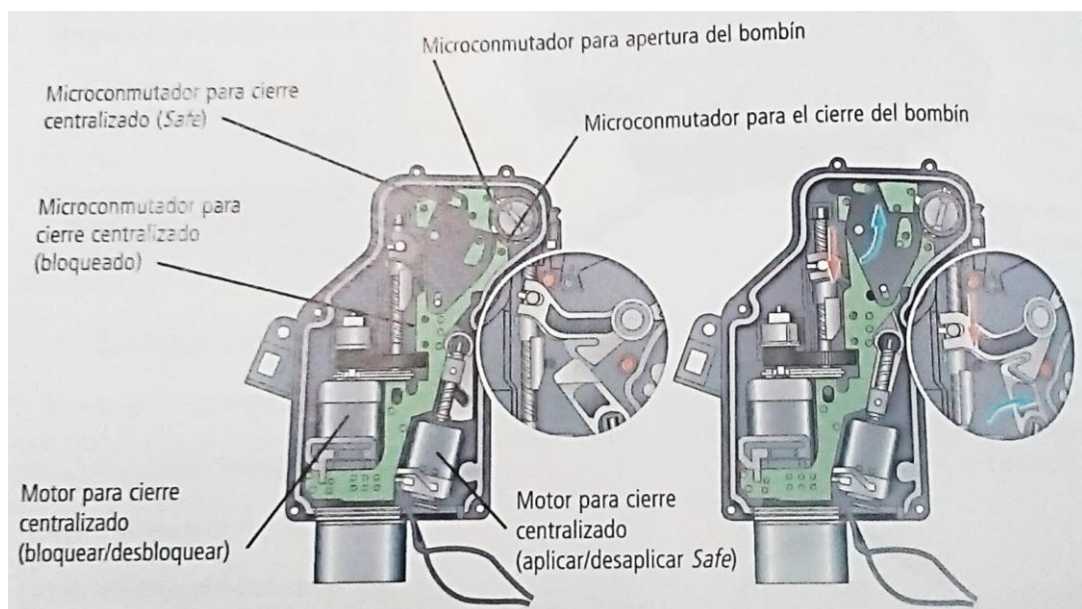


Figura 4. Cerradura abierta y cerrada (CESVIMAP, 2007)

1.2.3. ARDUINO

1.2.3.1. Introducción a Arduino

En breve resumen, *Arduino* es una plataforma electrónica útil para la programación a manera de arquitectura abierta la cual cuenta con una extensa gama de aplicaciones en ingeniería y ciencias exactas. Este sistema se lo ha identificado con el símbolo matemático de infinito, con los signos mas (+) y menos (-) en su interior. El sistema Arduino basa su principio de funcionamiento mediante la familia de microcontroladores *ATMEL* y el lenguaje de programación C y C++ los cuales son herramientas que permiten el modelado, procesado y análisis de programación que

representan un sistema empotrado permitiendo automatizar procesos físicos. Una de las grandes ventajas de este sistema es que su software es gratuito el cual puede obtenerse desde la página web principal de Arduino, sus tarjetas o módulos y diversos dispositivos pueden adquirirse como sistemas de desarrollo a precios muy accesibles, los cuales también pueden ensamblarse a mano. Además el sistema mantiene compatibilidad con casi todos los sistemas operativos como Windows, Linux y Macintosh OSX. La empresa bajo el nombre de Creative Commons se encarga de publicar los diagramas esquemáticos y módulos de componentes bajo su licencia, siendo realizadas versiones mejoradas, ampliadas y optimizadas de los módulos por diseñadores de sistemas electrónicos.



Figura 5. Arduino UNO y cable estándar USB (Amazon, 2016)

1.2.3.2. Sistemas empotrados

Este tipo de plataformas electrónicas cuyo sistema de funcionamiento se basa en el de un microcontrolador mediante un lenguaje programable indicado y una sección de periféricos ya sean timers, puertos I/O, contadores, FPGA's, PMW, y una instrumentación tipo electrónica tales como convertidores analógicos/digital y digital/ analógico para el respetivo acoplamiento de señales y comandos de servos, robots, sensores, etc. Estos sistemas se dedican en tiempo real al procesamiento, control y a la automatización de procesos y eventos físicos.

1.2.3.3. Librerías Arduino

Dentro de las librerías estándar del lenguaje C existe un subconjunto de librerías avr-libc, las cuales contienen funciones específicas que permiten utilizar los recursos de los microcontroladores RISC ATMEL AVR, por lo que

brindan la forma de operar o trabajar para realizar un amplio espectro en cuanto a aplicaciones dentro del ámbito de las ciencias exactas e ingeniería. A continuación se detallan algunas de las librerías del sistema Arduino y del lenguaje C.

Tabla 1. Librerías del lenguaje C y del sistema Arduino (Reyes Cortés & Cid Monjaraz, 2015)

Librería	Descripción
<code>#include <alloccate.h></code>	Asignación temporal de memoria stack.
<code>#include <math.h></code>	Funciones matemáticas.
<code>#include <stdlib.h></code>	Librerías estándar.
<code>#include <string.h></code>	Manejo y procesamiento de cadenas.
<code>#include <stdint.h></code>	Tipos estándar de enteros.
<code>#include <avr/boot.h></code>	Soporte y utilidades del arrancador/cargador (bootloader).
<code>#include <avr/cpufunc.h></code>	Funciones especiales del microcontrolador AVR.
<code>#include <avr/eeprom.h></code>	Manejo e memoria EEPROM.
<code>#include <avr/interrupt.h></code>	Manejo de interrupciones.
<code>#include <avr/io.h></code>	Manejo y definiciones de dispositivos entrada/salida.
<code>#include <Ethernet.h></code>	Librerías para comunicación por medio de Ethernet.
<code>#include <ArduinoRobot.h></code>	Librerías para el control de la tarjeta Robot.
<code>#include <GSM.h></code>	Conexión a red GSM/GRPS con GSM shield.
<code>#include <LiquidCrystal.h></code>	Control del exhibidor de cristal líquido.
<code>#include <SD.h></code>	Lectura y escritura de tarjetas SD.
<code>#include <Servo.h></code>	Funciones para manejo y control de servomotores.
<code>#include <SPI.h></code>	Comunicación con dispositivos usando el bus de Interface serial periférica (SPI).
<code>#include <Stepper.h></code>	Control de motores a paso.
<code>#include <WiFi.h></code>	Comunicación por medio de Internet usando Arduino WiFi shield.
<code>#include <Wire.h></code>	Comunicación para transmitir/recibir datos por medio del protocolo TWI/I2C.

1.2.3.4. Ejemplo de programa en lenguaje C

Para entender este tipo de lenguaje programable es necesario realizar programas partiendo desde los más sencillos, en este caso veremos la aplicación e importancia de comandos básicos que pueden ser utilizados en múltiples combinaciones para realizar programas más complejos y completos.

Crearemos un programa que imprima los mensajes “hola a todos”. Desarrollaremos la habilidad de cargar un programa que cree textos, compilarlo, cargarlo, ejecutarlo con éxito y saber dónde será la salida del mismo. Comprendidos estos proceso lo demás es sumamente fácil. En lenguaje C, el programa para imprimir el mensaje “hola a todos” será:


```
#include <stdio.h>

main( )
{
    printf("hola a todos\n");
}
```

Se ejecutará este programa según el sistema que se esté empleando. Por ejemplo, para el sistema operativo UNIX se debe agregar la extensión *.c*, como "hola.c", para luego ser compilado. Si al compilarse no se detectan fallas o errores, como escribir algo incorrectamente u omitir algún carácter, la compilación se realizará sin mostrar ningún mensaje y se creará un archivo ejecutable con nombre *a.out*. Si se ejecuta *a.out* al escribir la orden

```
a.out
```

se escribirá entonces

```
hola a todos
```

Un programa C se conforma de *funciones* y *variable*, donde las funciones contienen proposiciones las cuales especifican las operaciones para los cálculos que se realizarán, mientras que las variables se encargan de almacenar información o valores a utilizar en los cálculos.

Este ejemplo contiene una función llamada "main" (se puede asignar cualquier nombre), pero "main" es imprescindible en este lenguaje lo que quiere decir que todo programa en este lenguaje debe tener "main" en algún sitio. Generalmente "main" llama a otras funciones para que le ayuden a realizar su trabajo, algunas que el programador, usted o yo escribimos, y otras más de bibliotecas antes incluidas.

La primera línea de este programa (*#include <stdio.h>*) nos indica que se debe incluir información referente a la biblioteca estándar de entrada/salida; esta línea tiende a aparecer en muchos de los archivos fuente de C.

Para establecer una comunicación entre las funciones la función que llama debe proporcionar una lista de valores a la función que invoca denominados *argumentos*. La lista de argumentos son encerrados por los paréntesis ubicados después del nombre de la función. En este ejemplo se define a "main" para que no espere ningún argumento, por ello es indicado con () que la lista está vacía.

`#include <stdio.h>`

Se incluye información de la librería estándar.

`main ()`

Se define la función main la cual no recibe valores de argumentos.

```
{  
  printf( "hola a todos\n");  
}
```

Se encierran con llaves las proposiciones de main.
main hace un llamado a la función librería printf para escribir esta secuencia de caracteres.
El carácter de nueva línea se representa con \n.

1.2.4. MOTORES DE CORRIENTE DIRECTA (DC)

Este tipo de dispositivo o componente pertenece a la familia de actuadores eléctricos, los cuales son muy utilizados en el campo de la robótica y mecatrónica y generalmente se emplean en la vida cotidiana en casi todos los aspectos. Sus siglas en inglés son DC (direct current), es decir, motores de corriente directa.

Para poder producir movimiento en cualquier sistema mecánico estos motores transmiten energía, por lo cual basan su principio de funcionamiento en la repulsión ejercida entre los polos magnéticos entre el rotor y el estator ya que constan con la misma polaridad o signo, siendo que entre los polos norte-sur del imán ubicado en el estator del motor el eje del rotor gira de manera libre tal como se muestra en el gráfico.

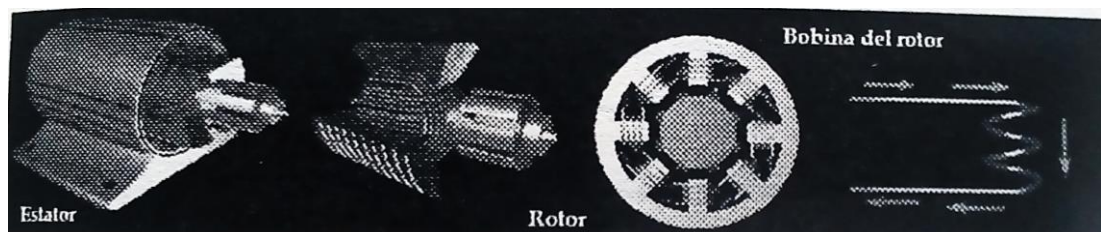


Figura 6. Funcionamiento de un motor eléctrico (Reyes Cortés & Cid Monjaraz, 2015)

El motor eléctrico se compone principalmente del rotor, estator, conmutador y la armadura.

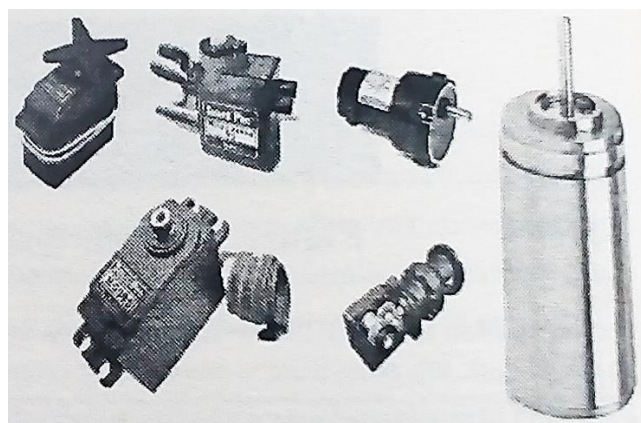


Figura 7. Tipos de motores de corriente directa (Reyes Cortés & Cid Monjaraz, 2015)

1.2.5. MÓDULO L298N

Este tipo de módulos utiliza el principio de funcionamiento del famoso “Puente H” para el control de motores eléctricos o motores paso a paso (sentido de giro y velocidad).

El circuito del puente H permite que los motores DC giren en ambos sentidos, que avancen o retrocedan. En ciertos circuitos integrados los puentes H ya vienen hechos, pero es posible construirlos utilizando Los siguientes componentes electrónicos:

- transistores 2N2222 (conmutaran las salidas)
- transistores TIP31 (actúan como interruptores)
- Diodos rectificadores (crean un puente de diodos)

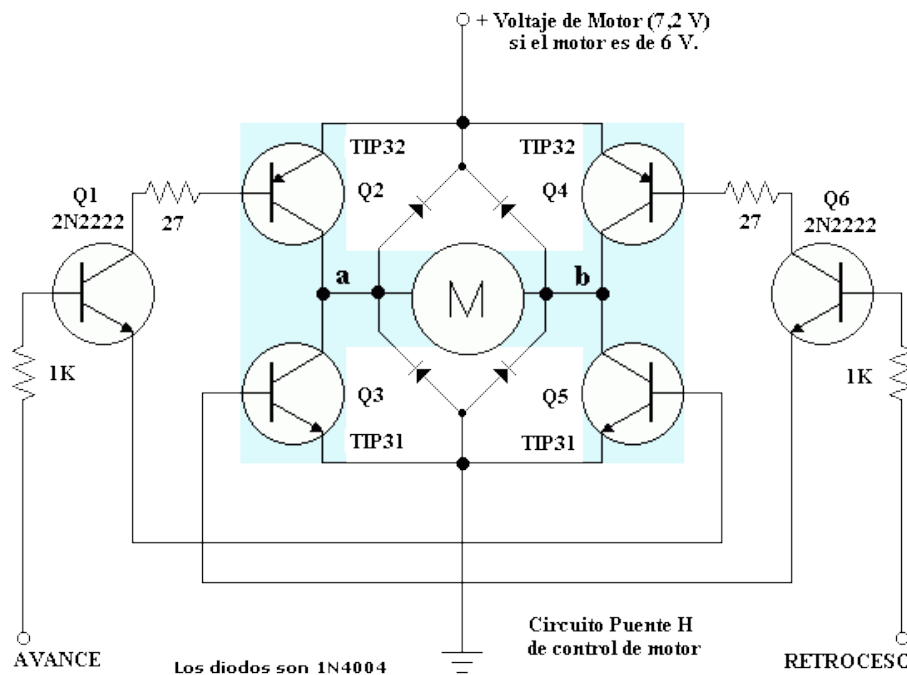


Figura 8. Esquema de un circuito de puente H (Frank, 2017)

Los componentes que conforman la estructura básica de estos módulos son:

- Driver L298N
- Diodos de protección
- Regulador de voltaje (78M05)

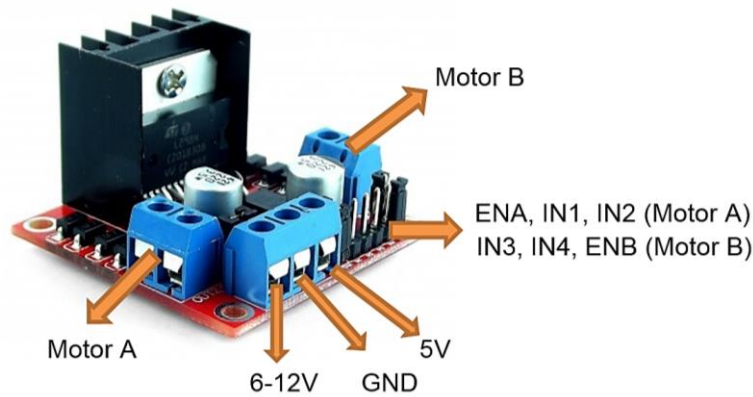


Figura 9. Módulo L298N (Naylamp Mechatronics SAC., 2016)

En la figura 10 se puede apreciar que estos módulos cuentan con un conector de 6 pines donde ingresan las señales TTL (transistor-transistor logic) que comandan los motores, una bornera de alimentación dependiendo de la necesidad o demanda de energía de los motores, y dos borneras para la conexión de los motores.

1.2.6. BLUETOOTH

Es una red inalámbrica de tipo WPAN (Wireless personal área network), ósea de área personal, la cual permite la comunicación de datos y de voz entre dispositivos de manera más fácil utilizando un enlace de radiofrecuencia en la banda de 2.4 GHz, dicha red tiene su especificación industrial BLUETOOTH. También se puede entender como un protocolo que de forma inalámbrica transmite datos especialmente diseñado para dispositivos de consumo bajo los cuales requieren de un corto alcance de emisión a un bajo costo.

En la Tabla 2 se detallan los tipos de dispositivos Bluetooth de acuerdo a la potencia emitida en la transmisión de datos, dándose tres tipos de dispositivos. La efectividad de cobertura de un dispositivo de segunda clase (2), se puede extender cuando es conectado con un transceptor de primera clase (1). Los dispositivos de clase 1 cuentan con características que les permiten señales de otros dispositivos a pesar de contar con una potencia de transmisión más débil.

Tabla 2. Dispositivos Bluetooth de acuerdo a la potencia de transmisión (Reyes Cortés & Cid Monjaraz, 2015)

Clase	Potencia máxima emitida		Alcance aproximado
	mW	dBm	
1	100	20	~ 30 m
2	2.5	4	~ 5 a 10 m
3	1	0	~ 1 m

El ancho de banda de los dispositivos Bluetooth puede establecer otra clasificación tal como se indica en la Tabla 3.

Tabla 3. Dispositivos Bluetooth de acuerdo al ancho de banda (Reyes Cortés & Cid Monjaraz, 2015)

Versión	Ancho de banda (Mbit/s)
1.2	1
2.0+EDR	3
3.0+HS	24
4.0	24

Su última versión es la 5.0, muy utilizada y dirigida para los smartphones de gama alta, la cual permite incrementar la distancia, capacidad y velocidad de manera significativa en la comunicación y transmisión de datos manteniendo el consumo de energía con respecto a sus anteriores versiones, estas prestaciones técnicas benefician mucho al operador, en especial cuando se trata de aplicaciones críticas en dispositivos de seguimiento médico.

1.2.6.1. Breve historia del Bluetooth

Esta herramienta red inalámbrica comenzaría a ser desarrollada por Jaap Haartsen y Mattisson Sven en 1994, a la vez que trabajaban para la empresa Ericsson en Lund, ciudad de Suecia, como una tecnología que reemplazaría a los cables y los conectores eléctricos. Bluetooth Special Interest Group fue quien publicara por primera vez sobre su desarrollo el 20 de mayo de 1998 a manera de un anuncio formal. Ericsson, Intel, Toshiba y Nokia fueron sus creadores. Ya más de 20,000 empresas cuentan con su membresía actualmente. La retro-compatibilidad es algo esencial y característico dentro de los diseños de todas las versiones estándares de Bluetooth, lo que quiere decir que la más moderna versión estándar cubre con todas las especificaciones de las versiones anteriores.

La idea del nombre Bluetooth fue propuesta en 1996 por Jim Kardach (desarrollador del sistema que permite la comunicación entre computadoras y teléfonos móviles y el enlace entre sistemas digitales), quien tomó como referencia e inspiración el nombre del rey danés y noruego Harald Blatand (935 D.C.), cuya traducción se ha dado como Harold Bluetooth, quien fue conocido por unificar a las tribus danesas, suecas y noruegas al cristianismo. El logo de Bluetooth es una representación de las iniciales en símbolos rúnicos de Hagall Berkana, la cuales forman un tipo de alfabeto de letras o runas, símbolos muy importantes utilizados durante la cristianización de la región representando un medio de comunicación codificado y datan de la época antigua y Edad Media (150 D.C.).

1.2.7. CONCEPTOS DE ELECTRÓNICA BÁSICA

1.2.7.1. El Transistor y reguladores

Dentro de la rama y clasificación de los semiconductores se encuentra este desarrollo electrónico, el transistor, el cual cuenta con propiedades muy ideales para los sistemas lógicos, ya sea de conmutación o secuenciales, aunque fue utilizado en el campo de la telecomunicación inicialmente. Es capaz de entregar ganancias de intensidad y tensión para excitar o alimentar varios componentes de su similitud con tan solo adicionar resistencias sin necesidad de alguna amplificación que suplementaria en el circuito. Para el transistorizado de circuitos o sistemas lógicos existe una gran variedad de componentes pero el transistor PNP es el que otorga mayores ventajas, el cual cuenta con tres regiones o partes, dos regiones de material P denominadas emisor y colector las cuales aceptan electrones, y una región de material N denominada base la cual dona electrones, tal como se ilustra en la Figura 12.

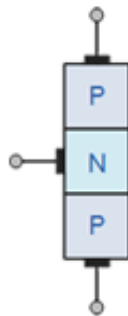


Figura 10. Esquema de los transistores PNP (Llamas, 2016)

Según este esquema si son conectadas dos fuentes CC, se puede apreciar que:

- En caso de ser negativa la base en relación al emisor, entre la base y el emisor una corriente I_b se establecerá, y más importante aún, entre el colector y el emisor se establecerá una corriente I_c . La corriente I_c crecerá más en cuanto el potencial de la base sea más negativa.
- La corriente I_c desaparecerá si la conexión es modificada para obtener un potencial de base más positivo que el emisor.

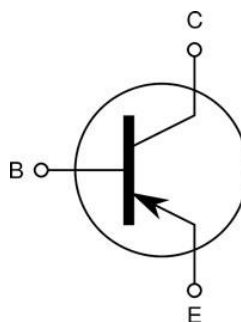


Figura 11. Simbología de un transistor (Cinjordiz, Infootec, 2018)

1.2.8. RELÉS Y ELEMENTOS DE PROTECCIÓN

Se puede denominar o llamar relé a todo dispositivo que basa su funcionamiento en el cambio de características de funcionamiento de otros elementos ya sean eléctricos o mecánicos dentro del mismo circuito o en otros circuitos eléctricos distintos. Se puede aclarar y entender mucho más esta definición con unos ejemplos prácticos.



Figura 12. Relé de armaduras (Cinjordiz, Infootec, 2018)

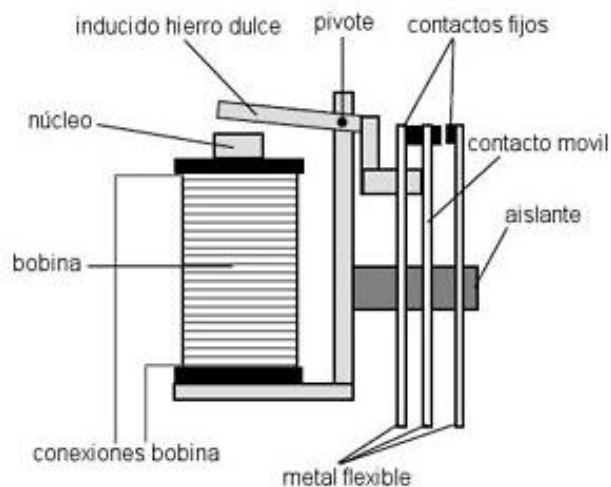


Figura 13. Partes de un relé de armaduras (Cinjordiz, Infootec, 2018)

En la Figura 16 se puede apreciar el diagrama de un interruptor de potencia, su conexión se la realiza a mano, en estado de tensión mecánica el interruptor al cerrarlo permanece sujeto por un trinquete solidario a la armadura de un relé ya que se acciona el muelle antagonista. El relé se constituye de una bobina con espiras de pletina de cobre la cual establece su conexión en serie con el circuito.

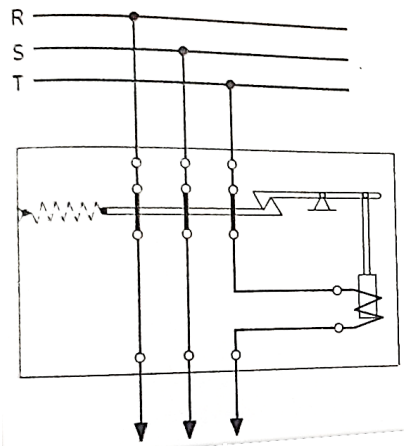


Figura 14. Conexionado relé de protección (Oñós Prados, Ramírez Vázquez, Ruiz Vasallo, & Pecanins Vallés, 1996)

En el siguiente caso el relé es utilizado para el accionamiento de un circuito eléctrico diferente. En este caso el relé funciona en base a la variación de las características de funcionamiento del circuito eléctrico mediante la conexión y desconexión del interruptor de mando, y permite que otros dispositivos funcionen, es decir, la armadura solidaria con los contactos del interruptor de potencia, en otro circuito eléctrico.

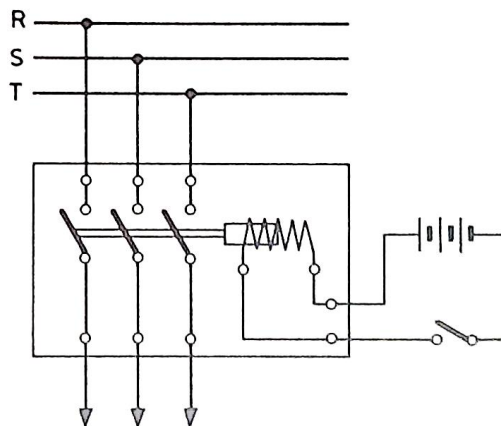


Figura 15. Conexionado relé de mando (Oñós Prados, Ramírez Vázquez, Ruiz Vasallo, & Pecanins Vallés, 1996)

De esta manera, de acuerdo con el objetivo de la instalación es posible establecer una primera clasificación de relés:

1. Relés de protección
2. Relés de mando
3. Relés de medida

Es posible clasificar a los relés en base a la función que estos deben cumplir en un circuito eléctrico ya que dependen de sus propias características constructivas.

1. Relés de funcionamiento continuo
 - a. Relés limitadores
 - b. Relés de comparación de módulos de magnitudes de acción
 - c. Relés de fase
 - d. Relés analógicos
2. Relés de funcionamiento por todo o nada
 - a. Relés instantáneos
 - b. Relés temporizados
 - c. Relés secuenciales

El presente proyecto tuvo como objetivo general implementar en un motor de encendido tipo CRDI existente en la Universidad Tecnológica Equinoccial, un sistema de encendido y activación de actuadores de un circuito de bloqueo central comandado por una aplicación de teléfono celular mediante la programación en una tarjeta Arduino. (González López, Mas Fito, & Vidal Pastor, 2011). Para lo cual fue necesario elaborar un circuito simulado en el software Proteus, realizar las respectivas pruebas de funcionamiento mediante simulación para su construcción real, y elaborar guías de práctica de conexión y programación que ayuden a entender el funcionamiento de este sistema con el fin de ser utilizado o reproducido por estudiantes de la UTE.

2. METODOLOGÍA

Este proyecto se realizó en la ciudad de Santo Domingo de los Tsáchilas en la Universidad UTE sede Santo Domingo, para el cual utilicé la metodología experimental, científica y analítica. Mediante la metodología científica desarrollé la estructura de la maqueta creando diseños de los circuitos de conexión de los componentes utilizados en el software Proteus (v8.7), como de la cubierta y soporte de la misma en Adobe Illustrator (2018), y se comprobó funcionamiento correcto de los elementos electrónicos a utilizar, los módulos de control o mando de relés y motores, el circuito regulador para la correcta entrega de alimentación al sistema. Una vez realizado el diseño de conexión ideal tomando en cuenta eficiencia y distribución de componentes en el menor espacio posible se procedió a realizar la construcción física de la estructura de la maqueta. Para el diseño y programación de la aplicación de celular, la cual permite el comando del sistema, se hizo uso del software online App Inventor y para la programación del sistema, el software Arduino (v1.8.8), para ello se empleó la metodología experimental ya que esta permitió elegir el método de control más adecuado y agradable para el usuario considerando tanto la sencillez y fácil utilización como la estética del proyecto. En cuanto al análisis de la interacción entre dispositivos y el correcto funcionamiento de cada uno de ellos se tomó en cuenta las siguientes variables:

2.1. ALIMENTACIÓN

Se midió y entregó la alimentación necesaria al sistema partiendo de una batería de auto común, energía eléctrica que se distribuye dentro de este hacia cada uno de los componentes y módulos de control previniendo sobre cargas que puedan perjudicar a estos componentes con la ayuda de un pequeño circuito regulador, lo cual se logró mediante las simulaciones el Proteus en donde se obtuvieron los valores de voltaje y amperaje adecuados para el correcto funcionamiento del sistema. También se tomó en cuenta la alimentación entregada por la maqueta para el encendido del MCI, comprobando así la eficiencia y confiabilidad del sistema.

2.2. SEÑALES DE RESPUESTA

Se analizó, midió y comprobó las señales digitales de respuesta entregadas por los módulos de control (Tarjeta Arduino, módulo LM298N) y las señales eléctricas entregadas por los relés de mando para el control de los actuadores y el encendido del MCI verificando voltajes, amperajes y ondas de señales digitales con la ayuda de un multímetro automotriz y de un osciloscopio automotriz. (Reyes Cortés & Cid Monjaraz, 2015)

2.3. PROGRAMACIÓN

Se aplicó el lenguaje y método de programación más sencillo de entender y aprender tanto del módulo Arduino como de la aplicación de celular, con el fin de que el sistema pueda ser simulado y reproducido por estudiantes de la universidad, para lo cual en el caso de la programación en Arduino se utilizó comandos y librerías muy útiles y necesarias que permiten entender mucho más el lenguaje y el mundo de la programación. En cuanto a la programación de la aplicación en App Inventor se aplicó condiciones de programación que permitan el fácil uso de ésta, y la comunicación entre dispositivos (módulo Bluetooth Arduino y celular).

2.4. DISEÑO Y SIMULACIÓN DEL CIRCUITO DE CONTROL

Mediante la utilización del software de simulación de circuitos y equipos electrónicos Proteus se procedió a diseñar la estructura del circuito que controla el arranque y apagado de un MCI, como también el accionamiento de los motores de un sistema de bloqueo central, por medio de las órdenes recibidas por una aplicación de celular vía Bluetooth, para lo cual se consideró factores importantes tales como: la alimentación o fuente del sistema y de cada uno de los componentes para su óptimo funcionamiento, el módulo central encargado del comando de todos los elementos que intervienen en el circuito de control mediante la lectura del programa en lenguaje C, el módulo de comunicación inalámbrica vía Bluetooth, los dispositivos a emplearse para el control del sentido de giro de los actuadores o motores DC, y los dispositivos y/o componentes electrónicos de los circuitos optoacopladores encargados de la apertura y cierre del suministro de energía para el control del encendido o apagado del MCI.

Los dispositivos y componentes electrónicos que intervienen en el circuito de control son:

- Módulo Arduino
- Módulo Bluetooth HC-06
- Módulos L298N
- Regulador LM7805
- Relés
- Botón interruptor
- Fusible

- Condensadores cerámicos
- Resistencias
- Diodos
- Transistores
- Conectores o cables

En la Figura 16 está representado el esquema del circuito de control en Proteus.

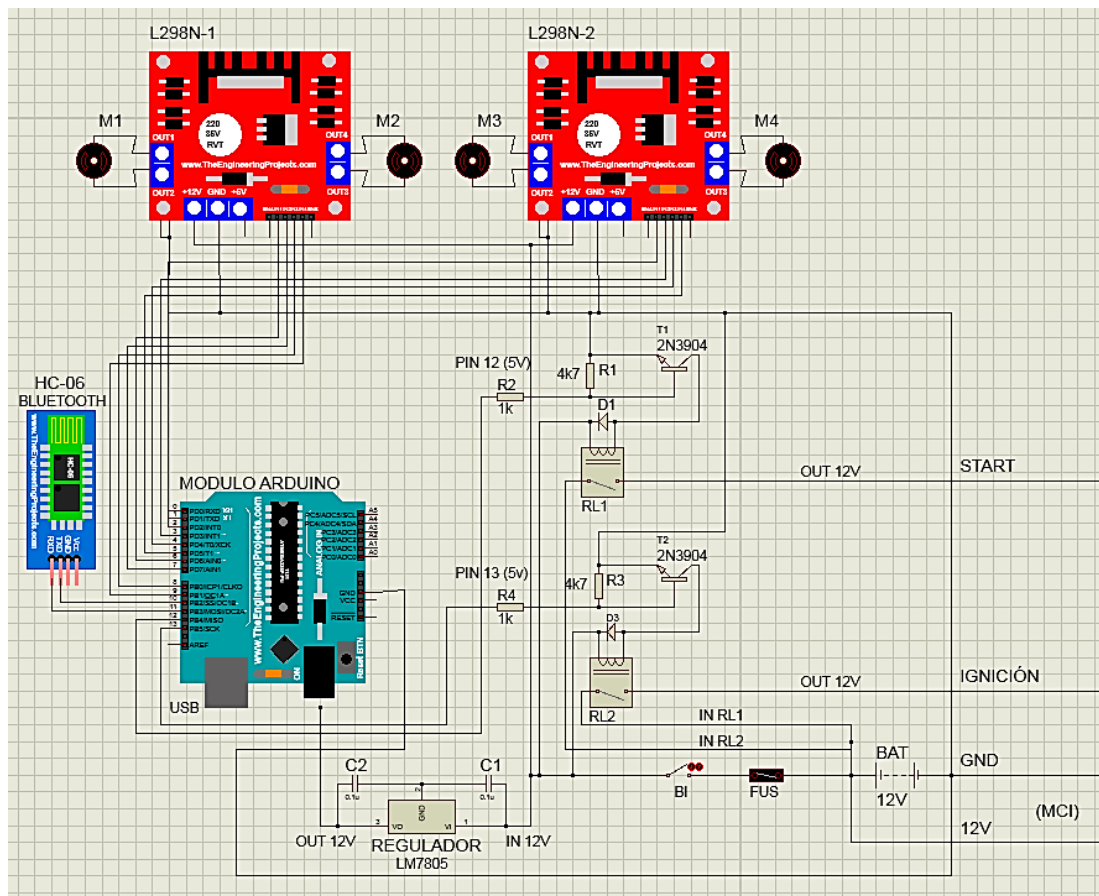


Figura 16. Simulación del circuito de control

Para el suministro de energía del circuito de control se tomó como fuente a la batería del banco de pruebas. Partiendo de esta fuente de alimentación, el polo positivo de la batería se conecta a un fusible y seguido a un interruptor. Al ser este accionado permite suministrar energía a todos los dispositivos internos del circuito de control, en otras palabras, se enciende el sistema. La energía o voltaje llega hasta un circuito regulador, el cual se encarga de otorgar de 5 a 6 V al módulo Arduino. El dispositivo Bluetooth HC-06 se conecta a este módulo mediante dos pines configurados para su comunicación. Al recibir la orden por parte de la aplicación de celular

enlazada con el dispositivo Bluetooth, del módulo Arduino parte 8 señales de los pines digitales hacia los módulos L298N, estos módulos permiten el control de los cuatro actuadores del sistema de bloqueo central utilizando como principio de funcionamiento el circuito de “puente H”. Los pines 12 y 13 de la tarjeta Arduino comandan el accionado de los relés mediante dos circuitos optoacopladores los cuales permiten enviar una señal de 12 V hacia ignición del swtich de encendido y una señal hacia start o arranque, de esta manera se controla el encendido y apagado del MCI. De tal manera que se ha establecido las partes que conforman el circuito de control en las cuales se ven distribuidos todos los elementos que conforman este sistema así:

- Fuente de alimentación
- Circuito regulador
- Módulo central
- Dispositivo de comunicación inalámbrica
- Módulos de control de actuadores
- Circuitos optoacopladores

2.4.1. FUENTE DE ALIMENTACIÓN

La implementación de este sistema está dirigida hacia los vehículos automotores, para lo cual se dispone siempre de una alimentación o fuente de una batería de auto, eso quiere decir que este sistema depende siempre de una fuente de 12 V, su diseño y funcionamiento se basan en este valor de voltaje, y por ellos es menester que se disponga de una batería de auto en buen estado y cargada. Tomando en cuenta todo esto se ha establecido en el software de simulación que la fuente de este circuito será una batería de 12 V, tal como se indica en la Figura 17.

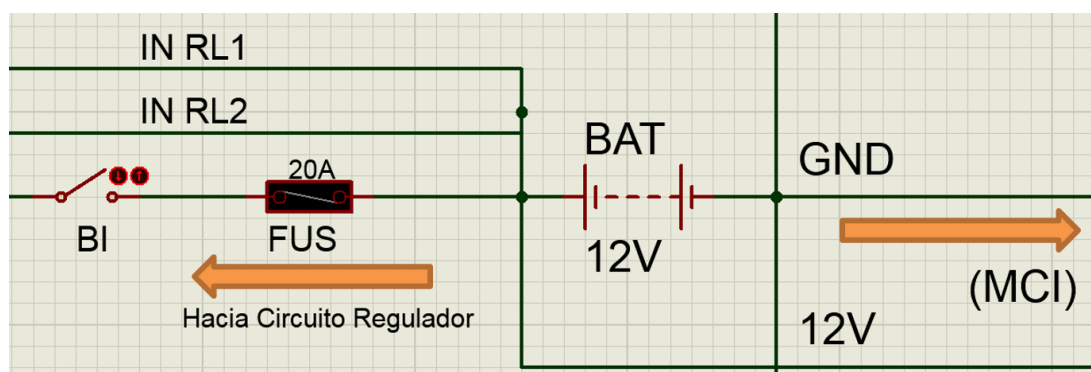


Figura 17. Simulación de la fuente de alimentación

La alimentación directa entregada por la batería hacia el sistema se ve interrumpida por un fusible (20 A, 250 V), que cumple la función de protegerlo en caso de algún cortocircuito o sobrecarga, de igual manera existe un botón interruptor el cual permite conectar y desconectar al circuito de su fuente a voluntad para evitar daños en caso de alguna emergencia o imprudencia. La batería otorga directamente su voltaje y corriente a los relés por medio de los conectores “IN RL1” y “IN RL2” debido a que estos dispositivos son menos propensos a daños de este tipo. Otro motivo es porque no se quiere que el voltaje y amperaje se disipen al pasar por el fusible o el interruptor ya que los relés tienen la función de permitir el paso de estas señales directamente al switch dando un mismo resultado en el encendido. De igual manera se ha determinado en la simulación que la batería suministre energía al MCI, como es el caso del banco de pruebas.

2.4.2. CIRCUITO REGULADOR

Se encarga de entregar la correcta alimentación a la tarjeta Arduino previniendo sobrecargas que puedan dañarla. Este circuito tiene como principal componente el regulador LM7805, el cual al serle entregado una alimentación de 12 V directa de la batería de auto, entrega un voltaje simplificado de 5.5 V a 6 V al módulo Arduino, ideal para su correcto funcionamiento.

Este regulador trabaja con dos condensadores los cuales permiten el correcto flujo de energía de alimentación o una “corriente más limpia” a la tarjeta Arduino, tal como se demuestra en la simulación realizada en la Figura 17.

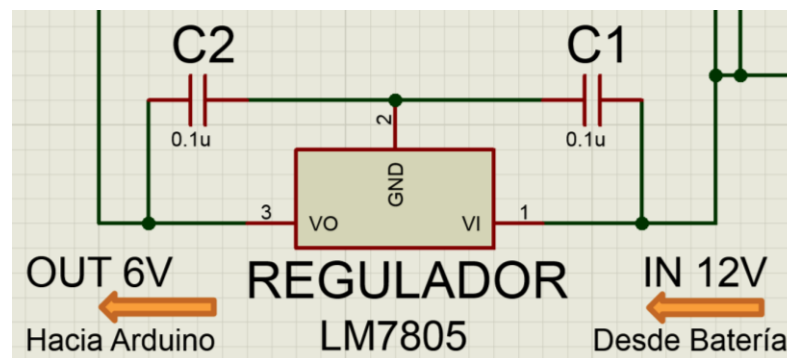


Figura 18. Simulación del circuito regulador de voltaje

2.4.3. MÓDULO CENTRAL

La tarjeta Arduino es el cerebro o módulo principal para el comando de todos los dispositivos que conforman el circuito de control, ya que recibe señales de orden por parte de la aplicación de celular la cual está enlazada con el módulo Bluetooth. A esta tarjeta se le puede cargar un programa en lenguaje

C por medio de un puerto USB (Type A o Type B, dependiendo del modelo) el cual es procesado y leído por el microcontrolador para enviar señales de mando a los módulos L298N como a los relés de control. Los pines que se emplearon para la comunicación Bluetooth son el pin 10 y el pin 11, los que permiten el control de los actuadores son los pines 2, 3, 4, 5, 6, 7, 8 y 9. Los pines que se utilizaron para el encendido del banco de pruebas y el arranque del MCI son el pin 13 y el pin 12 respectivamente. La señal a tierra del módulo Arduino (GND) es conectada con la señal negativa del sistema, de esta manera se incluye el módulo Arduino al circuito de control.

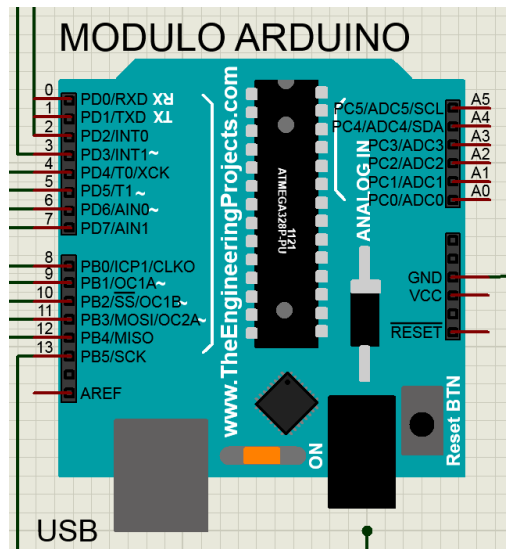


Figura 19. Implementación de la tarjeta Arduino en la simulación

Para la implementación de una tarjeta Arduino en Proteus fue necesario contar con una librería que facilite el uso de dispositivos Arduino (como tarjetas, módulos de control o sensores), las cuales no se incluyen en el paquete de instalación del software y deben ser adquiridas en internet.

2.4.4. DISPOSITIVO DE COMUNICACIÓN INALÁMBRICA

El módulo Bluetooth tiene una función clave en el circuito de control, ya que por medio de éste existe un enlace o comunicación inalámbrica con la aplicación de celular en el dispositivo móvil, lo cual permite comandar a voluntad el encendido y apagado del MCI, como también el accionado de todos y cada uno de los actuadores. Para este proyecto se ha optado por utilizar el módulo Bluetooth modelo HC-06, ya que este en específico actúa nada más que en modo esclavo, es decir, que sólo recibe órdenes, lo que los diferencia de los HC-05, que pueden ser programados para actuar en modo esclavo o maestro, algo innecesario en este proyecto. Éste módulo es conectado utilizando los pines TXD y RXD a los pines 10 y 11 de la tarjeta Arduino para establecer comunicación y enviar órdenes al módulo por medio de la aplicación.

En la Figura 20 se demuestra el diseño gráfico que este dispositivo tiene en la simulación. A diferencia de un dispositivo Bluetooth real, este en las simulaciones de Proteus no necesita conectarse a una fuente de alimentación para su funcionamiento, por lo que no se utilizan los pines de conexión Vcc (+) y GND (-).

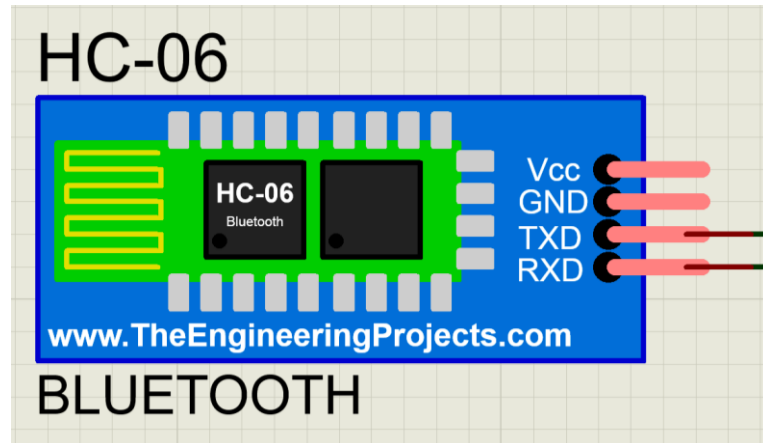


Figura 20. Implementación del módulo Bluetooth en la simulación

Al igual que la tarjeta Arduino, éste dispositivo solo puede ser utilizado en Proteus mediante las librerías de dispositivos Arduino.

2.4.5. MÓDULOS DE CONTROL DE ACTUADORES

Para este proyecto se ha decidido en emplear los módulos L298N debido a su fácil utilización y eficiente desempeño. Estos módulos tienen la función específica de controlar el sentido de giro y velocidad de motores DC o motores paso a paso, en este caso se utilizan dos, los cuales controlan cuatro actuadores o motores de un sistema de bloqueo central genérico, por lo que no es necesario controlar la velocidad de estos. Estos módulos están diseñados para trabajar muy de la mano con el sistema Arduino, ya que al recibir una señal digital emitida por la tarjeta Arduino comandan el accionado de cierre o apertura de los actuadores (según la manera o polaridad en que se conecten).

Estos dispositivos son implementados en la simulación con la ayuda de una librería de módulos L298N. En la Figura 21 se muestra su representación gráfica en Proteus con sus respectivos actuadores, los cuales son conectados en los pines OUT1, OUT2, OUT3 y OUT4, dos pines para cada actuador, estos son comandados cuando ingresan señales a los pines IN1, IN2, IN3 y IN4, emitidas por los pines digitales del módulo Arduino.

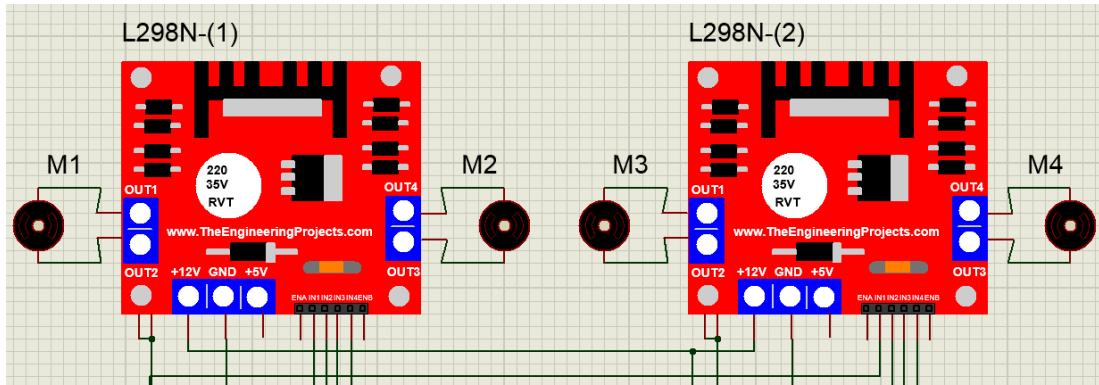


Figura 21. Implementación de los módulos L298N en la simulación

De los pines 2, 3, 4, 5, 6, 7, 8 y 9 de la tarjeta Arduino se envían señales a los módulos L298N para el control de actuadores, siendo que los pines 2 y 3 son para el actuador 1, los pines 4 y 5 para el actuador 2, los pines 6 y 7 para el actuador 3, los pines 8 y 9 para el actuador 4. Cada borne de dos pines de estos módulos comanda el sentido de giro y por ende la apertura o cierre de los actuadores, siendo que si es enviada una señal a uno de los pines el otro pin hará de tierra para el cierre del circuito provocando el giro del motor en un sentido, si estos pines intercambian papeles el sentido de giro del motor será opuesto.

A diferencia de una conexión real de estos módulos, en la simulación es necesario suministrar energía en ambos pines de alimentación (+12V y +5V), lo que provoca que suministren un voltaje mayor para el accionamiento de los motores en la simulación.

2.4.6. CIRCUITOS OPTOACOPLADORES

Los circuitos optoacopladores cumplen la función de permitir la conexión directa de alimentación desde la batería de 12 V hasta el switch de encendido (uno al cable de ignición y, otro al cable de arranque) cuando reciben una señal pequeña, las cuales son emitidas por el módulo Arduino desde los pines digitales 12 y 13, según se ordene desde la aplicación de celular. Desde el pin 13 se emite una señal estable cuando el operador presione el botón de "IGNICIÓN" en la aplicación de celular, y desde el pin 12 un pulso dependiendo del tiempo de presionado del botón "START" para el arranque del MCI. En la Figura 22 se ve representado un diagrama con los estados del switch dependiendo de la posición de la llave.

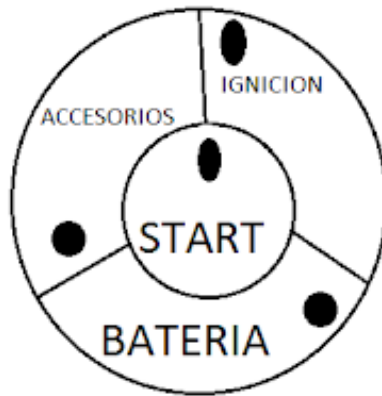


Figura 22. Diagrama del switch de encendido (Anónimo, 2016)

Cuando se pretende comandar un dispositivo que demande una alimentación más de la que pueda brindar el módulo Arduino sin sufrir daños (por ejemplo un televisor, ventilador, motor) es necesario contar con un circuito optoacoplador, el cual ayuda a alimentar al equipo a comandar con una fuente de alimentación externa al módulo Arduino al recibir una pequeña señal. En este caso se utilizaron dos relés 904 (de uso automotriz), los cuales cumplen la función de un optoacoplador ya que el voltaje y amperaje que circula por el sistema es elevado debido a que estamos hablando de un circuito que interviene en el sistema de encendido de un vehículo cualquiera. El circuito complementario del relé cuenta con un diodo para la protección del sistema, dos resistencias ($1K\Omega$ y $4.7K\Omega$) que mantienen el nivel de corriente ideal para el funcionamiento y accionamiento del relé y un transistor 2N3904 el cual amplifica la señal emitida por el módulo Arduino hacia el relé. En la Figura 23 se indica el esquema de conexión de los circuitos optoacopladores en la simulación.

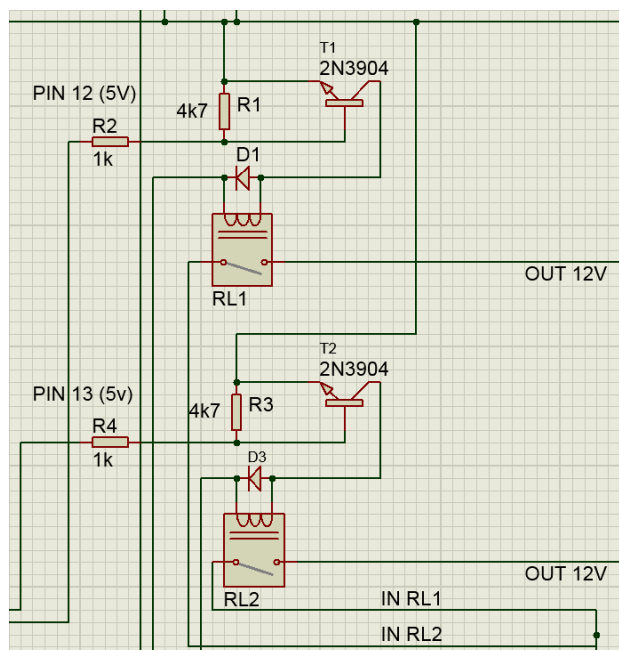


Figura 23. Simulación de los circuitos optoacopladores

2.5. PROGRAMACIÓN EN ARDUINO

El programa a cargar en el módulo Arduino se enfoca en establecer, mediante el lenguaje de programación C y lenguaje Arduino, una comunicación por medio del módulo Bluetooth a un celular o móvil, el cual cuente con un sistema operativo Android, además se establecieron condiciones que permitan manipular desde la aplicación de celular el encendido del MCI y la apertura y cierre de los actuadores del sistema de bloqueo central a través de módulos L298N los cuales están diseñados específicamente para el control de estos dispositivos. Como todo programa en Arduino, éste se compone de tres partes:

- Declaración de variables
- Sección Setup
- Sección Loop

2.5.1. DECLARACIÓN DE VARIABLES

En esta parte o sección del programa Arduino se declaran las variables a utilizar en el resto de la programación para que el módulo pueda reconocerlas, al declararlas se deben indicar el tipo de variable y luego un nombre para identificarlas, de manera opcional a estas se les puede asignar un valor. Esta declaración se debe realizar una sola vez en cada programa. (Delgado, 2016)

En la Figura 24 se detalla la declaración de variables para este programa, en este caso se incluyó la librería <SoftwareSerial.h>, la cual permite la comunicación serial en otros pines digitales del Arduino, en este caso, la comunicación con el dispositivo Bluetooth HC-06 en los pines 10 y 11, ya que de esta manera se demuestra cómo puede establecerse una comunicación con este dispositivo sin la utilización de los pines 0 y 1 (RX y TX) diseñados para esa función en específico. También se declaró la variable “Mensaje”, la cual se aplica en el envío de datos u órdenes por parte de la aplicación de celular.

```
#include <SoftwareSerial.h>

SoftwareSerial Serial_2 (10, 11);
String Mensaje;
```

Figura 24. Declaración de variables del programa en Arduino

2.5.2. SECCIÓN SETUP

En esta sección del programa, llamada “void setup”, se especifica el valor en Baudios a la que va a trabajar el puerto serial, para este caso se indicó que será de 9600. También se indicó el estado de los pines a utilizar para este programa, en este caso los pines 2, 3, 4, 5, 6, 7, 8, 9, 12 y 13 fueron configurados como pines de salida de señal por medio del comando “OUTPUT”, tal como se indica en la Figura 25.

```
void setup() {  
    Serial_2.begin(9600);  
    Serial.begin(9600);  
    pinMode(13, OUTPUT);  
    pinMode(12, OUTPUT);  
    pinMode(2, OUTPUT);  
    pinMode(3, OUTPUT);  
    pinMode(4, OUTPUT);  
    pinMode(5, OUTPUT);  
    pinMode(6, OUTPUT);  
    pinMode(7, OUTPUT);  
    pinMode(8, OUTPUT);  
    pinMode(9, OUTPUT);  
}
```

Figura 25. Sección Setup del programa en Arduino

2.5.3. SECCIÓN LOOP

En esta sección se crean las condiciones de la programación aplicando las variables declaradas, para lo que es necesario indicar el estado inicial de los pines a participar en la ejecución del programa, en este caso, todos los pines inician en estado 0 o “LOW”, lo cual indica que no emiten ningún tipo de señal hasta que se les indique. Luego se emplearon valores alfabéticos o letras que se almacenan en la variable “Mensaje” para el envío de datos y por ende la orden a solicitar por parte de la aplicación de celular al módulo Arduino, lo cual indica que pin debe emitir señal permitiendo el control de los módulos L298N y los relés de mando, tal como se detalla en la Figura 26 (parte de la programación).

```

void loop() {
  while (Serial_2.available()) {
    delay(5);
    char c = Serial_2.read();
    Mensaje += c;
  }
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(12,LOW);
  digitalWrite(13,LOW);

  if (Mensaje.length()>0) {
    Serial.println(Mensaje);
    if (Mensaje == "a") {
      digitalWrite(13,HIGH);
    }
    if (Mensaje == "b") {
      digitalWrite(13,LOW);
    }
  }
}

```

Figura 26. Parte de la sección Loop del programa en Arduino

Este programa fue cargado al módulo Arduino de la simulación en Proteus para la realización de las respectivas pruebas de medición.

2.6. PRUEBAS DE MEDICIÓN EN LA SIMULACIÓN

Antes de proceder a la construcción del circuito en la práctica fue necesario realizar las respectivas simulaciones que nos comprueben que el circuito funcione como lo esperado, para ello se empleó los instrumentos de medición y comprobación virtuales (voltímetro, amperímetro y osciloscopio) del software Proteus.

Se comenzó a medir la fuente de alimentación que suministra al sistema de voltaje necesario para el funcionamiento correcto del mismo. Con la ayuda de un multímetro de CC se midió el voltaje emitido por la batería a manera de comprobación, la cual nos dio un valor de 12 V.

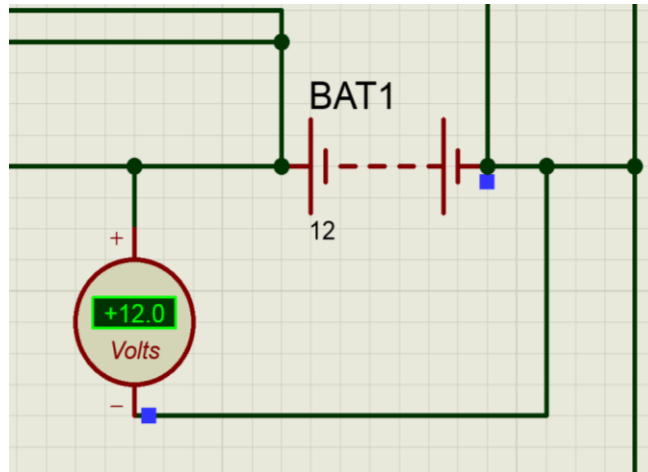


Figura 27. Verificación del valor de la fuente de alimentación

Con la ayuda del amperímetro se pudo medir la corriente que circula en el circuito, en este caso, el amperaje que emite la fuente de 12 V. Cabe mencionar que no se esperaban los mismos resultados en amperajes obtenidos en las mediciones de la práctica real. En la simulación podemos observar que se ha obtenido una corriente de 3.36 A, conectando el medidor en serie, entre la fuente y el fusible de protección, como se indica en la Figura 28.

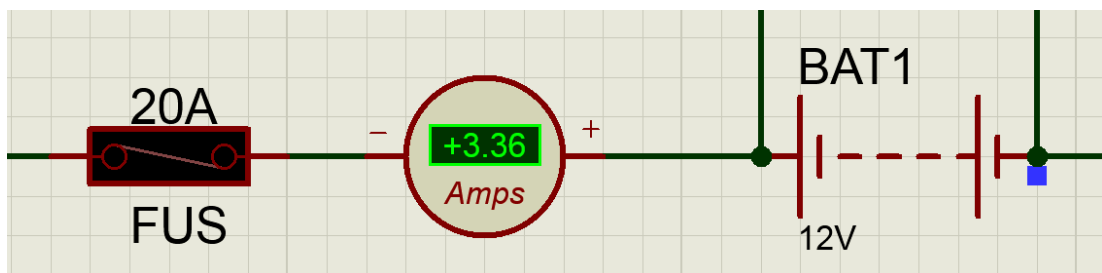


Figura 28. Obtención de amperaje emitido por la fuente de alimentación

Seguidamente se comprobó que el módulo Arduino recibía el voltaje necesario para su funcionamiento sin recibir daños. Se conectó el polo positivo del multímetro entre la salida de señal emitida por el circuito regulador y GND o tierra. De esta manera se verificó que el circuito regulador brindaba el nivel de voltaje necesario para el óptimo funcionamiento de la tarjeta Arduino (6 V) cumpliendo con su propósito en el sistema.

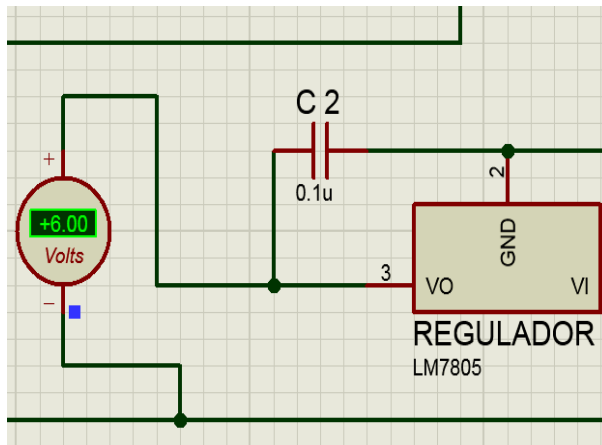


Figura 29. Verificación del funcionamiento del circuito regulador

Se comprobó también el correcto funcionamiento del módulo Arduino, para ello se conectó el polo positivo del multímetro en los pines 12 y 13, los cuales emiten una señal digital para el accionamiento de los relés y por consiguiente el encendido y apagado del MCI. En este caso ambos relés emiten el mismo valor de señal dentro de la simulación en Proteus por lo cual se evaluó solo la señal del pin 12, tal como se indica en la Figura 30. Se pudo verificar de esta manera que el módulo Arduino emite señales digitales de hasta 5 V, en este caso se obtuvo 4.92 V.

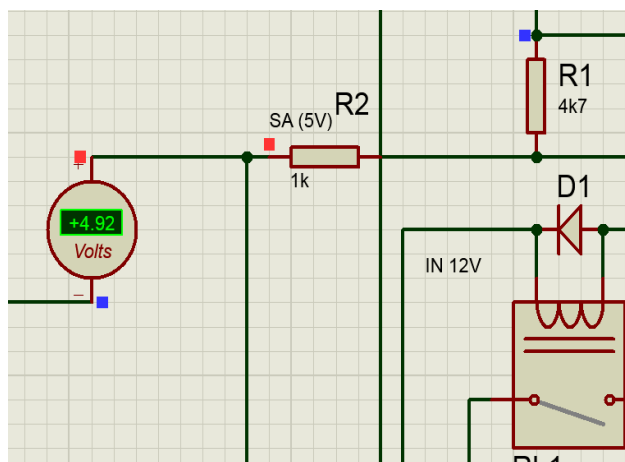


Figura 30. Verificación del valor de señal emitido por el módulo Arduino

También se diagnosticó el funcionamiento de los circuitos optoacopladores midiendo si existe paso de voltaje desde la fuente de 12 V hasta el switch de encendido, para ello se conectó el polo positivo del multímetro en la salida del relé hacia el switch y el polo negativo a tierra. De esta manera se pudo comprobar que existe un voltaje de 11.9 V, lo que quiere decir que los circuitos cumplen con su función al permitir que circule una fuente de alimentación hacia el banco de pruebas una vez que recibe la señal de mando del módulo Arduino.

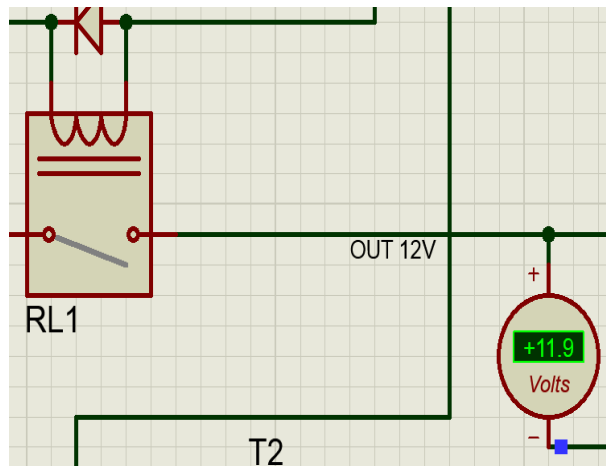


Figura 31. Verificación del funcionamiento de los circuitos optoacopladores

De igual manera se midió el valor en amperios obtenido en la salida de los circuitos optoacopladores con el fin de compararlo con el obtenido posteriormente en la práctica, en este caso el valor de la corriente ha disminuido debido a la resistencia que presentan los relés, dato que sí fue considerado por la simulación. Se obtuvo 0.99 A en la medición, como se indica en la Figura 32.

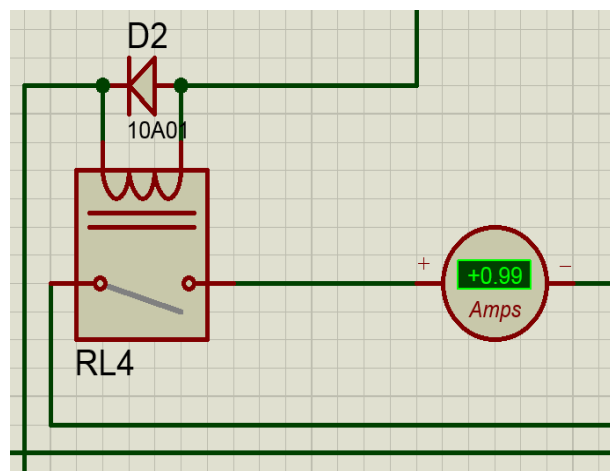


Figura 32. Obtención de amperaje de los circuitos optoacopladores

Se comprobó también los valores de señal emitidas por los módulos L298N en sus pines de salida (OUT 1, OUT 2, OUT 3 y OUT 4), que como ya se mencionó, estos valores tienden a ser mayor que los obtenidos en la práctica debido a que en la simulación se pide conectar o suministrar energía en ambos pines de alimentación para el funcionamiento de estos módulos.

Para la comprobación se conectó el polo positivo del multímetro entre el pin de salida del L298N y el actuador, mientras que el polo negativo va a tierra. Se obtuvo un voltaje de 11.1 V, valor suficiente para accionar los motores conectados, tal como indica la Figura 33.

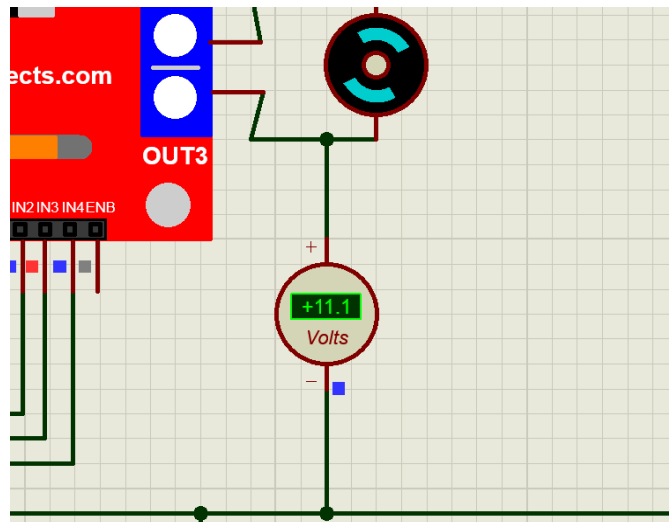


Figura 33. Obtención de señal de alimentación de actuadores

Finalmente se obtuvieron las ondas de señal digitales emitidas por los pines 12 y 13 con la ayuda del osciloscopio. En el pin 13 se obtuvo una señal estable ya que ésta acciona al relé que comunica a la fuente de 12 V con la señal de ignición del switch de encendido del banco de pruebas, y en el pin 12 se obtuvo un pulso de señal digital que acciona el relé que simula el pulso de arranque que realiza la llave en la posición de “start”. Para ello se conectó el osciloscopio con el conector del canal “A” en la salida del pin 13, prueba de medición que se la realizó durante 10 segundos. Ver Figura 34.

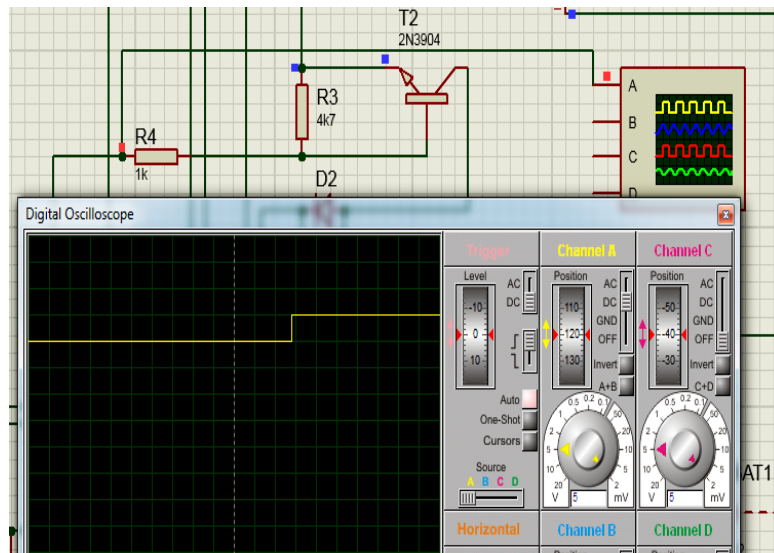


Figura 34. Onda de señal digital del pin 12 obtenida en la simulación

De igual manera para obtener la onda de señal del pin 12 se hizo uso del conector del mismo canal en la salida de este, dicha prueba se la realizó durante un lapso de 5 segundos, pudiendo apreciarse el pulso emitido por el módulo Arduino, el cual duró aproximadamente 1.3 segundos, como se indica en la Figura 35.

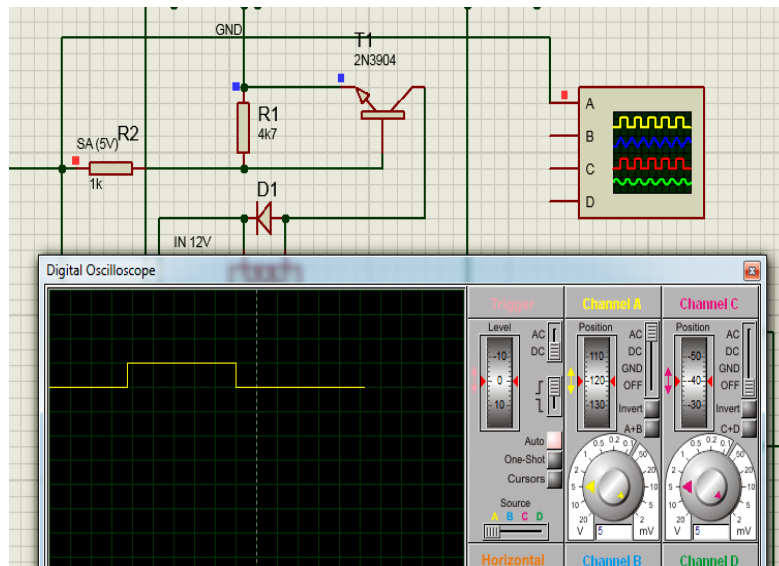


Figura 35. Onda de señal digital del pin 13 obtenida en la simulación

Una vez realizadas las pruebas de medición de voltajes, amperajes y ondas de señal digital se procedió a la distribución de elementos y armado del circuito de control.

2.7. CONSTRUCCIÓN DEL CIRCUITO DE CONTROL

Un circuito funciona mediante la interconexión de varios elementos eléctricos enfocados a realizar una tarea, ya sea entregar potencia, el procesado y transmisión de señales, medir alguna magnitud física o simplemente almacenar información. Un circuito cuenta con una entrada donde es aplicada una tensión o electricidad y una salida, donde se observa una tensión o electricidad. Se puede denominar *sistema* a la interconexión de circuitos, redes o incluso otros sistemas. (Fernández Ferrer, 1979)

Para la construcción física del circuito de control fue necesario contar con elementos de conexión como cables y bornes, y un sistema de alimentación común, elementos que muchas veces no son considerados en una simulación por computadora, en este caso en el software Proteus, siendo que las propiedades de estos elementos pasan por alto, como por ejemplo la resistencia ante el paso de la corriente que estos tienen, por lo que se esperaba que en las mediciones de amperaje comparadas entre la simulación y la práctica para la demostración de funcionamiento existan diferencias.

Los cables permitieron conducir la alimentación a todo el sistema desde la batería de 12 V, como también el establecer las conexiones entre todos los dispositivos del circuito de control, entre: circuito regulador, módulo Arduino, módulo Bluetooth, módulo L298N, relés de mando, circuitos

optoacopladores, switch de encendido y actuadores del bloqueo central. Se utilizaron cables flexibles N° 10, 12, 14, y cables para protoboard.



Figura 36. Cables flexibles N° 10,12 y 14

Además se necesitó de un cable común USB para el envío de programas a la tarjeta Arduino. En el interior de la maqueta se conectó al módulo Arduino un adaptador que permite la comunicación con un cable común USB.

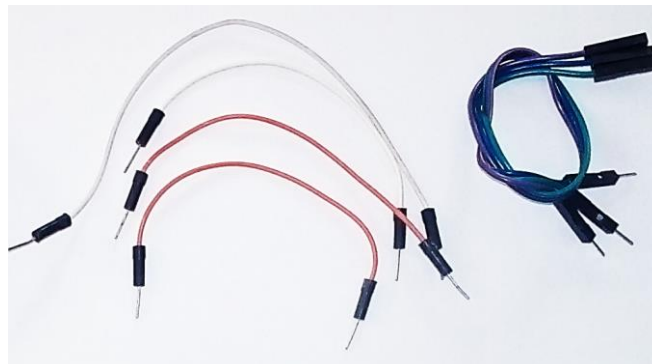


Figura 37. Cables de protoboard



Figura 38. Cable común USB



Figura 39. Adaptador USB

Tal como en la simulación, este circuito partió de una fuente de alimentación de 12 V, en este caso es la batería del banco de pruebas de un motor diésel tipo DIS existente en el taller automotriz de la UTE.



Figura 40. Batería de 12 V

Esta fuente se conecta a los bornes ubicados en la cara frontal de la cubierta, los cuales permiten establecer conexiones desde el exterior del sistema al interior de este, ósea, al circuito de control. Estos ayudan, por medio de cables, que la alimentación llegue desde la batería al circuito regulador, como también a que las señales emitidas por los relés lleguen hasta el switch de encendido del MCI.

Los soquets cumplen la misma función, pero estos permiten la llegada de señal desde el exterior de la maqueta del MCI al interior hacia el switch de encendido.



Figura 41. Bornes de conexión

El circuito continúa desde el borne de conexión positivo hacia un porta fusible de protección añadido al sistema, y luego hacia un botón interruptor ON/OFF que permite el paso de los 12 V hacia un componente de alimentación común, para lo cual se ha empleado una pequeña barra metálica con agujeros enroscados en los cuales se sujetan los cables que alimentan los dispositivos dentro del circuito de control con la ayuda de tornillos.



Figura 42. Porta fusible y fusible



Figura 43. Botón interruptor



Figura 44. Barra metálica de alimentación común

Para el circuito regulador se ha utilizado una baquelita a la cual se le diseñó la pista necesaria para conectar los respectivos elementos electrónicos (2 condensadores y un regulador LM7805). Se tomó como referencia el circuito simulado en Proteus. Posterior a ello se pintaron con un espray color negro la parte posterior a la pista para darle un acabado más estético.

Este circuito se conecta al módulo Arduino por medio de un plug de adaptador de 9V, el cual es compatible con el conector DC del módulo ya que el puerto USB no sirve para alimentar el sistema, sino que éste es utilizado solamente para cargar programas al módulo. Circuito regulador en físico en la Figura 46.

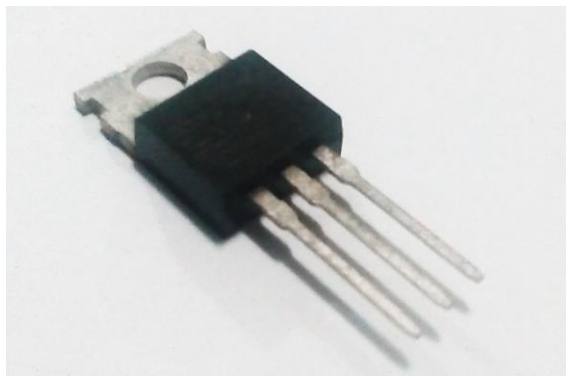


Figura 45. Regulador LM7805

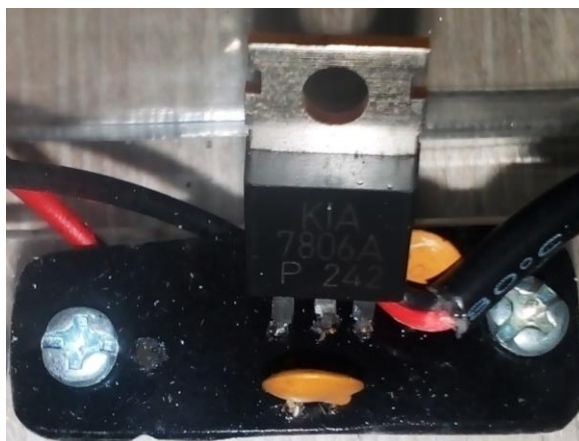


Figura 46. Circuito regulador de voltaje

Para el módulo central se decidió utilizar una tarjeta Arduino modelo Leonardo, debido a que éste es una variante con modalidades más completas a las del modelo UNO, siendo compatible con más dispositivos y sensores disponibles en el mercado y por ende con más librerías Arduino, por lo que no se limita a la modificación y mejora que se le pueda hacer este sistema en un futuro.

En la Figura 47 se muestra un esquema del módulo Arduino Leonardo donde se indican los pines digitales y analógicos con los que cuenta este modelo, además se muestra el tipo de microcontrolador (Atmega32U4) con el que cuenta para el procesamiento de información, y los puertos por donde se pueden conectar a una fuente de alimentación. Al igual que la mayoría de tarjetas Arduino esta cuenta con un botón de “reset” o reinicio, el cual permite reiniciar a la tarjeta en caso de necesitar interrumpir el programa que se esté ejecutando.

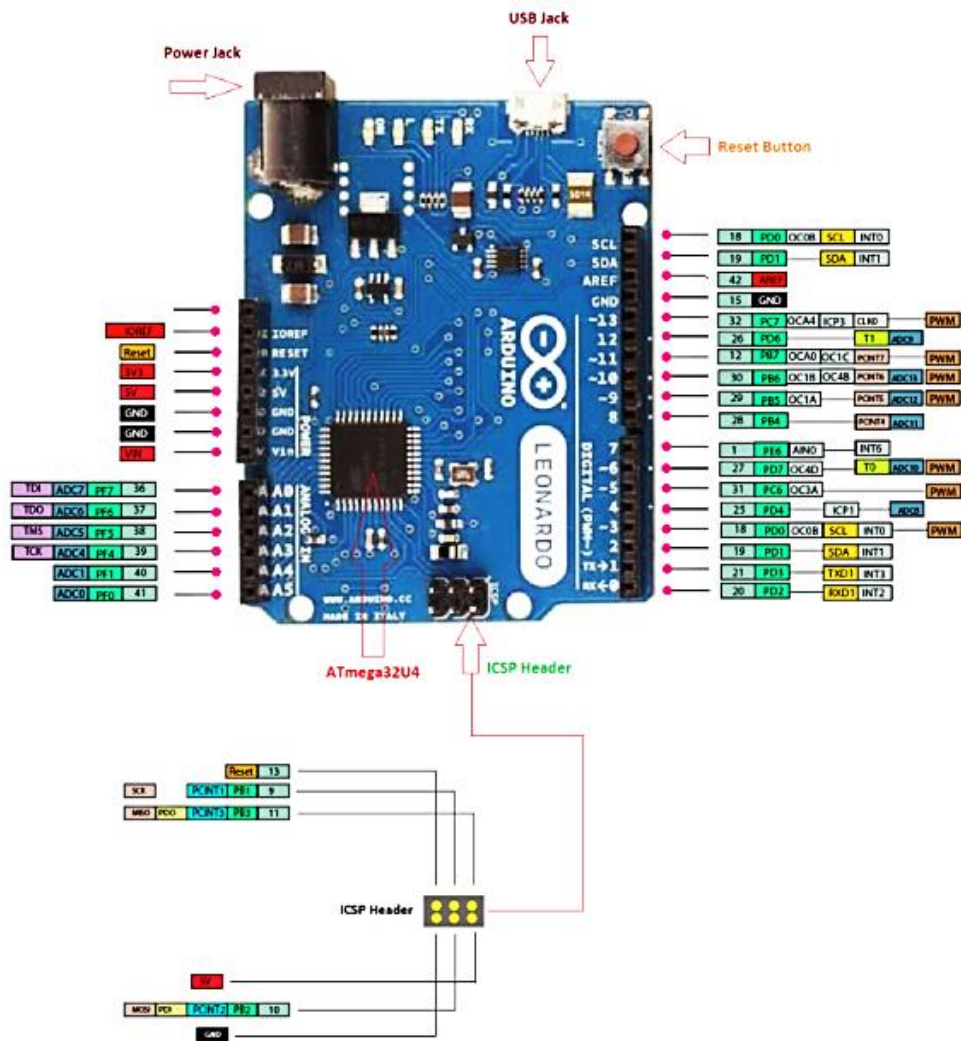


Figura 47. Esquema Arduino Leonardo (Aqeel, 2018)

El dispositivo Bluetooth está directamente conectado con el módulo Arduino, en este caso se programaron dos pines digitales para que funcionen como pines de comunicación serial, los pines 10 y 11, los cuales fueron configurados como RX y TX respectivamente. Y los pines de alimentación (Vcc y GND) fueron conectados en los pines 5 V y GND del módulo Arduino.



Figura 48. Módulo Bluetooth HC-06

Este dispositivo no puede recibir alimentación externa a la tarjeta debido a que podría ocasionarle daños. Se implementó el mismo modelo Bluetooth (HC-06) utilizado para la simulación en Proteus. Ver Figura 48.

Continuando con el diagrama de conexión del sistema nos encontramos con los circuitos optoacopladores, para los cuales también se ha diseñado una pista en baquelita de acuerdo al esquema del circuito realizado en Proteus para la conexión de los elementos que lo componen (resistencia 1K Ω , resistencia 4.7 K Ω , diodo y transistor 2N3904). Se pasó una capa de espray de color negro en la cara superior de este circuito. Véase la Figura 49.



Figura 49. Circuito optoacoplador

Los relés empleados para este circuito son los mismos que se designó en la simulación de Proteus, los relés 904, exclusivos para uso automotriz. Ver Figura 50. Una vez recibida las señales por parte del módulo Arduino estos permiten el envío de voltaje desde la batería de 12 V hasta las señales de

ignición y arranque del switch de encendido. Para lo cual se utilizaron soquets macho-hembra de dos cables o pines que permiten realizar esta conexión sin la necesidad de abrir el compartimento donde se encuentran las conexiones eléctricas del banco de pruebas.



Figura 50. Relé 904

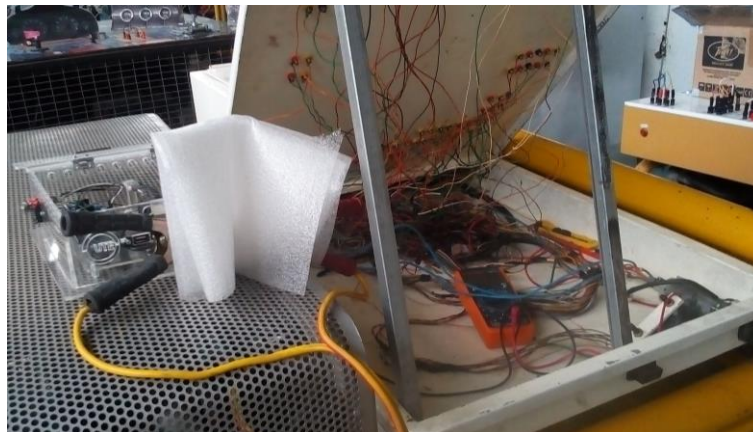


Figura 51. Instalación eléctrica del banco de pruebas



Figura 52. Conexión al switch de encendido en ignición y arranque

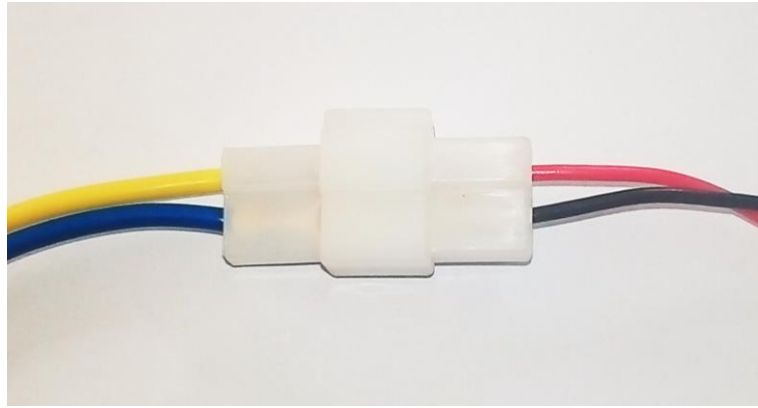


Figura 53. Soquets macho-hembra de dos pines

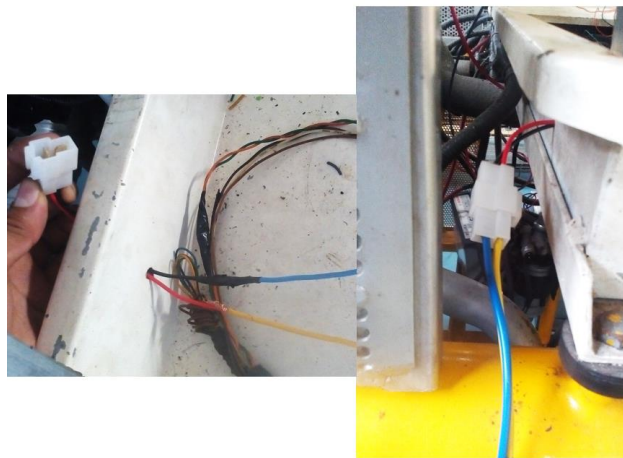


Figura 54. Soquets instalados en el banco de pruebas

Esta parte del circuito culmina con la conexión de los relés al switch de encendido del MCI del banco de pruebas.



Figura 55. Motor CRDI del banco de pruebas

Siguiendo con el esquema del circuito de control, se realizaron las conexiones de los módulos L298N en los correspondientes pines digitales de la tarjeta Arduino (2, 3, 4, 5, 6, 7, 8 y 9).

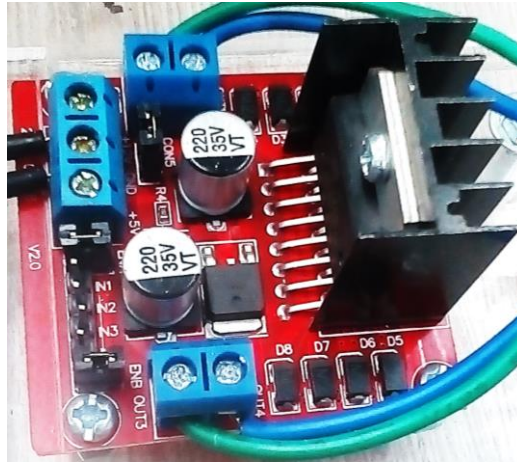


Figura 56. Módulo L298N

En la práctica no hace falta conectar alimentación en los dos pines de ingreso de voltaje de estos módulos, ya que a diferencia del simulador solo se conectará alimentación de 12 V al pin +12 V y tierra al pin GND, suficiente para el correcto funcionamiento y control de los actuadores.

Para realizar las conexiones entre módulos L298N desde el interior de la maqueta hacia los actuadores es decir, hacia exterior, se ha visto la necesidad de contar con los bornes más adecuados para esto, por lo que se ha implementado al sistema unos bornes de altavoces, los cuales permiten conectar cables del mismo número y grosor como los que tienen los actuadores para su conexión en el sistema de bloqueo central. Ver Figura 57.

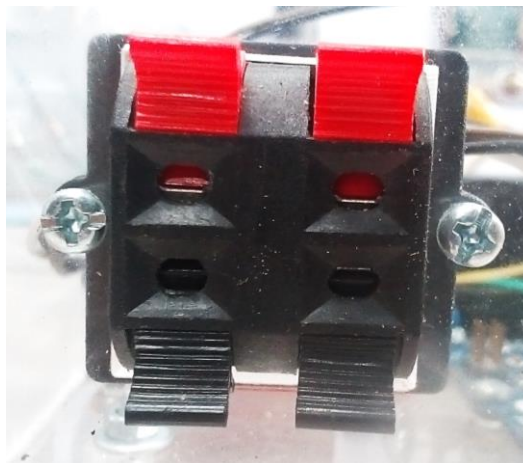


Figura 57. Bornes de altavoz

Los actuadores implementados en el sistema son de marca genérica ya que estos sirven perfectamente para la comprobación del correcto funcionamiento del sistema.

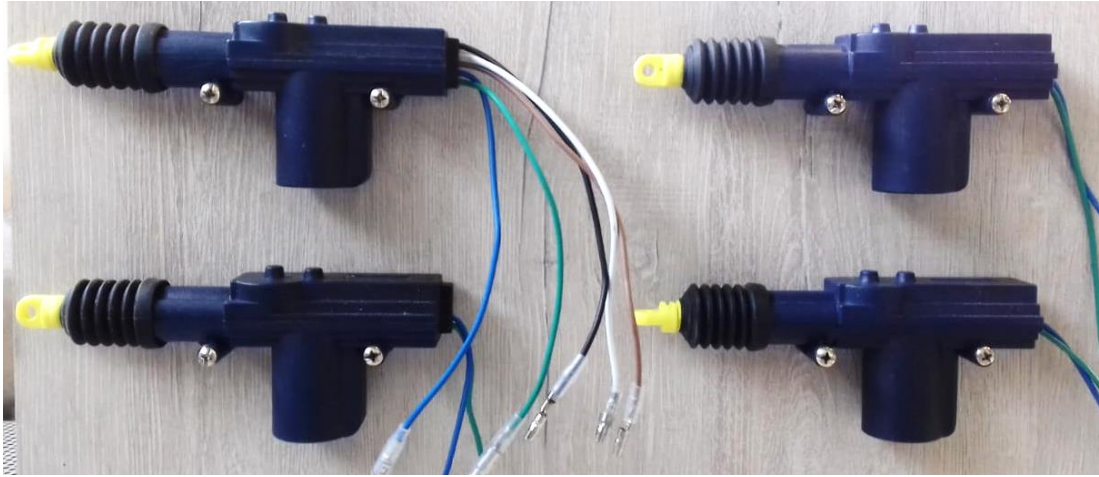


Figura 58. Conjunto de actuadores a implementar en el sistema

2.7.1. CUBIERTA Y SOPORTE

La cubierta tiene la función de proteger al sistema en un ambiente cerrado para impedir que polvo o humedad afecten al circuito de control, mientras que el soporte es la base en donde se apoyará la maqueta (circuito de control y actuadores del bloqueo central).

Se ha diseñado el modelo y dimensiones Adobe Ilustrador de la cubierta de acuerdo a la distribución de los dispositivos que conforman el circuito de control, diseño el cual fue plasmado en físico en una lámina de acrílico transparente de 4 cm de espesor la cual fue cortada a láser. En la Figura 59 se detalla las dimensiones de las 6 caras de la cubierta además de una superficie interna donde se apoyan todos los dispositivos. Las dimensiones son:

Caras A1 y A2 (tapa y base): 22 cm X 20 cm

Cara B (superficie o base interna): 17 cm X 19 cm

Caras C1 y C2 (laterales): 8cm X 20 cm

Caras D1 y D2 (frontal y posterior): 8 cm X 22 cm

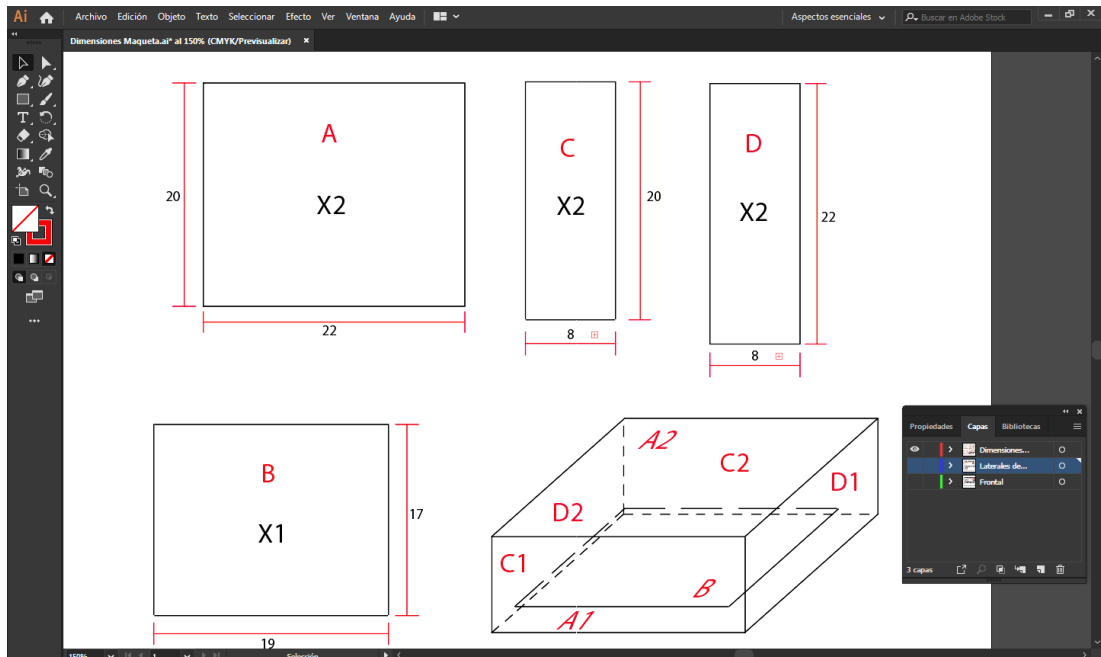


Figura 59. Alto, ancho y profundidad de la cubierta en AI

En la Figura 60 se detallan las dimensiones de las caras laterales:

C1:

Perforaciones para bornes (voltajes de salida): 3/16 pulg Ø

Perforación para borne de altavoz: 2,4 cm X 2,6 cm

Perforaciones de ventilación: 1 cm Ø

C2:

Perforación para borne de altavoz: 2,4 X 2,6 cm

Perforaciones de ventilación: 1 cm Ø

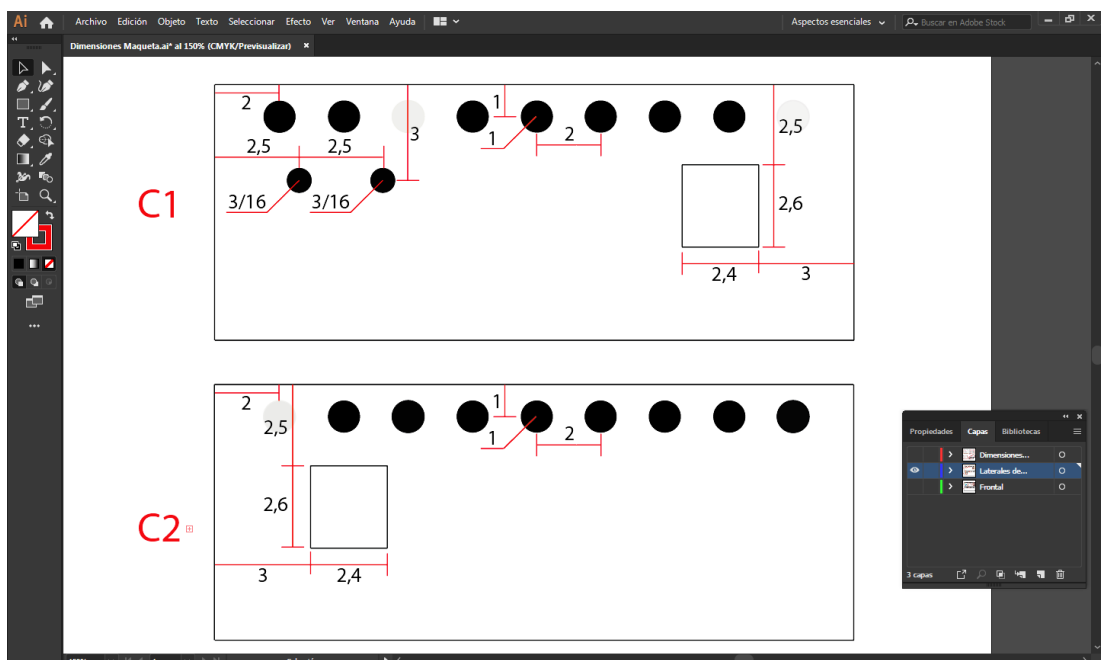


Figura 60. Dimensiones internas de las caras C1 y C2 en AI

En la Figura 61 se dan las dimensiones internas de las caras frontal y posterior:

D1:

Perforación para porta fusible: 1,4 cm Ø

Perforación para botón interruptor: 2,1 cm Ø

Perforaciones para bornes (+, -): 0,5 cm Ø

Perforación para puerto USB: 1,8 cm X 1 cm

Perforaciones de ventilación: 1 cm Ø

D2:

Perforaciones de ventilación: 1 cm Ø

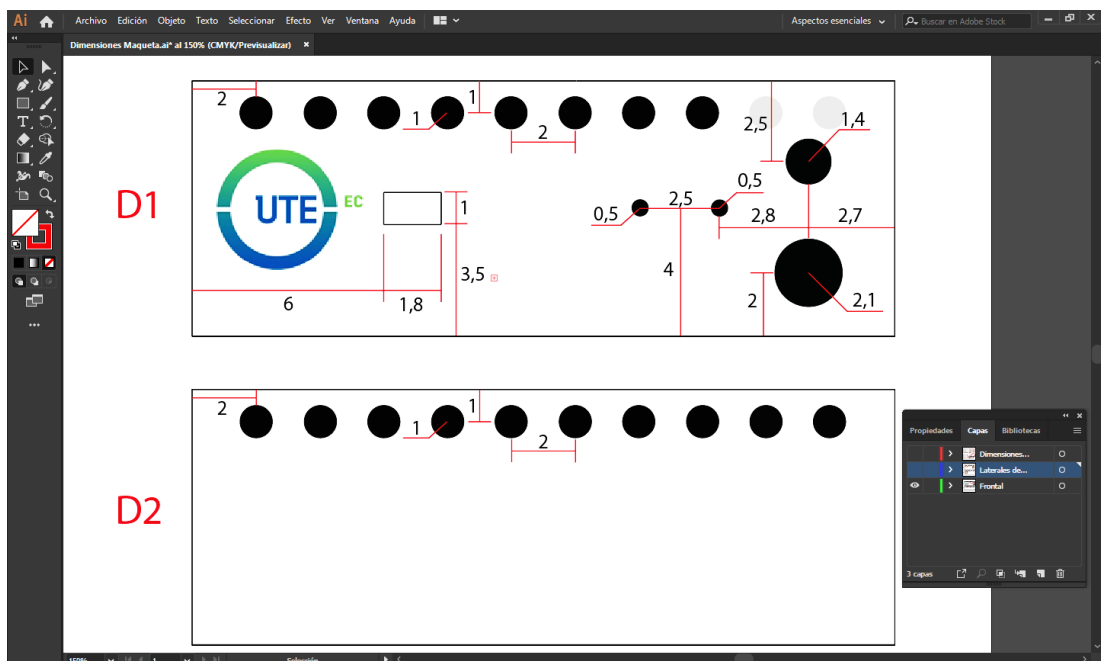


Figura 61. Dimensiones internas de las caras D1 y D2 en AI

El soporte básicamente está conformado por dos tabloncillos de madera artificial de aglomerado chapado de 15 mm de espesor que forman una "L" de 90° ya que la tabla en sentido paralelo a la base es donde se apoya la maqueta del circuito de control, mientras que la tabla perpendicular a la base sostiene los actuadores del bloqueo central. Ver Figura 62.

A (soporte base): 44 cm X 43 cm

B (soporte de actuadores): 44 cm X 30 cm

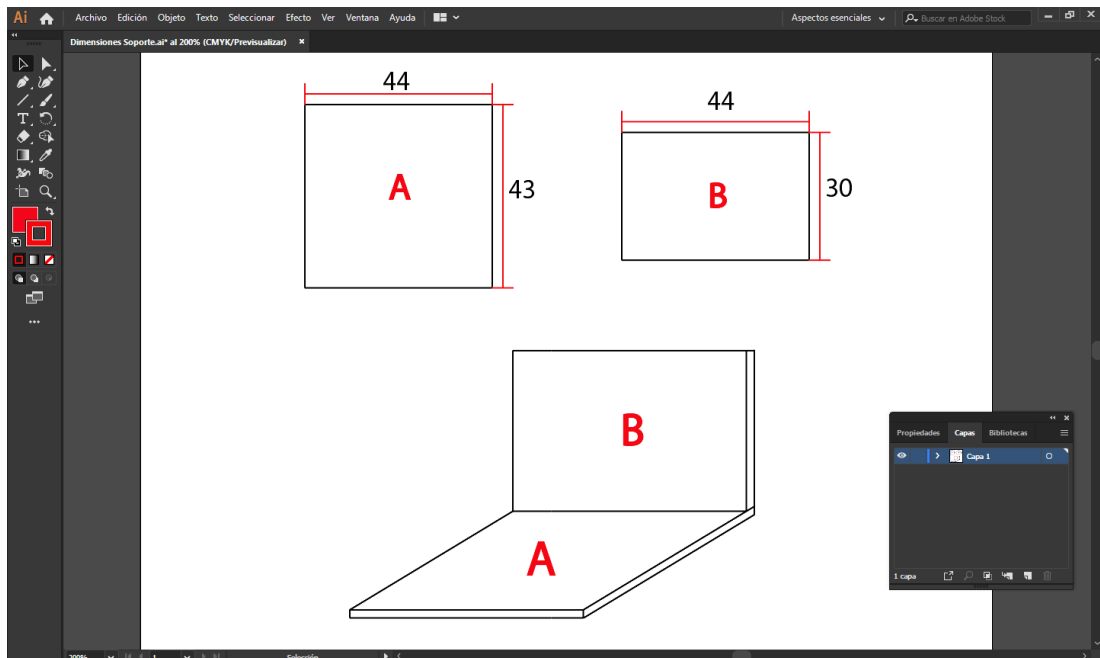


Figura 62. Ancho y largo de las tablas de soporte en AI

2.8. DISEÑO Y PROGRAMACIÓN DE LA APLICACIÓN DE CELULAR

Para la creación del diseño y la programación de la aplicación para teléfonos móvil con sistema operativo Android se utilizó el software online App Inventor, donde se estableció que la aplicación permitiera realizar funciones tales como: conectarse y desconectarse del módulo Bluetooth HC-06, comandar por medio de botones en un “Panel de Control” el encendido del banco de pruebas, el arranque y apagado del MCI como también el accionamiento de los actuadores, para lo cual se ha dividido esta sección en dos partes:

- Diseño de la aplicación
- Programación de la aplicación

2.8.1. DISEÑO DE LA APLICACIÓN

Se optó por un diseño sencillo, agradable a la vista y con una operabilidad fácil de aprender. La aplicación cuenta con tres ventanas o “Screens” (no cuentan con una numeración en orden), cada una cumple una función en específico:

Screen “1”: Muestra una intro de la aplicación la cual tiene como fondo de pantalla un logo de la UTE y un logo de la facultad de Ciencias de la

Ingeniería, tal como se indica en la Figura 63. Esta intro dura 1.8 segundos antes de pasar a la Screen 3.

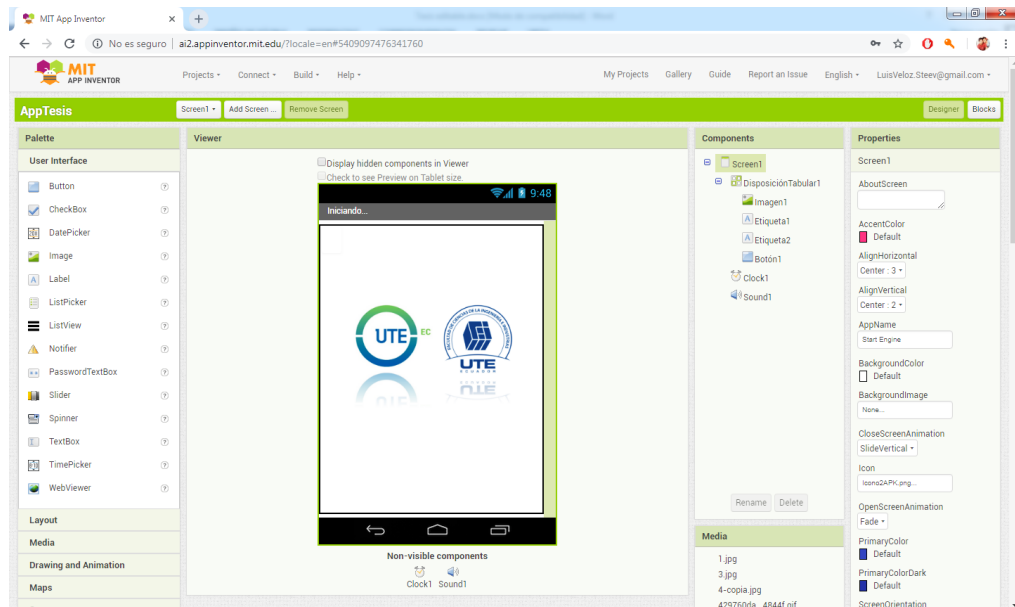


Figura 63. Screen “1” de la aplicación de celular

Screen “3”: En esta ventana se encuentra el “Panel de mando” el cual nos permite manipular el encendido del MCI y el sistema de bloqueo central por medio del circuito de control una vez que se establezca la comunicación vía Bluetooth entre móvil y el dispositivo HC-06, tal como se indica en la Figura 64.

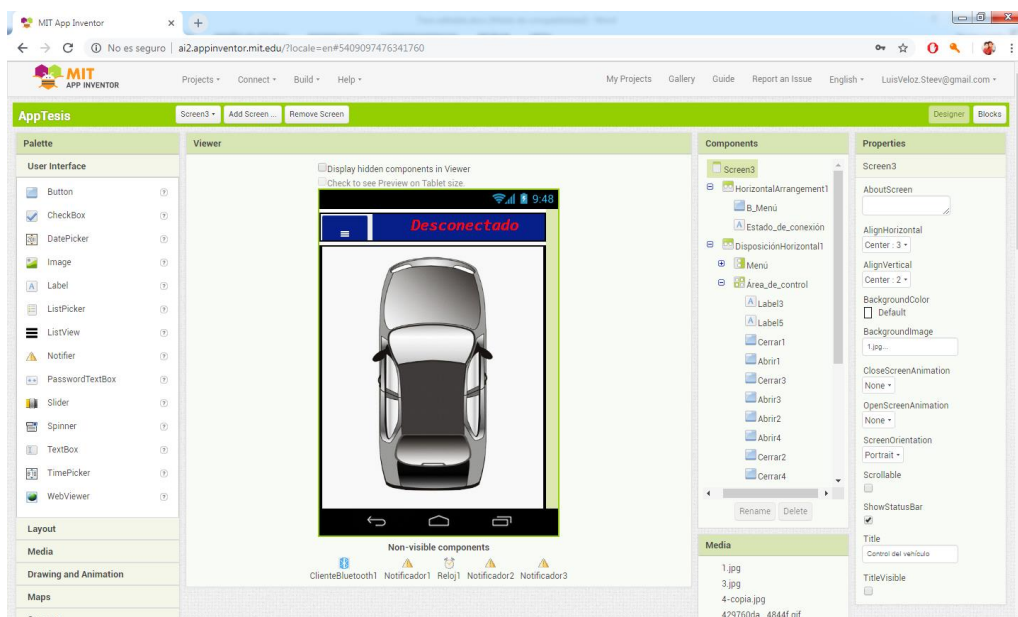


Figura 64. Screen “3” de la aplicación de celular

Screen “4”: En esta ventana se muestran los créditos con datos del proyecto como autor y tutor de tesis, además se ilustra el logo de la institución. Ver Figura 65.

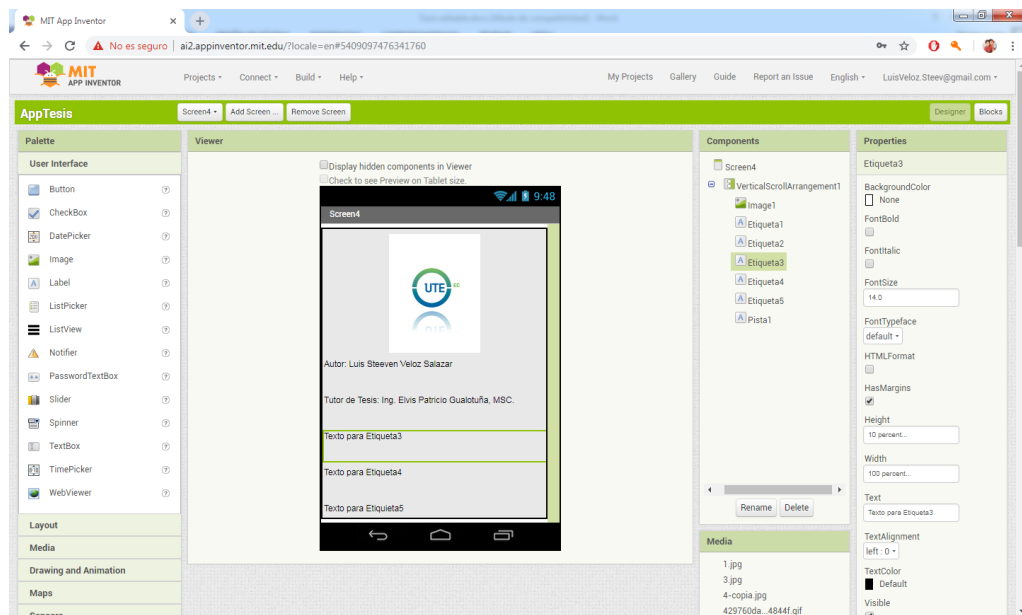


Figura 65. Screen “4” de la aplicación de celular

Para el ícono identificativo se decidió realizar el diseño en el software Adobe Illustrator para lo cual se optó por colocar en un fondo azul el cual hace referencia al campo de la tecnología, el cual se muestra en la Figura 66.



Figura 66. Fondo para el icono de la aplicación (Keo, 2018)

Se sobrepuso delante del fondo una imagen del vehículo al que corresponde el MCI que tiene el banco de pruebas, el Sportage Diesel 2010, como se muestra en la Figura 67.



Figura 67. Foto de Sportage Diésel 2010 para el ícono (Microsoft, 2010)

Se añadió el símbolo de Bluetooth en la imagen indicando así que esta aplicación utiliza este método de comunicación inalámbrica para su funcionamiento. Símbolo en la Figura 68.



Figura 68. Símbolo identificativo de Bluetooth para el ícono (Raya, 2015)

El diseño final del ícono para la aplicación se aprecia en la Figura 69.



Figura 69. Diseño final del ícono de la aplicación

Se decidió poner el nombre de “Start Engine” a la aplicación por su significado en español “encender el motor” o “arrancar el motor” y por su corta escritura la cual se aprecia muy bien en un celular cuando es instalada.

2.8.2. PROGRAMACIÓN DE LA APLICACIÓN

Cada ventana cuenta con un bloque de programación el cual permite establecer condiciones y variables que dirijan el comportamiento de la aplicación según lo deseado, en este caso comunicarse con el módulo Bluetooth HC-06 y comandar el encendido de un MCI y la apertura y cierre de actuadores de un bloqueo central. Este software cuenta con un lenguaje de programación básico fácil de aprender y entender, permitiendo establecer estas variables y condiciones de manera muy didáctica por medio de piezas que pueden ser ordenadas de acuerdo a la programación necesitada.

Bloque Screen “1”: Es la ventana inicial de la aplicación donde se ha establecido que sea reproducido un sonido y que ésta realice una transición a la Screen “3” en un lapso de tiempo de 1.8 segundos ya que fue insertado un cronómetro que permite cumplir esta condición. Si se intenta retroceder la aplicación cuando se está ejecutando esta ventana, se ha establecido que se cierre la misma. En la Figura 70. Se representa esta programación.

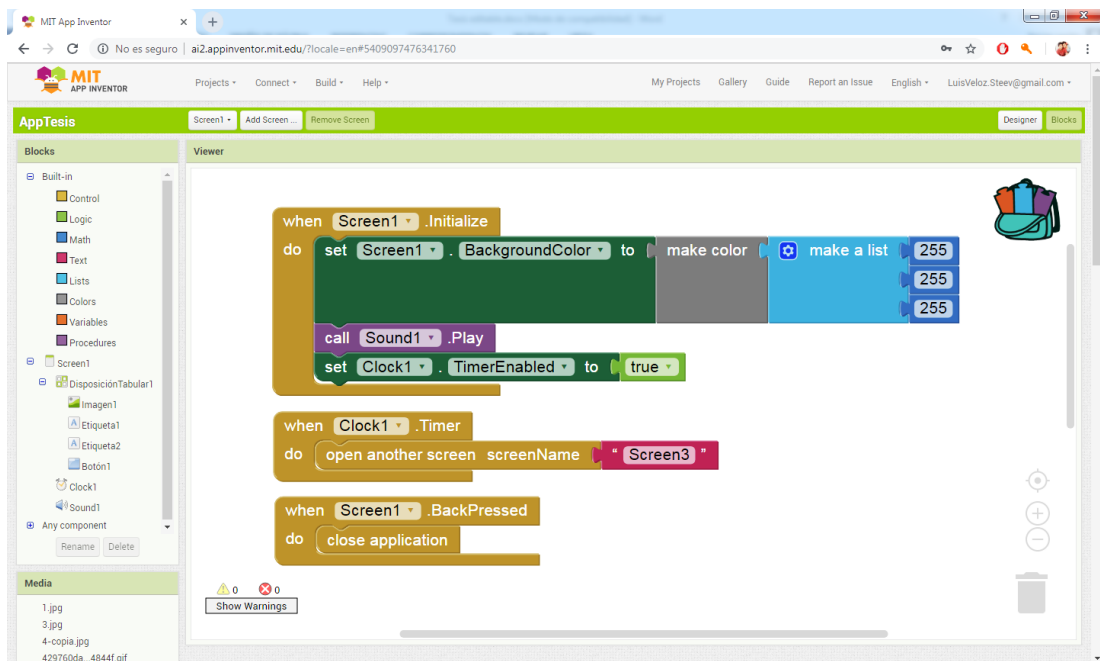


Figura 70. Bloque de programación Screen “1”

Bloque Screen “3”: En este bloque determinó que por defecto se muestre un mensaje de “Desconectado” con color de letra en rojo, una vez establecida la conexión con el módulo Bluetooth llamado “KIA Sportage 2010” (nombre asignado por un programa en Arduino) este mensaje cambia a “Conectado” con color de letra en verde, además se habilitan los botones de control en el área del “Panel de control” de la ventana. Para hacerlo se debe ingresar en un menú plegable cuyo botón de accionamiento se encuentra en la parte superior izquierda de la ventana en la barra de color

azul. Este menú fue programado para que esté oculto por defecto. La programación de este menú indica su velocidad de desplazamiento y su estado, el cual ordena al menú desplegarse cuando esté oculto y ocultarse cuando esté desplegado. El bloque de programación de esta ventana se indica en las Figuras 71, 72, 73 y 74.

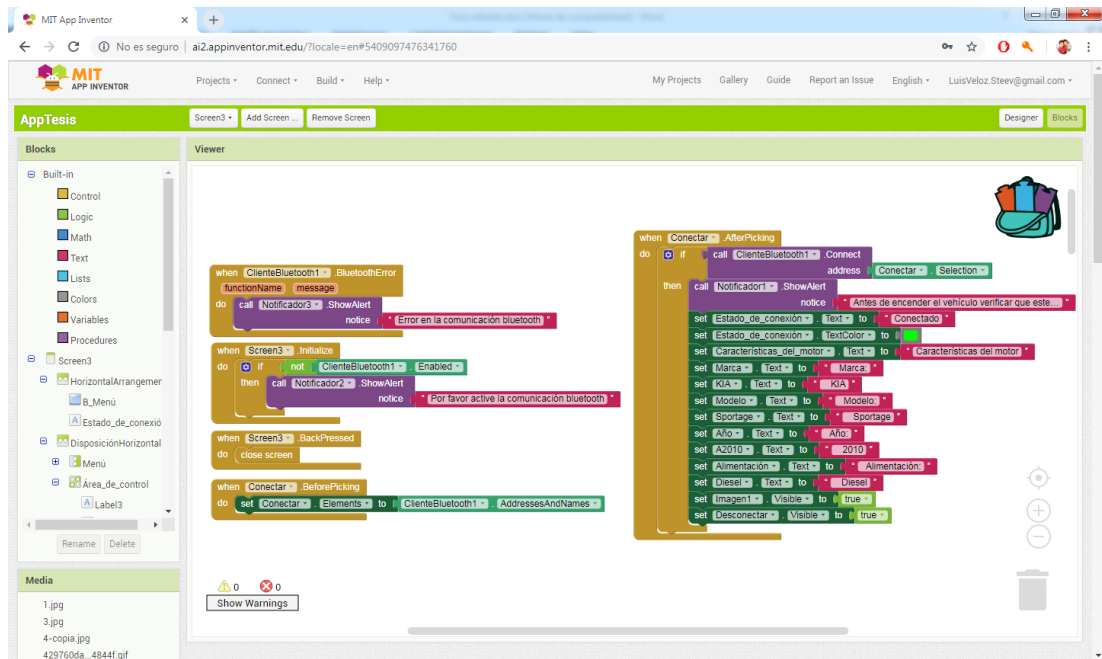


Figura 71. Primera parte del bloque de programación Screen “3”

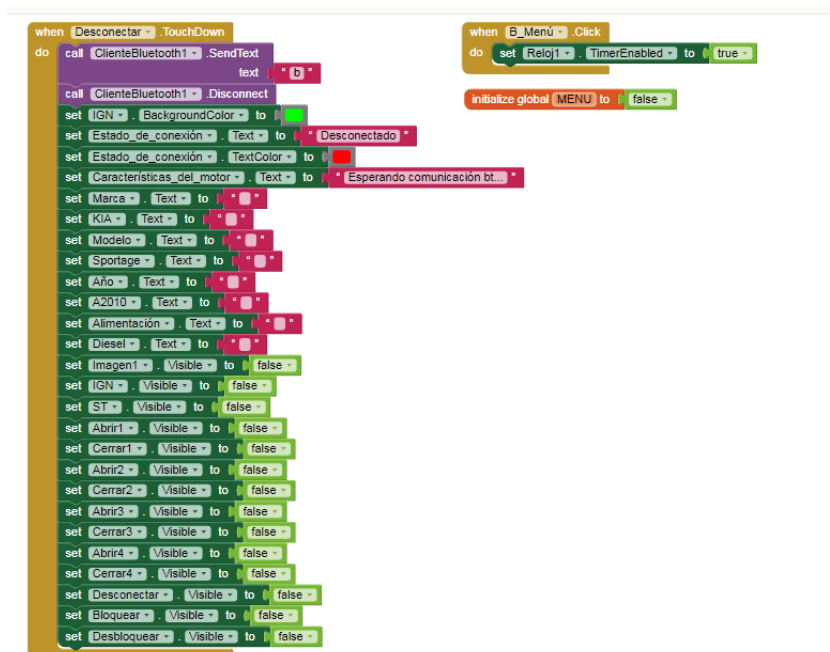


Figura 72. Segunda parte del bloque de programación Screen “3”

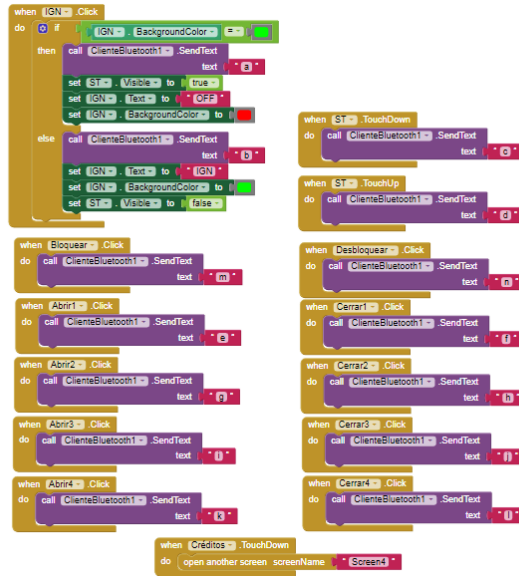


Figura 74. Cuarta parte del bloque de programación Screen “3”

Bloque Screen “4”: Como ya se explicó, esta ventana contiene los créditos o nombres de quienes participaron en el proyecto, para lo cual solo se vió la necesidad de programar el regreso a la Screen “3” en caso de ser presionado el botón de retroceso del móvil, como se ilustra en la Figura 75.

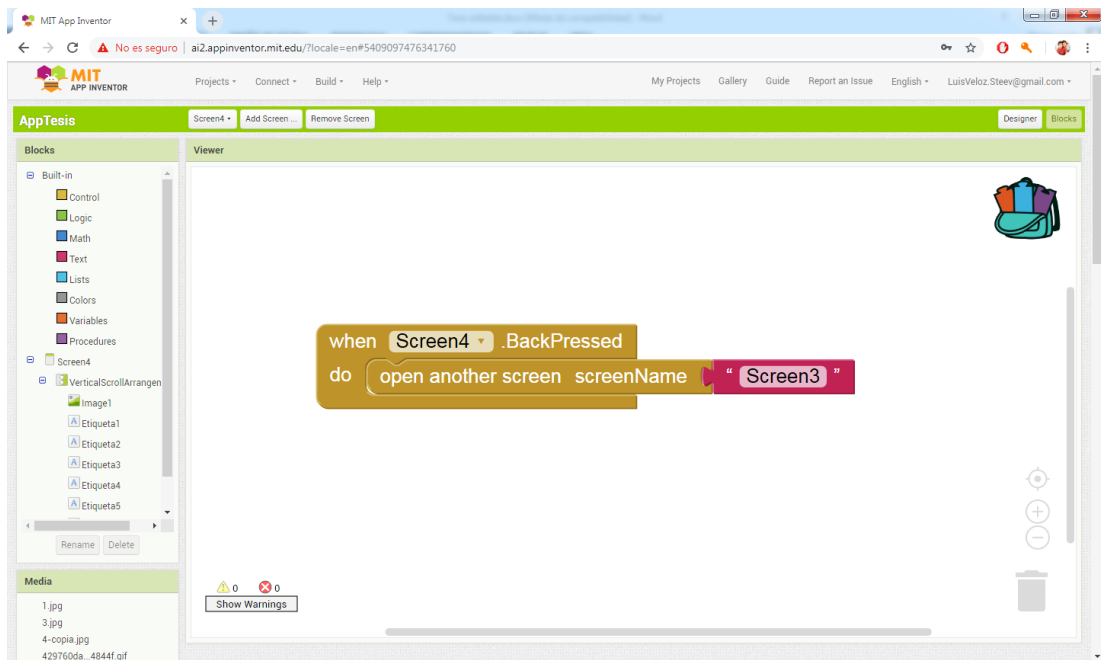


Figura 75. Bloque de programación Screen “4”

2.9. PRUEBAS DE MEDICIÓN EN LA PRÁCTICA

Se verificó el correcto funcionamiento del circuito de control mediante la comprobación de mediciones de voltajes, amperajes y ondas de señal entre Simulación-Práctica. Se aplicó un método analítico para comprobar que el sistema cuenta con la fuente de alimentación necesaria para funcionar correctamente y suministrar la energía necesaria para el encendido del MCI y el control de los actuadores del bloqueo central. Estos resultados se comprobaron y compararon con instrumentos de medición de uso automotriz, el multímetro y el osciloscopio.



Figura 76. Multímetro automotriz utilizado en las pruebas de medición



Figura 77. Osciloscopio automotriz utilizado en las pruebas de medición

Para estas comparaciones se dividieron los resultados obtenidos en secciones de voltajes, amperajes y ondas con el fin de establecer un orden detallando los valores comparados en tablas.

2.9.1. VOLTAJES (VOLTÍMETRO)

Los voltajes de alimentación medidos en el simulador de Proteus y en la práctica con multímetro automotriz de cada uno de los dispositivos se detallan a continuación:

Tabla 4. Comparación Simulación-Práctica de fuente 12V

Alimentación del sistema:		Observación:
Simulación	12 V	La diferencia entre los datos obtenidos es mínima.
Práctica	12.71 V	

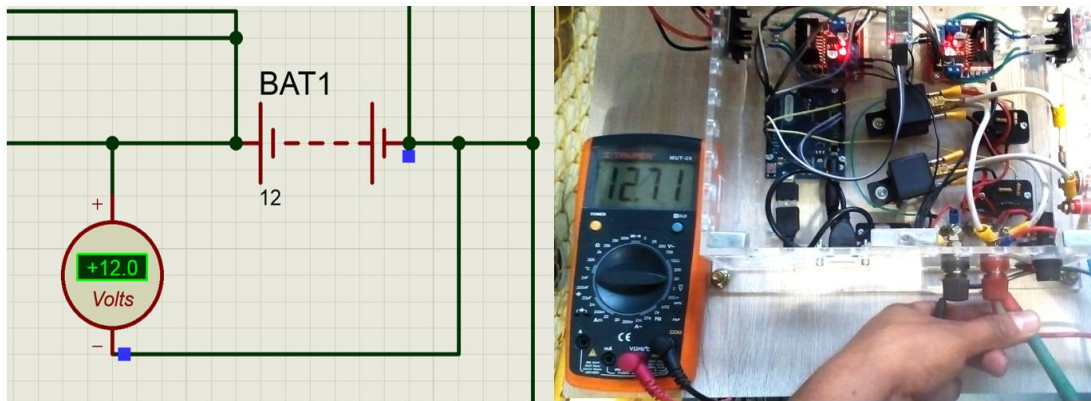


Figura 78. Comparación Simulación-Práctica de fuente 12V

Tabla 5. Comparación Simulación-Práctica de fuente Arduino

Alimentación del módulo Arduino:		Observación:
Simulación	6 V	Existe una diferencia mínima entre los dos valores obtenidos.
Práctica	5.92 V	

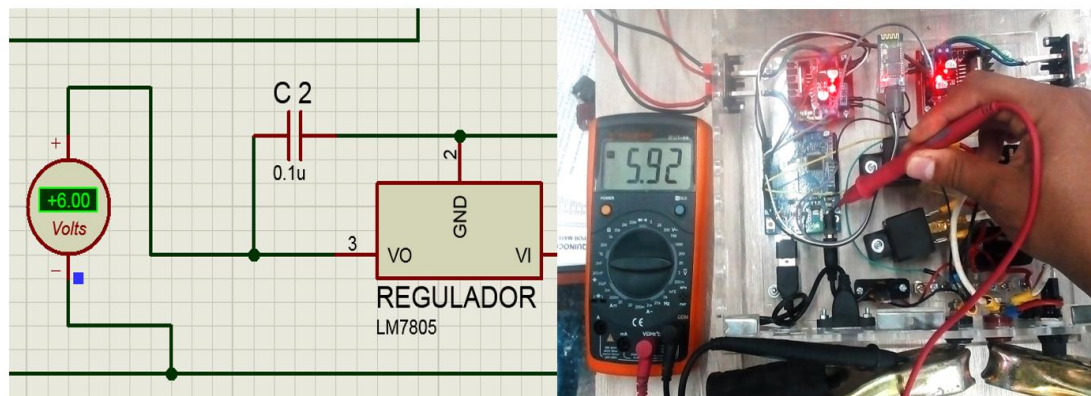


Figura 79. Comparación Simulación-Práctica de fuente Arduino

Tabla 6. Comparación Simulación-Práctica de señal emitida por Arduino

Señal emitida por el módulo Arduino para la activación de los relés:		Observación:
Simulación	4.92 V	La diferencia entre ambos valores es mínima.
Práctica	4.84 V	

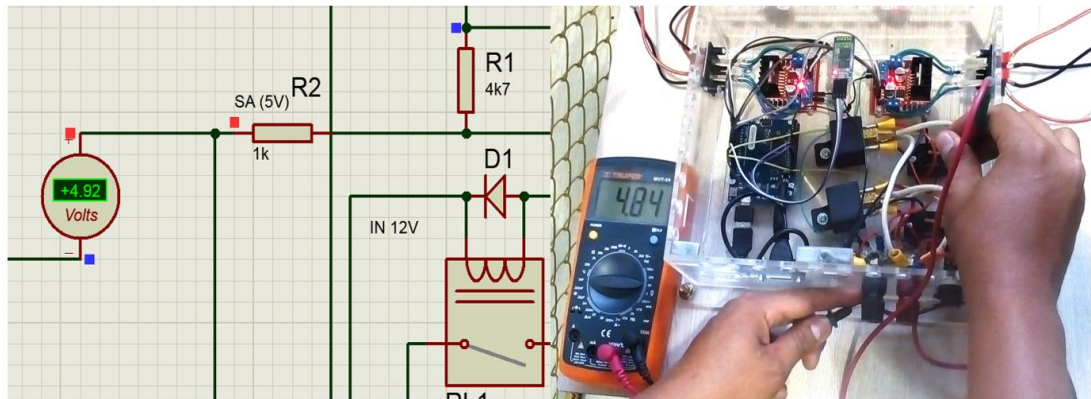


Figura 80. Comparación Simulación-Práctica de señal emitida por Arduino

Los voltajes de las señales de respuesta por parte de los relés de mando y los módulos L298N obtenidos en el simulador de Proteus y en la práctica con multímetro automotriz se detallan a continuación:

Tabla 7. Comparación Simulación-Práctica de señal de relés

Señal emitida por los relés de mando:		Observación:
Simulación	11.9 V	Existe una diferencia de 1 V debido a la caída de tensión por materiales empleados.
Práctica	10.99 V	

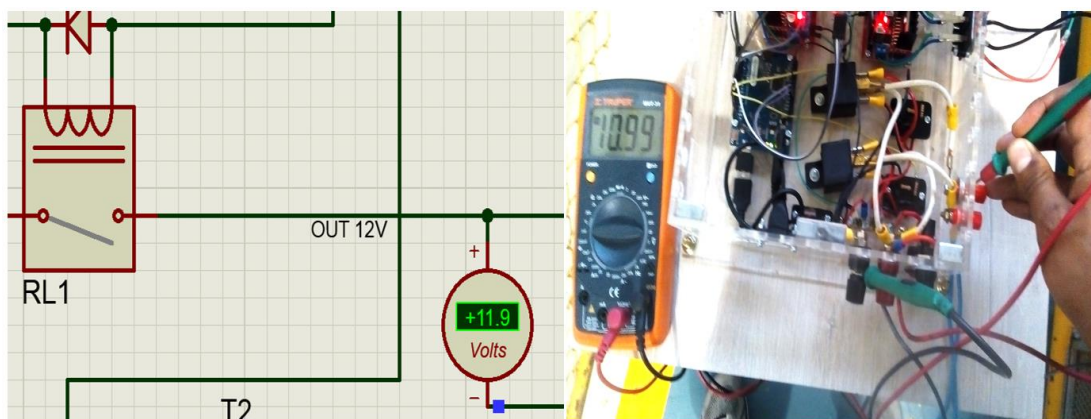


Figura 81. Comparación Simulación-Práctica de señal de relés

Tabla 8. Comparación Simulación-Práctica de señal de módulos L298N

Señal emitida por los módulos L298N:		Observación:
Simulación	11.1 V	Existe diferencia entre valores debido a que en la simulación es requerida alimentación en VCC y VS de los L298N.
Práctica	6.06 V	

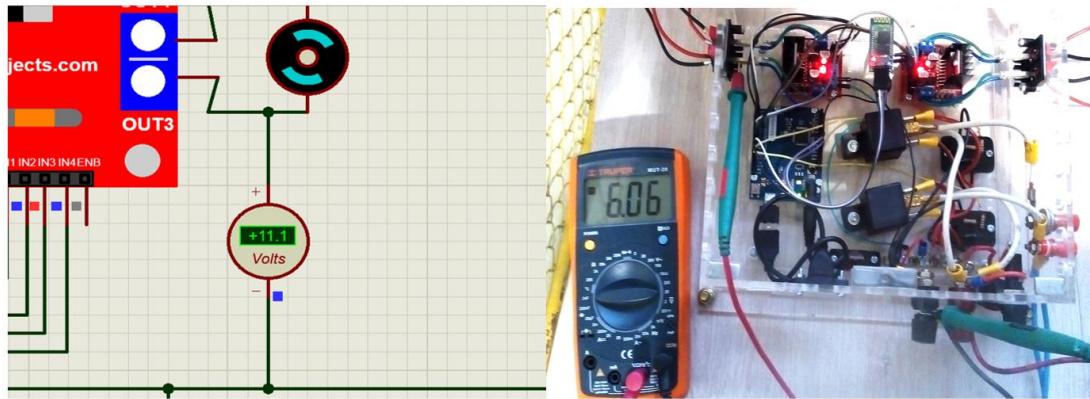


Figura 82. Comparación Simulación-Práctica de señal de módulos L298N

2.9.2. AMPERAJES (AMPERÍMETRO)

La corriente medida en el simulador de Proteus y en la práctica con multímetro automotriz de la fuente de alimentación se detalla a continuación:

Tabla 9. Comparación Simulación-Práctica de amperaje de fuente

Amperaje de la fuente de alimentación:		Observación:
Simulación	3.36 A	Existe diferencia de amperajes debido al material de composición de la barra metálica.
Práctica	0.14 A	

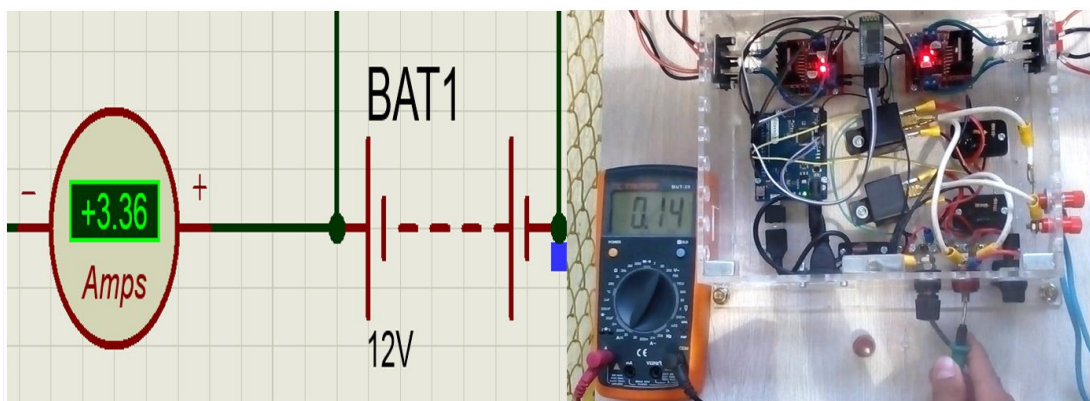


Figura 83. Comparación Simulación-Práctica de amperaje de fuente

La corriente medida en el simulador de Proteus y en la práctica con multímetro automotriz en las señales de respuesta por parte de los relés de mando se detalla a continuación:

Tabla 10. Comparación Simulación-Práctica de amperaje de relés

Amperaje de señal emitida por los relés de mando:		Observación:
Simulación	0.99 A	Existe diferencia debido a materiales empleados en los bornes de conexión.
Práctica	0.14 A	

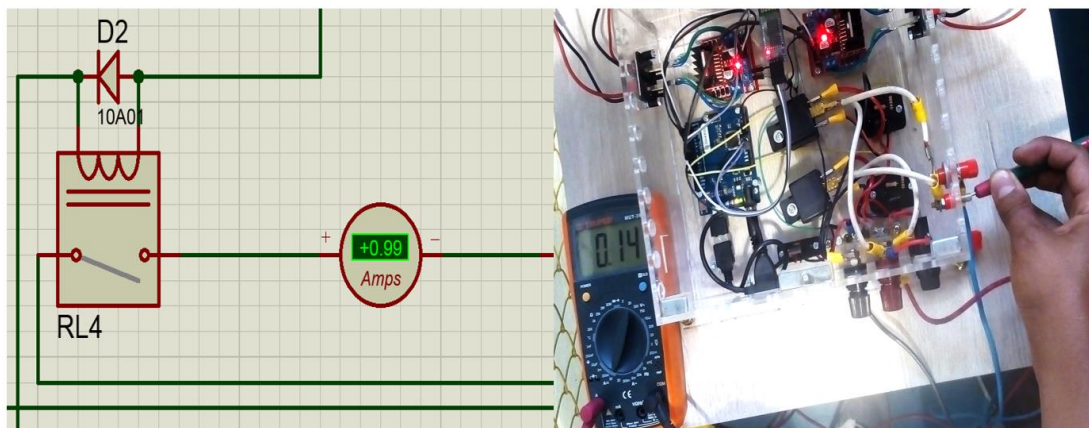


Figura 84. Comparación Simulación-Práctica de amperaje de relés

2.9.3. ONDAS DIGITALES (OSCILOSCOPIO)

Las señales obtenidas mediante osciloscopio tanto en la simulación de Proteus como en la práctica se representan de la siguiente manera:

Onda de señal digital emitida por el pulso del pin 13 al relé de Ignición durante 5 segundos:



Figura 85. Comparación Simulación-Práctica onda de señal digital pin 13

Onda de señal digital emitida por el pulso del pin 12 al relé de Start durante 2 segundos:

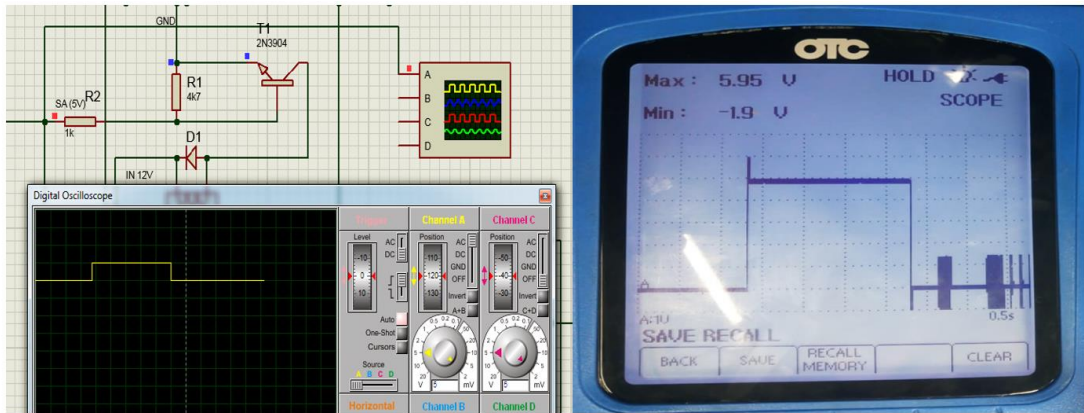


Figura 86. Comparación Simulación-Práctica onda de señal digital pin 12

3. RESULTADOS Y DISCUSIÓN

3.1. PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA

Estas pruebas comprobaron el correcto funcionamiento del sistema evaluando:

- Funcionamiento de la aplicación
- Conexión vía Bluetooth
- Encendido y apagado del MCI
- Accionamiento de los actuadores del bloqueo central

El sistema se conectó a la fuente de 12 V, y se procedió a presionar el botón interruptor para que éste empiece su funcionamiento.



Figura 87. Conexión del sistema a la fuente 12V para las pruebas

De acuerdo al diagrama de conexión debemos comunicar las señales emitidas por los relés del circuito de control al switch de encendido del banco de pruebas, para lo cual utilizamos dos cables flexibles cuyos extremos están conectados a un soquet. De igual manera se respetan los colores para el conectado, el cable con marca de color rojo se conecta al borne rojo o borne con rodela (pin 12), y el cable con marca de color negro se conecta al borne negro (pin 13). Una vez realizadas las conexiones al circuito de control se conecta el soquet al que se encuentra en el lateral izquierdo del panel de conexiones del banco de pruebas.

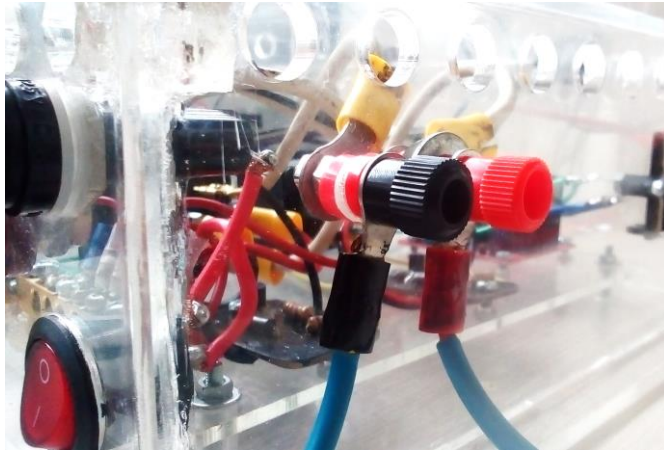


Figura 88. Conexión del sistema al banco de pruebas



Figura 89. Conexión entre circuito de control y banco de pruebas

Se debe revisar que las conexiones estén correctamente realizadas para proceder a encender el sistema, para lo cual es necesario solamente presionar el botón interruptor del circuito de control permitiendo de esta manera que la alimentación de 12 V encienda todos los dispositivos internos, los cuales disponen de leds que indican su encendido.

3.1.1. FUNCIONAMIENTO DE LA APLICACIÓN

Se pudo comprobar que la aplicación de celular inició sin ningún problema y como lo esperado ya que se ejecutaron las ventanas según lo programado. Esta prueba se la realizó en tres tipos de celulares distintos con sistema operativo Android. Ver Figura 90.

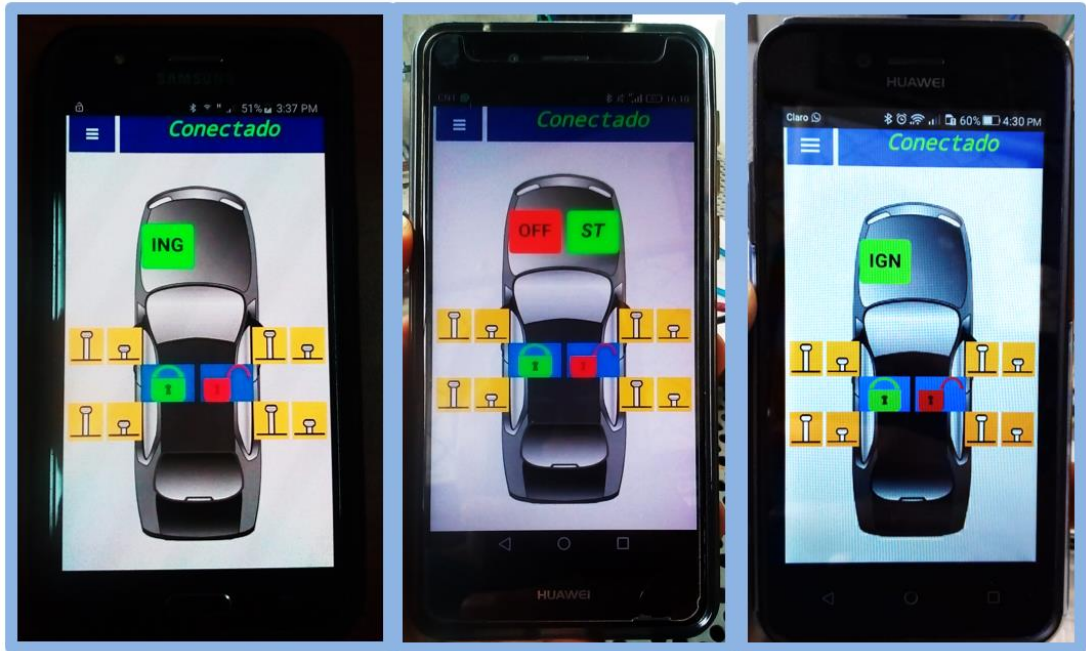


Figura 90. Funcionamiento de la aplicación en tres modelos de celulares

3.1.2. CONEXIÓN VÍA BLUETOOTH

Para esta prueba primero se enlazó el dispositivo Bluetooth con el móvil desde su configuración. Donde se puede encontrar al módulo HC-06, el cual ha sido modificado su nombre a “KIA Sportage 2010” mediante un programa cargado al módulo Arduino. Cuando se establece por primera vez la conexión Bluetooth se debe ingresar una contraseña de cuatro dígitos la cual puede ser modificada mediante un programa en Arduino. En este caso la contraseña ha sido modificada a “1995”. Una vez aceptada la contraseña el dispositivo Bluetooth éste se agrega a la lista de dispositivos conocidos. Ver Figura 91.

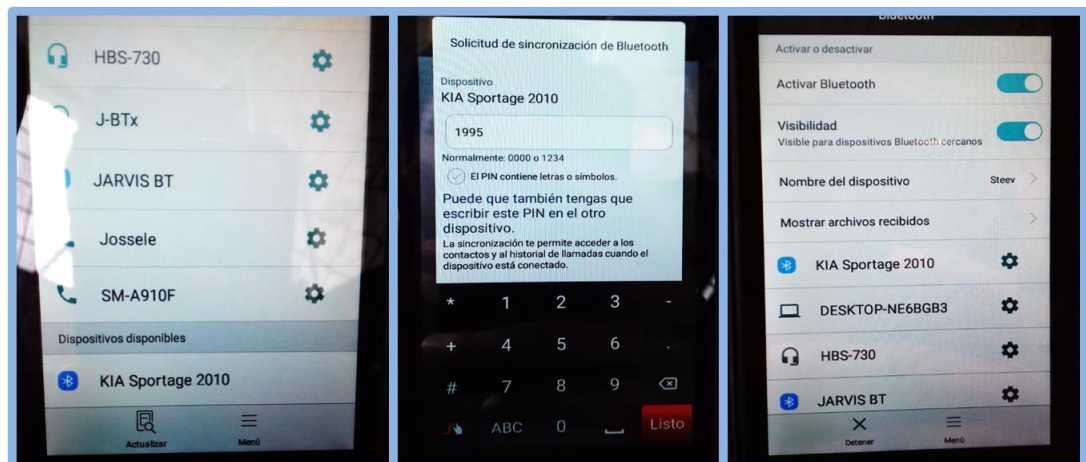


Figura 91. Enlazando móvil a dispositivo Bluetooth

Una vez establecida la conexión entre móvil y el dispositivo HC-06 debemos ingresar a la aplicación la cuál una vez instalada como cualquier otra se identifica con el logo característico y el nombre de “Start Engine”, como se indica en la Figura 92. Se recomienda utilizar móviles con versiones superiores a la v5 de Android para el correcto funcionamiento de la aplicación.

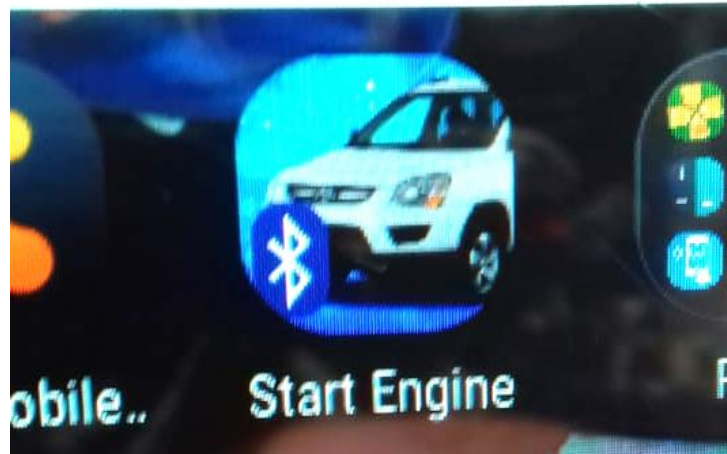


Figura 92. Ícono identificativo de la aplicación

Ingresando a la aplicación nos aparece la pantalla de inicio. Ver Figura 93.

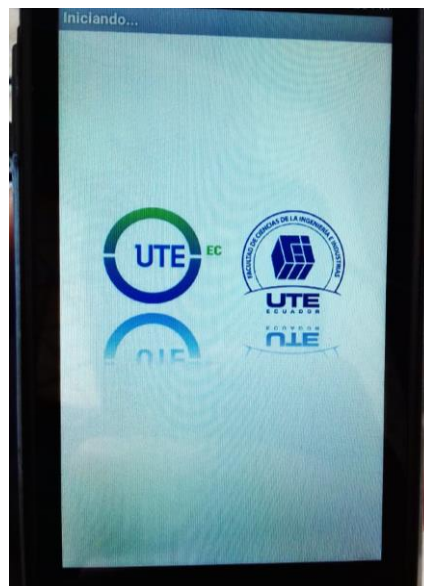


Figura 93. Pantalla de inicio de la aplicación

Como ya se mencionó, antes de ingresar a la aplicación de celular para intentar establecer conexión desde allí, se debe encender la comunicación Bluetooth en el celular o móvil, caso contrario se muestra un mensaje en la aplicación indicando que el operario debe hacerlo, como se indica en la Figura 94.

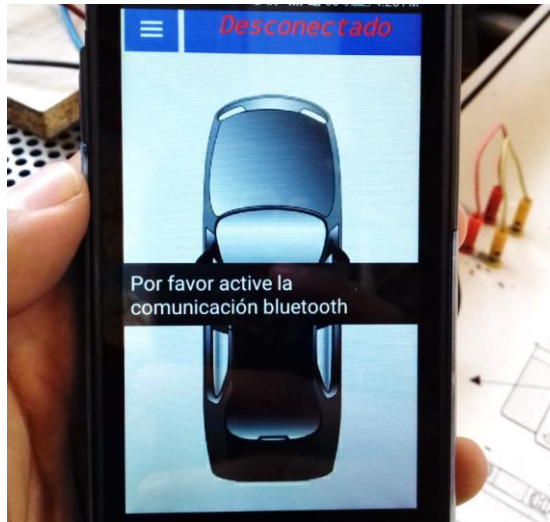


Figura 94. Mensaje de advertencia de comunicación Bluetooth

Para establecer conexión entre aplicación y módulo HC-06 debemos ingresar al menú plegable mediante el ícono de menú ubicado en la parte superior izquierda, el menú se despliega desde el lado izquierdo de la pantalla hacia la derecha, allí se encuentra ubicado el botón “Establecer conexión” el cual nos permite acceder a la lista de dispositivos Bluetooth agregados con la aplicación, luego seleccionamos el de nombre “KIA Sportage 2010”, y esperamos a que este conecte con el módulo Bluetooth. En algunos casos puede tardar entre 15 o 20 segundos. Ver Figura 95.

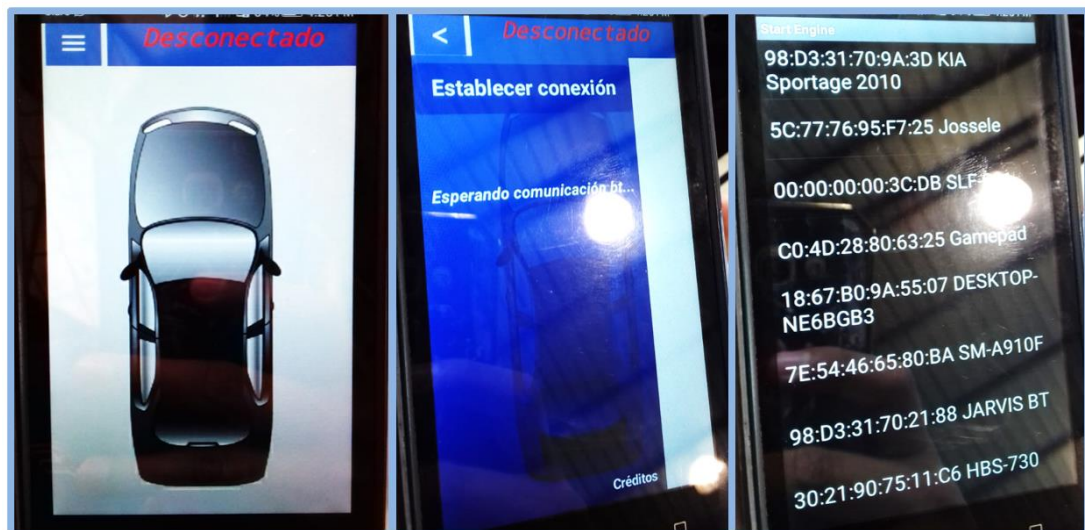


Figura 95. Enlazando aplicación a dispositivo Bluetooth

Cuando se establece conexión con “KIA Sportage 2010” la aplicación pasa del estado “Desconectado” a “Conectado” lo cual indica que se puede hacer uso de los botones antes ocultos del panel de mando o control que está ubicado por encima del fondo de pantalla (gráfico de un vehículo visto desde arriba). Se nos muestra un mensaje de advertencia indicando que el

vehículo debe encontrarse en Neutro en caso de ser instalado este sistema en un vehículo real. De esta manera se comprueba que la comunicación Bluetooth entre dispositivos se logra con éxito.



Figura 96. Conexión establecida entre aplicación y dispositivo Bluetooth

3.1.3. ENCENDIDO Y APAGADO DEL MCI

Una vez enlazada la aplicación con el sistema y verificando que todo esté debidamente conectado, se procedió a realizar las pruebas de encendido y apagado del motor. Para comenzar con la prueba de funcionamiento se presiona el botón de IGNICIÓN en la aplicación, lo que ordena al sistema dirigir la señal de alimentación de 12 V al cable de ignición del switch de encendido, permitiendo encender el banco de pruebas como si se lo hiciera con la llave. Esto habilita el botón de START en la aplicación antes oculto y el botón de IGNICIÓN en verde cambiará a OFF con un color rojo. Tal como se muestra en la Figura 97.



Figura 97. Encendido del banco de pruebas por medio de la aplicación

Luego se procede a encender el motor por medio del botón START simulando el arranque que se da con la llave para el encendido del motor, lo cual debe realizarse en un lapso de tiempo de 1-2 segundos. En las pruebas realizadas esto fue comprobado de manera eficiente.

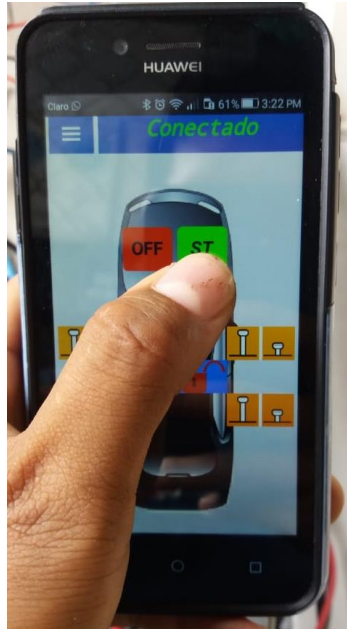


Figura 98. Encendido del MCI por medio de la aplicación

Para apagar el motor se debe presionar el botón de OFF cortando así el suministro de energía de 12 V hacia el cable de arranque del switch de encendido.



Figura 99. Apagado del MCI por medio de la aplicación

De esta manera fue comprobada la efectividad del sistema en cuanto al encendido y apagado del MCI del banco de pruebas.

3.1.4. CONTROL DE ACTUADORES

Para el control de la activación de los motores del bloqueo central se dispone de botones en el panel de control los cuales indican que puerta o, en este caso, que motor controlan y que acción realizarán, apertura o cierre, dependiendo de la polaridad empleada en la conexión de los actuadores al sistema.



Figura 100. Control individual de actuadores por medio de la aplicación

También se dispone de dos botones con símbolos de apertura y cierre por medio de un candado abierto en uno y un candado cerrado en otro, los cuales permiten el comando simultáneo de los cuatro actuadores. En las pruebas realizadas se verificó que este sistema cumple de manera eficaz tal y como lo hace un sistema de bloqueo central en un vehículo real.

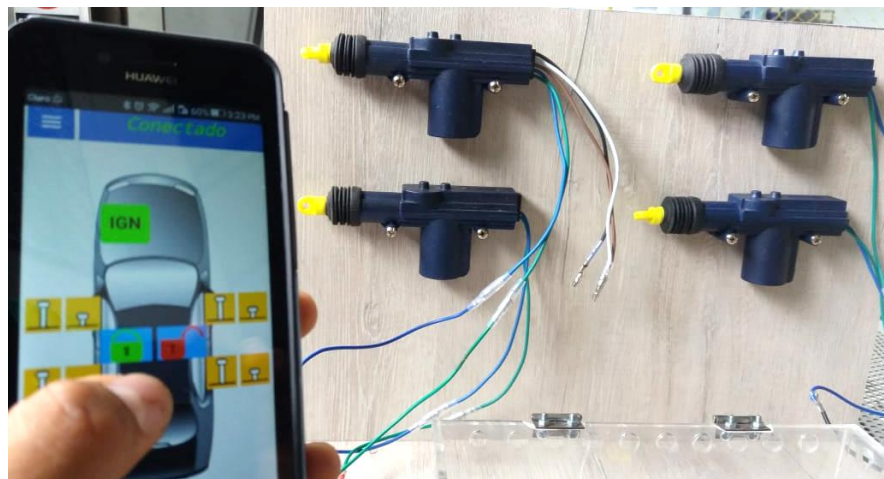


Figura 101. Control simultáneo de actuadores por medio de la aplicación

De esta manera fue comprobada la efectividad en el funcionamiento del sistema en cuanto al comando de apertura y cierre de los actuadores del bloqueo central.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Se concluyó que el sistema funciona como lo esperado y programado siempre y cuando la fuente de alimentación disponga de un nivel de voltaje superior a los 10 u 11 Voltios. De esta manera se comprobó la eficiencia en cuanto al uso práctico de este sistema en un vehículo real, ya que su diseño se enfoca también en la optimización de espacio para su fácil y cómoda instalación.
- El método de programación aplicado resultó ser indicado para el proceso de lectura y ejecución por parte del módulo Arduino, ya que se utilizó la librería software serial, estableciendo una comunicación serial entre dispositivos y modificar los pines digitales para que funcionen como puerto serial (RX y TX), donde se logró conectar el módulo Bluetooth. Esta librería permitió designar una variable cuyo valor es almacenado en "Mensaje", esta variable está encargada de grabar la orden emitida por la aplicación mediante una letra alfabética la cual es interpretada por el módulo Arduino y como consecuente este ejecuta una acción. De esta manera se comprobó la correcta interacción entre dispositivos y su rápida respuesta ante órdenes, lo que beneficia al usuario al momento de hacer uso del sistema.
- También se pudo concluir que en la mayoría de comparaciones entre las simulaciones en Proteus y la práctica los valores obtenidos son muy semejantes, a excepción de los valores en amperajes de la alimentación de la fuente debido al material empleado en la barra metálica de alimentación común y en los bornes cuyas resistencias no son consideradas en Proteus. De igual manera los valores obtenidos en la señal de salida de los módulos L298N durante la simulación fueron mayores a las obtenidas en práctica debido al suministro de energía en los dos pines de alimentación de estos módulos en Proteus, ya que así lo demanda la simulación para su funcionamiento.
- La utilización de las guías de práctica de conexión y programación que realicé son muy importantes al realizar las pruebas de funcionamiento en el sistema, ya que estas nos facilitan y garantizan la correcta conexión y programación lo cual previene posibles inconvenientes y daños provocados además de ayudarnos a ganar tiempo y no perderlo.

- El uso de programas o softwares nos ayudaron a realizar circuitos y sistemas de manera virtual que permitieron verificar su funcionamiento mediante herramientas de medición como voltímetros, amperímetros y osciloscopios, lo que en la actualidad es indispensable en el campo de la ingeniería ya que esto evita pérdida de tiempo y dinero en la realización de pruebas de funcionamiento.

4.2. RECOMENDACIONES

- Es necesario comparar las conexiones físicas del circuito de control con las de los diagramas establecidos en las guías de práctica antes de suministrar energía al sistema para evitar daños considerables en los componentes del mismo. Es muy recomendable contar con la supervisión de un docente encargado de las prácticas.
- El estado de la batería fuente para el sistema es muy importante ya que una batería descargada demandará más por parte del sistema provocando fatiga y sobreesfuerzo en los componentes, en algunas ocasiones esto provoca el reinicio automático del módulo Arduino para lo cual es necesario esperar aproximadamente cinco segundos antes de volver a realizar otra prueba. Por lo que se recomienda revisar el estado de la batería en todo vehículo para el correcto desarrollo de todo sistema eléctrico y electrónico en el mismo
- Se recomienda conservar una copia de seguridad del programa original de Arduino antes de editarlo ya que si la información es perdida la copia de seguridad servirá de apoyo. Es posible editar el programa utilizando otra metodología con nuevos comandos y variables para que este cumpla la misma función.
- Hoy en día la tecnología como la ciencia avanzan de manera apresurada, por lo que la innovación es algo muy aplicado y demandado en el campo de ingenieril. Es recomendable promover el continuo estudio de los sistemas bajo lenguaje programable y sistemas automatizados.

BIBLIOGRAFÍA

- A. G. (25 de Noviembre de 2011). *SlideShare*. Obtenido de <https://es.slideshare.net/adrianmengym/sistema-de-encendido-o-arranque>
- Amazon. (26 de Febrero de 2016). *Amazon*. Obtenido de https://images-na.ssl-images-amazon.com/images/I/51Lv3Bou%2BJL._SX425_.jpg
- Anónimo. (28 de Febrero de 2016). *eltecnicodesmt.blogspot.com*. Obtenido de https://4.bp.blogspot.com/-HqM6jejs4sA/VtNz_5DXA2I/AAAAAAAAA-Q/mFbYWjECnTw/s640/SWITCH%2BDE%2BENCENDIDO.png
- Aqeel, A. (2018). *The Engineering Projects - Tutorials & Projects for Engineers*. Obtenido de <https://www.theengineeringprojects.com/wp-content/uploads/2018/10/introduction-to-arduino-leonardo-2-2-768x886.pngleonardo-2-2.png>
- Artero, O. T. (2013). *Arduino: Curso práctico de formación*. Madrid: RC Libros.
- Banzi, M., & Shiloh, M. (2014). *Introducción a Arduino*. Sebastopol: Maker Media.
- Barrera Doblado, Ó. (2012). *Sistema de seguridad y confortabilidad*. Madrid: Paraninfo.
- Bosch. (2000). *Control del motor en motores de gasolina - Sistemas de encendido*. Alemania: Robert Bosch GmbH.
- Bosch. (2000). *Sistemas neumáticos para vehículos industriales - Conceptos, sistemas y esquemas*. Alemania: Robert Bosch GmbH.
- CESVIMAP. (2007). *Sistemas de seguridad y confortabilidad*. Constitución y Leyes (Colex) S.A.
- Cinjordiz, C. (2018). *Infootec*. Obtenido de <https://www.infootec.net/wp-content/uploads/2015/03/transistor1.jpg>
- Cinjordiz, C. (2018). *Infootec*. Obtenido de <https://www.infootec.net/wp-content/uploads/2016/08/rele-300x269.jpg>
- Cinjordiz, C. (2018). *Infootec*. Obtenido de <https://www.infootec.net/wp-content/uploads/2016/08/partes-de-un-rele-300x263.jpg>
- Consejo de Educación Superior. (2013). *Reglamento de Régimen Académico*. Quito: Gaceta Oficial del Consejo de Educación Superior.

- Daniel. (2014). *Aficionados a la mecánica*. Obtenido de http://www.aficionadosalamecanica.net/imagescurelec/esquema_elet_dis.jpg
- Delgado Crespo, M. (2016). *manueldelgadocrespo.blogspot.com*. Obtenido de <http://manueldelgadocrespo.blogspot.com/p/biblioteca.html>
- Europa Press. (2 de 11 de 2017). *Europa Press*. Obtenido de <https://www.europapress.es/portaltic/sector/noticia-bluetooth-50-principales-ventajas-nueva-tecnologia-inalambrica-smartphones-insignia-20171101112943.html>
- Fernández Ferrer, J. (1979). *Introducción al estudio de los circuitos, la electrónica y el análisis de señales*. Barcelona: Reverte.
- Frank. (3 de Mayo de 2017). *Ingeniería Mecafenix*. Obtenido de <https://i2.wp.com/www.ingmecafenix.com/wp-content/uploads/2017/05/MvedY.gif?resize=560%2C414>
- González López, M. Á., Mas Fito, J. J., & Vidal Pastor, F. J. (2011). *Sistemas antirrobo y de confort (Sistemas de seguridad y confortabilidad)*. Editex.
- Keo. (2018). *PIXERS*. Obtenido de [https://img.pixers.pics/pho_wat\(s3:700/FO/43/63/56/74/700_FO43635674_30a1af2f7975f2b9e0d1e1fb490a42fe.jpg,578,700,cms:2018/10/5bd1b6b8d04b8_220x50-watermark.png,over,358,650,jpg\)/wall-murals-abstract-technology-blue-background.jpg.jpg](https://img.pixers.pics/pho_wat(s3:700/FO/43/63/56/74/700_FO43635674_30a1af2f7975f2b9e0d1e1fb490a42fe.jpg,578,700,cms:2018/10/5bd1b6b8d04b8_220x50-watermark.png,over,358,650,jpg)/wall-murals-abstract-technology-blue-background.jpg.jpg)
- Kernighan, B., & Ritchie, D. (1991). *El Lenguaje de Programación C*. Ciudad de Mexico: Pearson Educación.
- Llamas, L. (23 de Junio de 2016). *Luis Llamas*. Obtenido de <https://www.luisllamas.es/wp-content/uploads/2016/06/arduino-transistor-bjt-esquema.png>
- Microsoft. (2010). *msn.com*. Obtenido de [https://c-6rtwjumjzx7877x24nrl-x78-rx78s-htrx2efpfrfnejix2esjy.g01.msn.com/g00/3_c-6bbb.rx78s.htr_/c-6RTWJUMJZX77x24myyux78x3ax2fx2fnrl-x78-rx78s-htr.fpfrfneji.sjyx2fyjsfsyx2ffruux2fjsynydrix2fFFjpThh.nrlx3fbx3d809x26mx3d781x26rx3d2x26n65h.rfwp3dnrflj_\\$\\$](https://c-6rtwjumjzx7877x24nrl-x78-rx78s-htrx2efpfrfnejix2esjy.g01.msn.com/g00/3_c-6bbb.rx78s.htr_/c-6RTWJUMJZX77x24myyux78x3ax2fx2fnrl-x78-rx78s-htr.fpfrfneji.sjyx2fyjsfsyx2ffruux2fjsynydrix2fFFjpThh.nrlx3fbx3d809x26mx3d781x26rx3d2x26n65h.rfwp3dnrflj_$$)
- Naylamp Mechatronics SAC. (2016). *Naylamp Mechatronics*. Obtenido de http://www.naylampmechatronics.com/img/cms/Blog/Tutorial%20de%20Uso%20del%20M%C3%B3dulo%20L298N/Tutorial%20modulo%20L298N_0.jpg

Oñós Prados, E., Ramírez Vázquez, J., Ruiz Vasallo, F., & Pecanins Vallés, F. (1996). *Maniobra, mando y control eléctricos*. Barcelona: CEAC.

Pardiñas, J. (2012). *Estudio de los sistemas de encendido (Sistemas auxiliares del motor)*. Madrid: Editex.

Raya, A. (28 de Enero de 2015). *Omicrono*. Obtenido de <https://omicrono.elespanol.com/wp-content/uploads/2015/01/bluetooth.jpg>

Reyes Cortés, F., & Cid Monjaraz, J. (2015). *Arduino: aplicaciones en robótica, mecatrónica e ingenierías*. Marcombo S.A.

Saenz Flores, M. (2018). *Curso Básico de Arduino*. Misael Saenz Flores.

ANEXOS

**ANEXO 1. GUÍA DE PRÁCTICA PARA LA CONEXIÓN
DEL SISTEMA DE ENCENDIDO Y BLOQUEO
CENTRAL VÍA BLUETOOTH AL BANCO DE PRUEBAS**

ASIGNATURA: ELECTRÓNICA AUTOMOTRIZ Nivel: 5

TEMA: CONEXIÓN Y PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH Práctica Nº: 1

1. INTRODUCCIÓN:

Durante la ejecución de esta práctica los estudiantes realizarán las conexiones de un sistema de encendido y bloqueo central comandado por una aplicación de celular vía Bluetooth a un banco de pruebas con motor CRDI tipo DIS, una vez realizadas respectivas las conexiones los estudiante realizan las respectivas pruebas de medición y funcionamiento.

2. OBJETIVO GENERAL:

- Controlar el encendido y bloqueo central de un vehículo con el celular mediante simulación en un banco de pruebas

2.1. OBJETIVOS ESPECÍFICOS	Nivel*
Realizar el conexionado del sistema al banco de pruebas	M
Establecer comunicación entre sistema y aplicación	M
Realizar mediciones de V, A y ondas de señal en el sistemas	M

*Alto(A), Medio (M), Bajo (B)

3. DEFINICIONES:

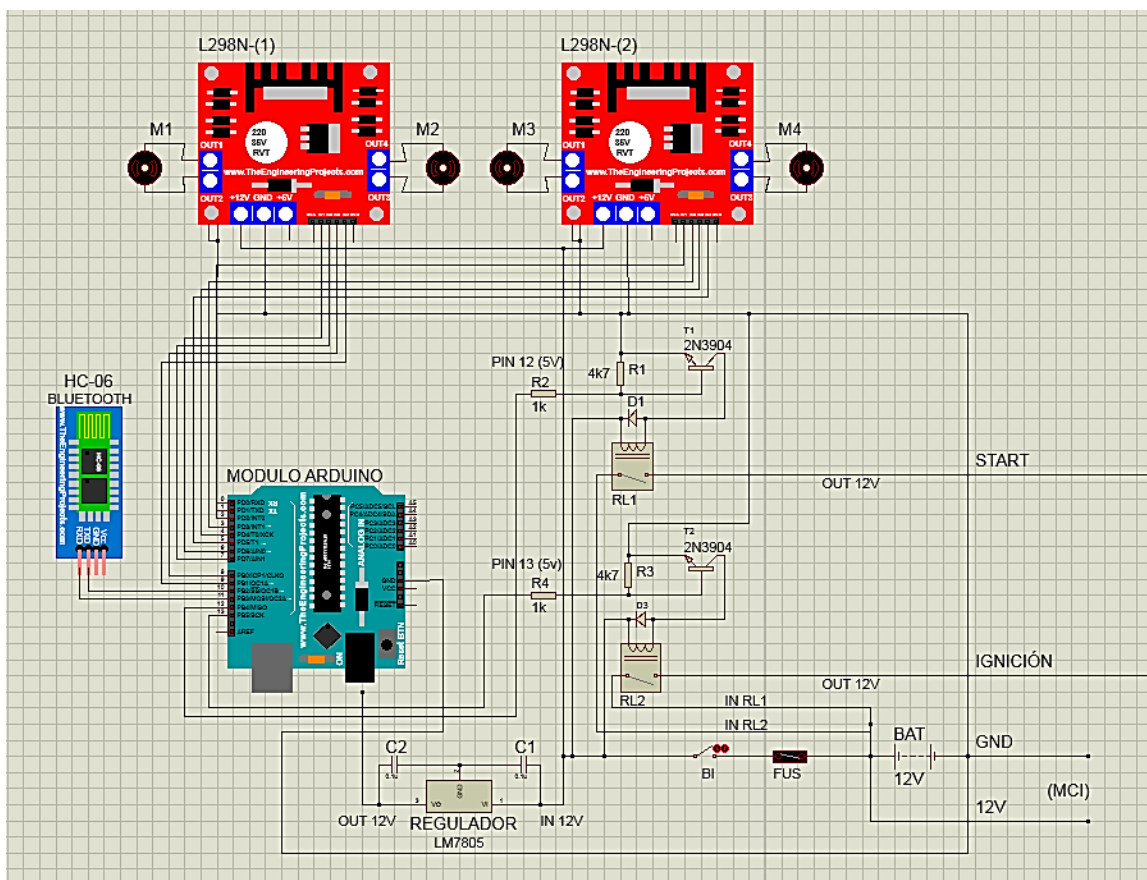


Imagen 1. Esquema del circuito de control

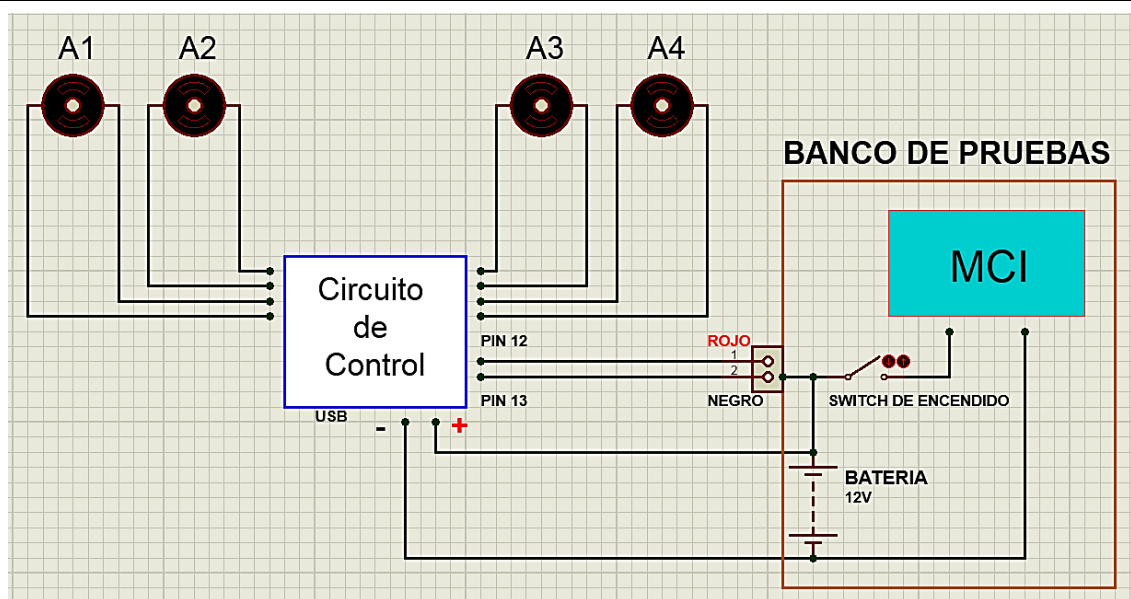


Imagen 2. Esquema de conexión del sistema

4. BASE CONCEPTUAL:

Sistema de encendido tipo DIS

Este tipo de sistema no depende del distribuidor como en la mayoría de los casos para lo cual se envía la chispa a dos de los cilindros, si no que basa su funcionamiento en la recepción de datos y condiciones enviadas por la computadora mediante sensores (posición, temperatura, revoluciones) para saber en qué momento lanzar la chispa. Este sistema suprime ciertos elementos mecánicos los cuales sufrirían daños y averías.

En el sistema de encendido tipo DIS se gana tiempo potencia y fiabilidad al momento de generar la chispa lo cual previene inconvenientes a altas revoluciones, debido al acercamiento de las bobinas a las bujías suprimiendo en algunos casos los cables de alta tensión.

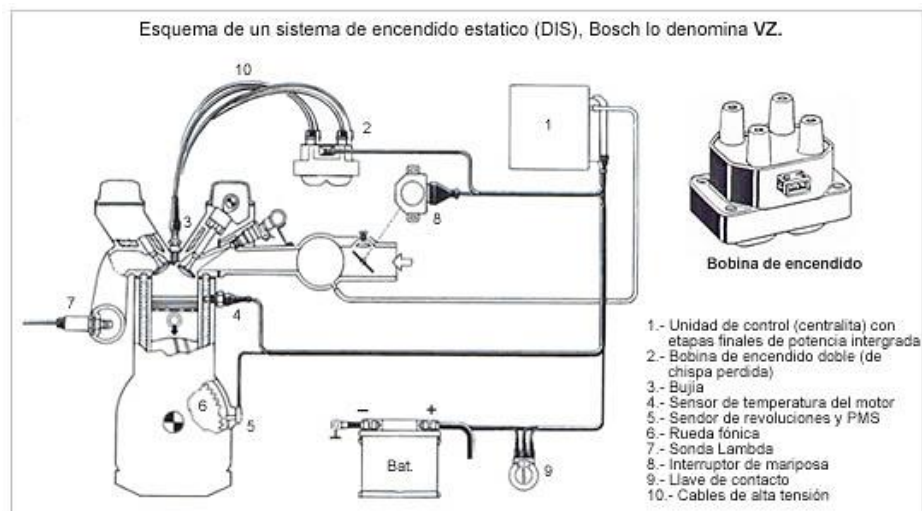


Imagen 3. Diagrama sistema de encendido DIS

Bloqueo central o cierre centralizado

En la actualidad la gran mayoría de los vehículos han adoptado la evolución del bloqueo central, cuyo principio de funcionamiento se enfoca en la apertura y el cierre de las puertas de un vehículo simultáneamente, proporcionando comodidad al evitar la apertura y cierre de todas las puertas de manera individual. Este sistema además se enfoca en brindar seguridad, evitando que alguna puerta quede abierta por error o descuido. Generalmente no es necesario el contacto entre las llaves y el bombín para que se lleve a cabo esta operación, se lo hace a distancia.

Cuando el sistema actúa, un sensor se encarga de enviar una señal hacia la centralita la cual la distribuye a cada uno de los actuadores de las cerraduras para así bloquearlas o desbloquearlas a la vez.

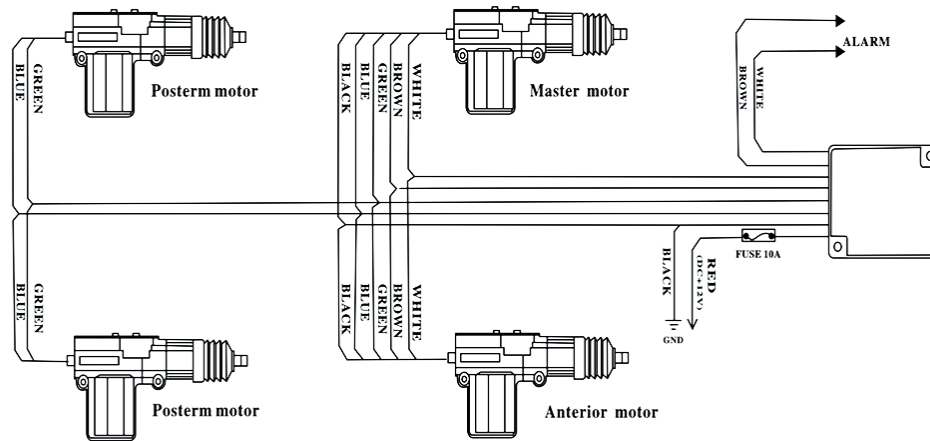


Imagen 4. Diagrama del bloqueo central

Arduino

En breve resumen, *Arduino* es una plataforma electrónica útil para la programación a manera de arquitectura abierta la cual cuenta con una extensa gama de aplicaciones en ingeniería y ciencias exactas. Este sistema se lo ha identificado con el símbolo matemático de infinito, con los signos más (+) y menos (-) en su interior.

En la siguiente figura se muestra un esquema del módulo Arduino Leonardo donde se indican los pines digitales y analógicos con los que cuenta este modelo, además se muestra el tipo de microcontrolador (Atmega32U4) con el que cuenta para el procesamiento de información, y los puertos por donde se pueden conectar a una fuente de alimentación. Al igual que la mayoría de tarjetas Arduino esta cuenta con un botón de "reset" o reinicio, el cual permite reiniciar a la tarjeta en caso de necesitar interrumpir el programa que se esté ejecutando.

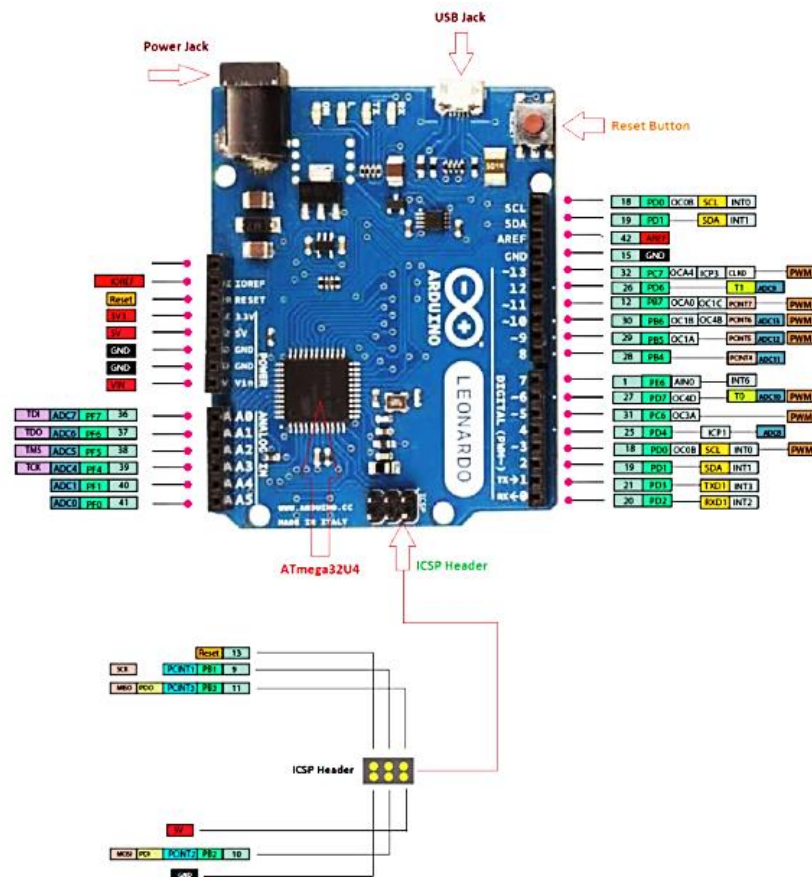


Imagen 5. Esquema Arduino Leonardo

Módulo Bluetooth HC-06

Bluetooth es una red inalámbrica de tipo WPAN (Wireless personal área network), ósea de área personal, la cual permite la comunicación de datos y de voz entre dispositivos de manera más fácil utilizando un enlace de radiofrecuencia en la banda de 2.4 GHz, dicha red tiene su especificación industrial BLUETOOTH. También se puede entender como un protocolo que de forma inalámbrica transmite datos especialmente diseñado para dispositivos de consumo bajo los cuales requieren de un corto alcance de emisión a un bajo costo.

El módulo Bluetooth modelo HC-06 actúa nada más que en modo esclavo, es decir, que sólo recibe órdenes, lo que los diferencia de los HC-05, que pueden ser programados para actuar en modo esclavo o maestro. Físicamente se diferencian por el número de pines. En el HC-06 tiene un conector de 4 pines mientras que el HC-05 trae uno de 6 pines. EL HC-06 tiene 4 pines:

- Vcc, Voltaje positivo de alimentación, aquí hay tener cuidado porque hay módulos que solo soportan voltajes de 3.3V, pero en su mayoría ya vienen acondicionados para que trabajen en el rango de 3.3V a 6V pero es bueno revisar los dato técnicos de nuestro modulo antes de hacer las conexiones.
- GND, Voltaje negativo de alimentación, se tienen que conectar al GND del Arduino o al GND de la placa que se esté usando.

- TX, Pin de Transmisión de datos, por este pin el HC-06 transmite los datos que le llegan desde la PC o Móvil mediante bluetooth, este pin debe ir conectado al pin RX del Arduino.

- RX, pin de Recepción, a través de este pin el HC-06 recibirá los datos del Arduino los cuales se transmitirán por Bluetooth, este pin va conectado al Pin TX del Arduino.

En la siguiente figura podemos observar un esquema de conexión de este dispositivo con una tarjeta Arduino.

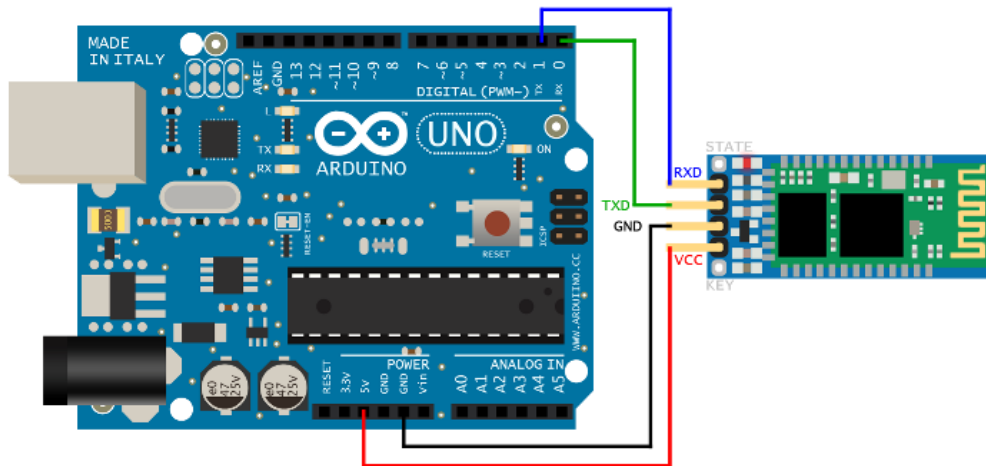


Imagen 6. Esquema de conexión Bluetooth HC-06

Módulo L298N

Este tipo de módulos utiliza el principio de funcionamiento del famoso “Puente H” para el control de motores eléctricos o motores paso a paso (sentido de giro y velocidad).

Los componentes que conforman la estructura básica de estos módulos son:

- Driver L298N
- Diodos de protección
- Regulador de voltaje (78M05)

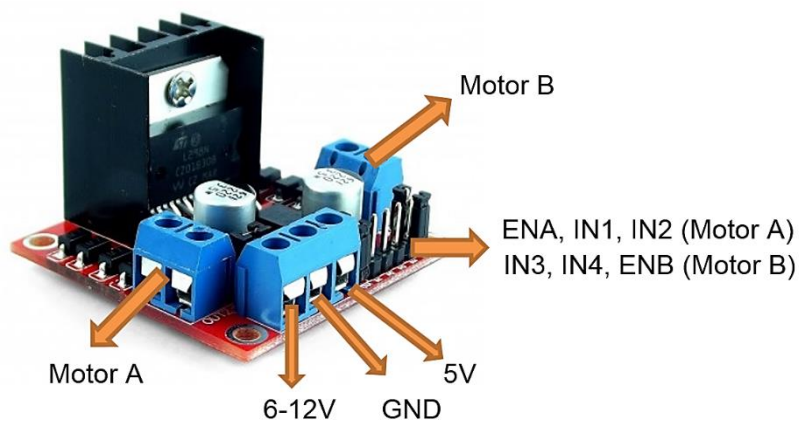


Imagen 6. Módulo L298N

Relés

Se puede denominar o llamar relé a todo dispositivo que basa su funcionamiento en el cambio de características de funcionamiento de otros elementos ya sean eléctricos o mecánicos dentro del mismo circuito o en otros circuitos eléctricos distintos.

Vemos que el relé de la figura de abajo tiene 2 contactos, uno abierto (NC) y otro cerrado (NO) (pueden tener más). Cuando metemos corriente por la bobina, esta crea un campo magnético creando un electroimán que atrae los contactos haciéndolos cambiar de posición, el que estaba abierto se cierra y el que estaba normalmente cerrado se abre. El contacto que se mueve es el C y es el que hace que cambien de posición los otros dos.

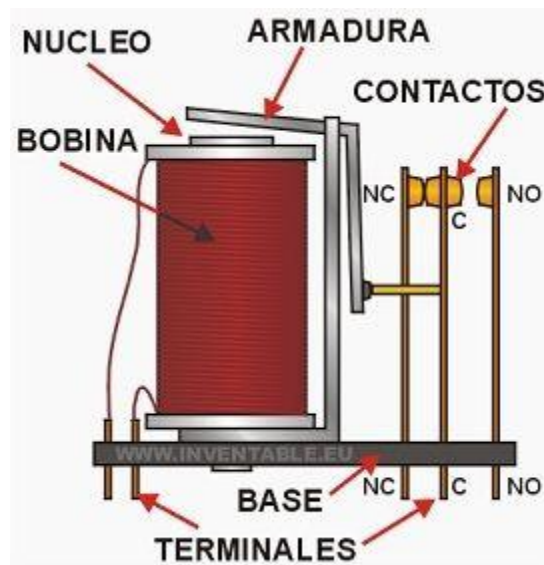


Imagen 7. Representación gráfica de un relé

Como ves habrá un circuito que activa la bobina, llamado de control, y otro que será el circuito que activa los elementos de salida a través de los contactos, llamado circuito secundario o de fuerza.

Los relés Pueden tener 1, 2, 3 o más contactos de salida, y estos puede ser normalmente abiertos o normalmente cerrados (estado normal = estado sin corriente).

5. ACTIVIDADES PREVIAS:

El estudiante deberá explorar el tema en conceptos como:

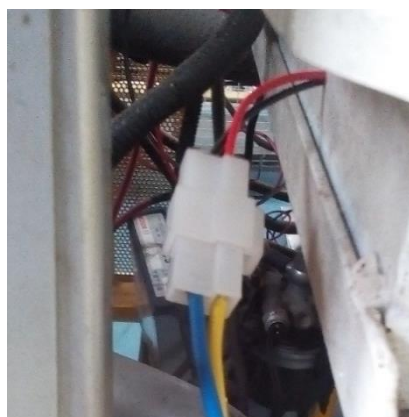
- Sistema de encendido
- Sistema de bloqueo central
- Módulos Arduino
- Módulo HC-06 (Bluetooth)
- Módulo L298N (Control de motores DC)
- Relés de mando

6. MÉTODO/PROCEDIMIENTO

- A. CONEXIÓN A LA FUENTE DE ALIMENTACIÓN** Utilizando el diagrama de conexión del sistema como guía, conectamos la batería de 12 V al banco de pruebas con los bornes positivo y negativo de este, seguido utilizamos los cables flexible rojo y negro para conectarlos junto a los bornes mencionados, el otro extremo de estos cables se los conecta a los bornes del circuito de control respetando la polaridad mediante los colores.

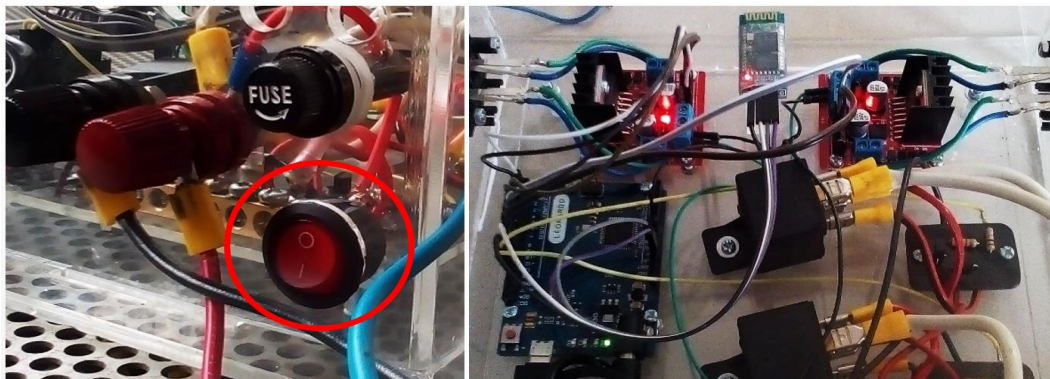


- B. CONEXIÓN AL BANCO DE PRUEBAS** De acuerdo al diagrama de conexión debemos comunicar las señales emitidas por los relés del circuito de control al switch de encendido del banco de pruebas, para lo cual utilizamos dos cables flexibles cuyos extremos están conectados a un soquet. De igual manera se respetan los colores para el conectado, el cable con marca de color rojo se conecta al borne rojo o borne con rodela (pin 12), y el cable con marca de color negro se conecta al borne negro (pin 13). Una vez realizadas las conexiones al circuito de control se conecta el soquet al que se encuentra en el lateral izquierdo del panel de conexiones del banco de pruebas.

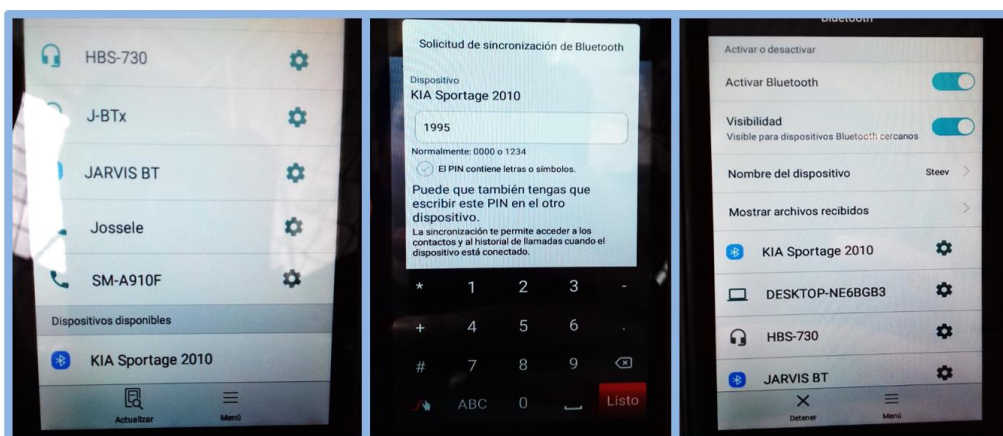


C. ENCENDIDO DEL SISTEMA Y ENLACE CON LA APLICACIÓN DE CELULAR

Se debe revisar que las conexiones estén correctamente realizadas para proceder a encender el sistema, para lo cual es necesario solamente presionar el botón interruptor del circuito de control permitiendo que la alimentación de 12 V encienda todos los dispositivos internos, los cuales disponen de leds que indican su encendido.

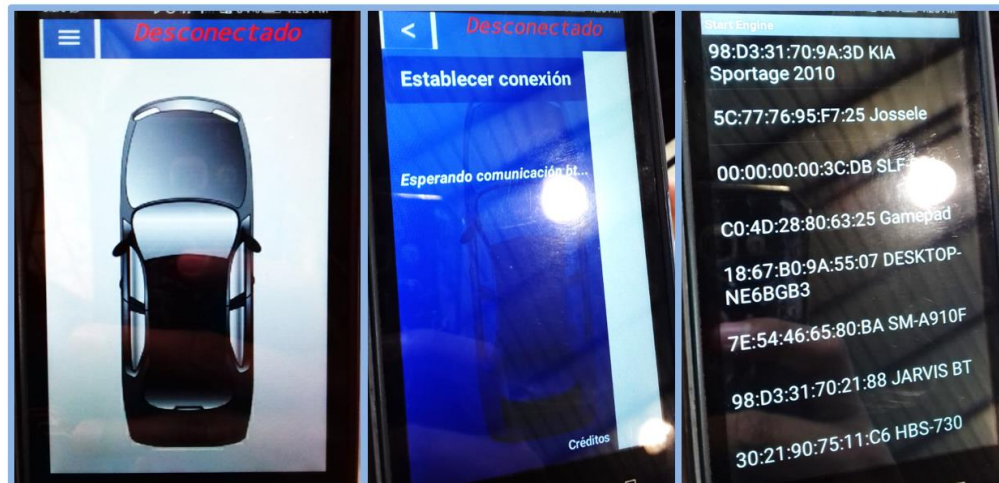


Para establecer conexión entre aplicación y sistema debemos antes enlazar el dispositivo Bluetooth con el móvil desde su configuración. Donde se puede encontrar al módulo HC-06, el cual ha sido modificado su nombre a “KIA Sportage 2010” mediante un programa cargado al módulo Arduino. Cuando se establece por primera vez la conexión Bluetooth se debe ingresar una contraseña de cuatro dígitos la cual puede ser modificada mediante un programa en Arduino. En este caso la contraseña ha sido modificada a “1995”. Una vez aceptada la contraseña el dispositivo Bluetooth éste se agrega a la lista de dispositivos conocidos.



Seguidamente debemos ingresar a la aplicación la cuál una vez instalada como cualquier otra se identifica con el logo característico y el nombre de “Start Engine”. Se recomienda utilizar móviles con versiones superiores a la v5 de Android para el correcto funcionamiento de la aplicación. Como ya se mencionó, antes de ingresar a la aplicación de celular para intentar establecer conexión desde allí, se debe encender la comunicación Bluetooth en el celular o móvil, caso contrario se muestra un mensaje en la aplicación indicando que el operario debe hacerlo.

Para establecer conexión entre aplicación y módulo HC-06 debemos ingresar al menú plegable mediante el ícono de menú ubicado en la parte superior izquierda, el menú se despliega desde el lado izquierdo de la pantalla hacia la derecha, allí se encuentra ubicado el botón “Establecer conexión” el cual nos permite acceder a la lista de dispositivos Bluetooth agregados con la aplicación, luego seleccionamos el de nombre “KIA Sportage 2010”, y esperamos a que este conecte con el módulo Bluetooth. En algunos casos puede tardar entre 15 a 20 segundos.



Se nos muestra un mensaje de advertencia indicando que el vehículo debe encontrarse en Neutro en caso de ser instalado este sistema en un vehículo real. De esta manera se comprueba que la comunicación Bluetooth entre dispositivos se logra con éxito.

D. CONTROL DE ENCENDIDO DEL BANCO DE PRUEBAS, ARRANQUE Y APAGADO DEL MCI

Para comenzar con la prueba de funcionamiento se presiona el botón de IGNICIÓN en la aplicación, lo que ordena al sistema dirigir la señal de alimentación de 12 V al cable de ignición del switch de encendido, permitiendo encender el banco de pruebas como si se lo hiciera con la llave.



Luego se procede a encender el motor por medio del botón START simulando el arranque que se da con la llave para el encendido del motor, lo cual debe realizarse en un lapso de tiempo de 1-2 segundos.

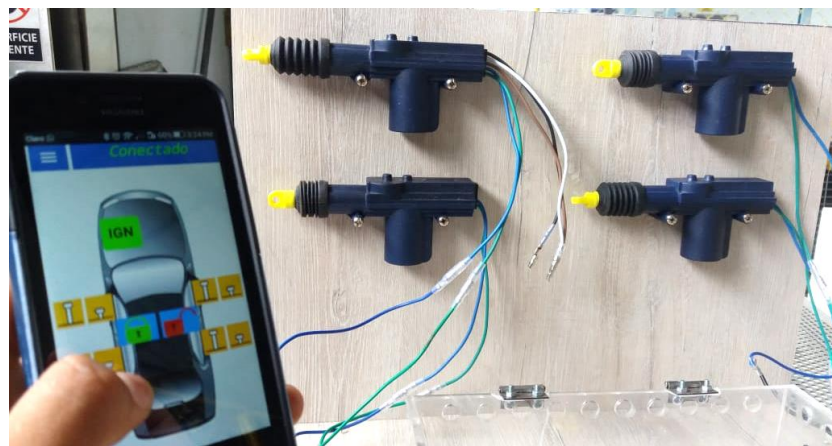


Para apagar el motor se debe presionar el botón de OFF cortando así el suministro de energía de 12 V hacia el cable de arranque del switch de encendido.

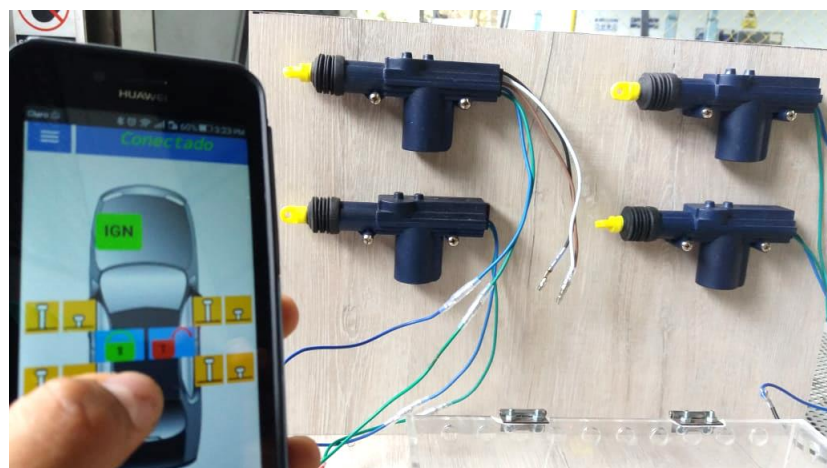


E. CONTROL DEL ACCIONAMIENTO DE LOS ACTUADORES

Para el control de la activación de los motores del bloqueo central se dispone de botones en el panel de control los cuales indican que puerta o, en este caso, que motor controlan y que acción realizarán, apertura o cierre, dependiendo de la polaridad empleada en la conexión de los actuadores al sistema.

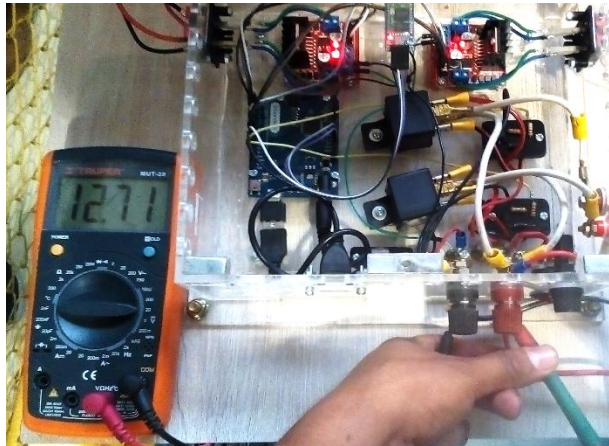


También se dispone de dos botones con símbolos de apertura y cierre por medio de un candado abierto en uno y un candado cerrado en otro, los cuales permiten el comando simultáneo de los cuatro actuadores.



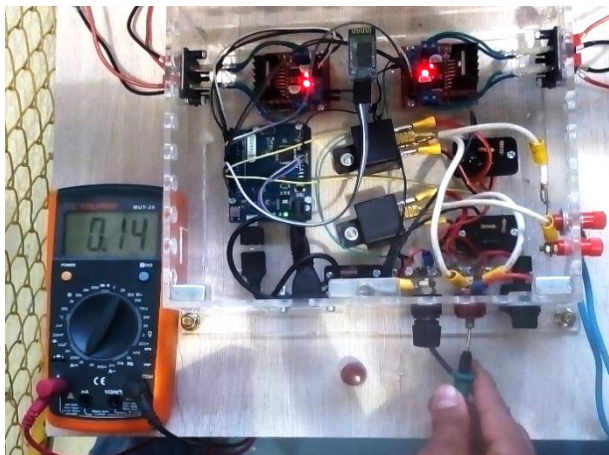
F. MEDICIÓN DE VOLTAJES, AMPERAJES Y ONDAS DE SEÑAL DIGITAL

Para las pruebas de voltaje se debe hacer uso del multímetro automotriz. Se comienza a medir la fuente de alimentación que suministra al sistema de voltaje necesario para el funcionamiento correcto del mismo. Con la ayuda de un multímetro de CC se mide el voltaje emitido por la batería a manera de comprobación, la cual nos debe dar un valor aproximado a 12 V. Para esto se conecta el polo positivo del multímetro al borne positivo del circuito de control y el borne negativo del multímetro al otro borne, es decir al negativo.

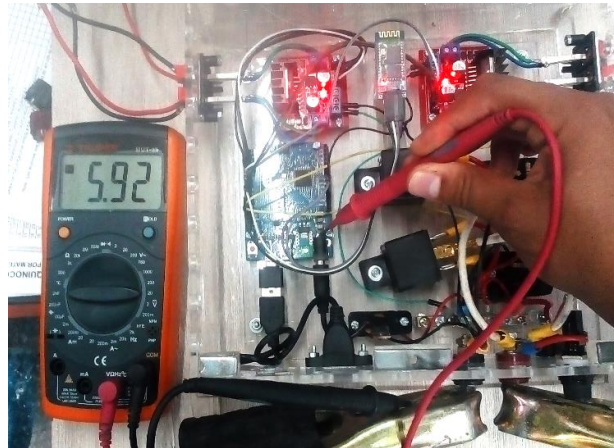


Es necesario que el sistema cuente con este nivel de voltaje ya que está diseñado para ser aplicado en un vehículo real, caso contrario el arranque del motor demandará sobreesfuerzo al sistema provocando que el módulo Arduino se reinicie.

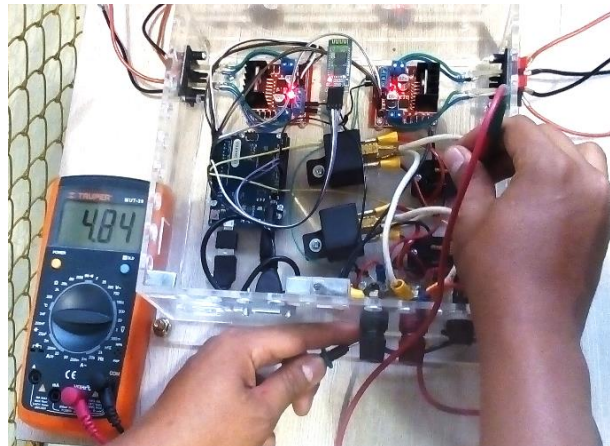
Con la ayuda del multímetro en función de amperímetro se puede medir la corriente que circula en el circuito, en este caso, el amperaje que emite la fuente de 12 V, la cual resulta ser de aproximadamente 0.14 A. Para esto se debe conectar el amperímetro en serie, entre el polo positivo de la batería y el borne rojo de alimentación.



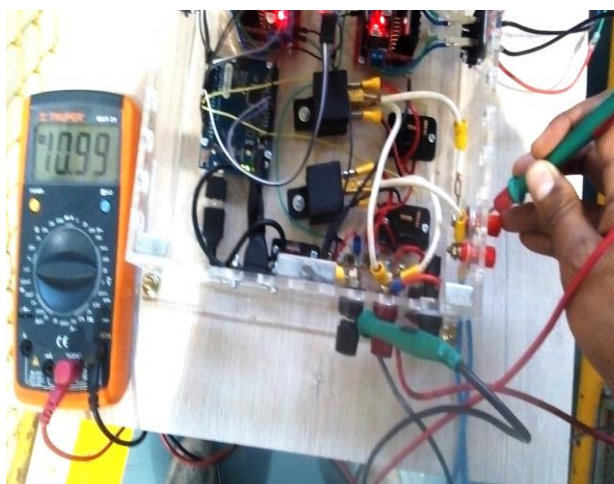
Seguidamente se comprueba que el módulo Arduino reciba el voltaje necesario para su funcionamiento. Se debe conectar el polo positivo del multímetro entre la salida de señal emitida por el circuito regulador y GND o tierra. De esta manera se verifica que el circuito regulador brinda el nivel de voltaje necesario para el óptimo funcionamiento de la tarjeta Arduino (6 V) cumpliendo con su propósito en el sistema.



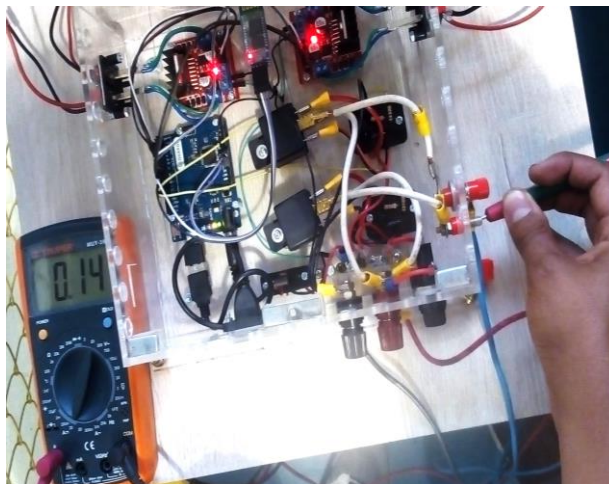
Debemos comprobar el correcto funcionamiento del módulo Arduino, para ello se conecta el polo positivo del multímetro en los pines 12 y 13, los cuales emiten una señal digital para el accionamiento de los relés y por consiguiente el encendido y apagado del MCI, y se conecta GND a tierra del sistema. Se puede verificar de esta manera que el módulo Arduino emite señales digitales de hasta 5 V.



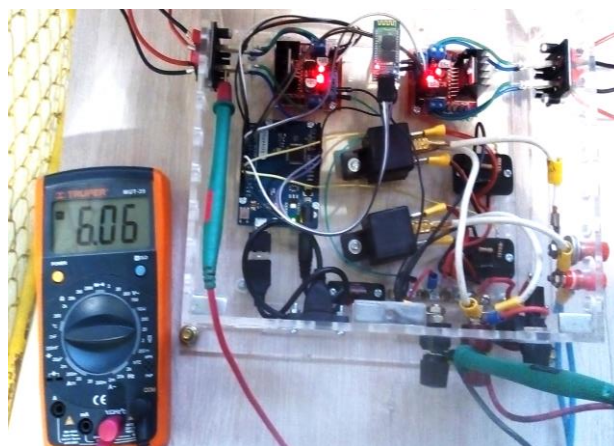
También se debe diagnosticar el funcionamiento de los circuitos optoacopladores midiendo si existe paso de voltaje desde la fuente de 12 V hasta el switch de encendido cuando se comanda el encendido del banco de pruebas y el arranque del motor, para ello se conecta el polo positivo del multímetro en la salida del relé hacia el switch y el polo negativo a tierra. De esta manera se puede comprobar que existe un voltaje aproximado a los 11 V debido a la caída de tensión.



De igual debemos medir el valor en amperios obtenido en la salida de los circuitos optoacopladores (relés) con el fin de compararlo con el valor de corriente a la entrada del circuito de control, para ello se conecta el amperímetro en serie, entre la salida de relés y el soquet del banco de pruebas. Se debe obtener un valor muy parecido al amperaje de la fuente, es decir, 0.14 A.

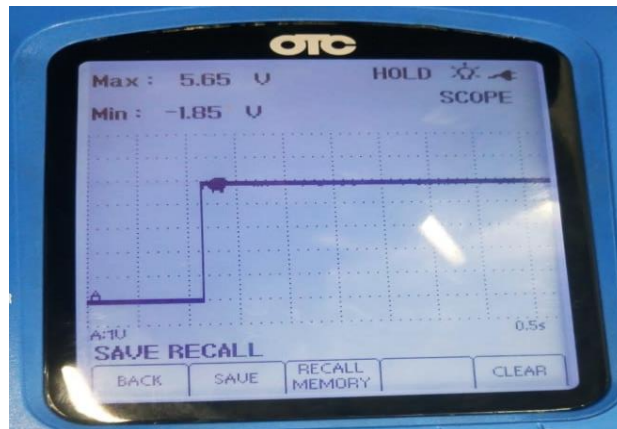


Para obtener los valores de señal emitidas por los módulos L298N en sus pines de salida (OUT 1, OUT 2, OUT 3 y OUT 4) se conecta el polo positivo del multímetro entre el pin de salida del L298N y el actuador, mientras que el polo negativo va a tierra. Se obtiene un voltaje aproximado de 6 V, valor suficiente para accionar los actuadores del bloqueo central.

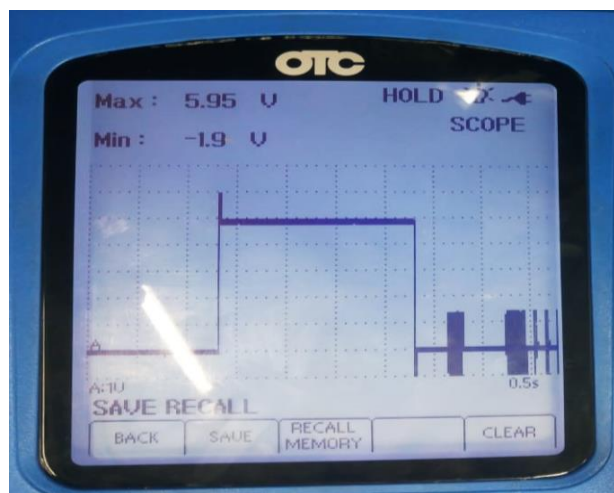


Finalmente debemos obtener las ondas de señal digitales emitidas por los pines 12 y 13 con la ayuda del osciloscopio. En el pin 13 se debe obtener una señal estable ya que ésta acciona al relé que comunica a la fuente de 12 V con la señal de ignición del switch de encendido del banco de pruebas, y en el pin 12 un pulso de señal digital que acciona el relé que simula el pulso de arranque que realiza la llave en la posición de “start”.

Para ello se conecta el osciloscopio con el conector del canal “A” en la salida del pin 13 y negativo a tierra, prueba de medición que se debe realizar mínimo durante 5 segundos.



De igual manera para obtener la onda de señal del pin 12 se hace uso del conector del mismo canal en la salida de este, dicha prueba se debe realizar durante un lapso mínimo de 5 segundos dando un arranque al motor hasta que este encienda, lo cual nos permite obtener una onda de señal digital de 1 a 2 segundos de duración.



Pueden obtenerse las dos ondas de señal utilizando los canales A y B del osciloscopio para realizar una gráfica comparativa en el mismo lapso de tiempo.

VALORES OBTENIDOS

Razón de la prueba	Valor aproximado ($\pm 10\%$)
Voltaje de alimentación del sistema	12 V
Voltaje de alimentación del Arduino	6 V
Señal emitida por el Arduino	5 V
Voltaje emitido por relés	11 V
Voltaje emitido por módulos L298N	6 V
Corriente emitida por relés	0.14 A
Corriente de la alimentación del sistema	0.14 A
Onda de señal digital pin 12	Señal digital o cuadrada durante 10 seg.
Onda señal digital pin 13	Señal digital o cuadrada durante 1.5 seg.

7. EQUIPOS Y MATERIALES:

EQUIPO DE LABORATORIO	MATERIAL Y EQUIPO DE APOYO
MULTÍMETRO AUTOMOTRIZ	SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH
OSCILOSCOPIO AUTOMOTRIZ	BANCO DE PRUEBAS CRDI
	TELÉFONO CELULAR
	MESA DE TRABAJO
	INFORME DE PRACTICA

8. CONDICIONES DE SEGURIDAD:

- Para el ingreso al laboratorio, el estudiante debe portar mandil azul y su respectiva identificación.
- No ingresar alimentos, bebidas, bebidas alcohólicas, ni fumar dentro o alrededor del laboratorio.
- Se prohíbe estrictamente el ingreso a prácticas habiendo ingerido antes bebidas alcohólicas.
- Utilizar equipos con responsabilidad y bajo la supervisión del Docente o Responsable del laboratorio.
- Bajo ninguna circunstancia se deberá dejar basura en la mesa, piso o cajones de las mesas de trabajo, para esto se cuenta con depósitos apropiados.
- Si el usuario detecta cualquier anomalía en los quipos, tiene que informar inmediatamente al responsable del laboratorio para que se pueda solucionar de manera oportuna.
- Cumplir con las señales, avisos de seguridad y manejo de equipos.
- En el caso de sufrir algún accidente comunicar inmediatamente al Docente o al responsable del laboratorio.

9. EVALUACIÓN DEL APRENDIZAJE:

Resultados de aprendizaje de la carrera	Resultados de aprendizaje de la asignatura	Nivel del aprendizaje de la práctica
<i>Integra las tecnologías de área eléctrica, mecánica, electrónica, control y automatización en el manejo y control de procesos de producción.</i>	<i>1.1 Aplica los principios algebraicos que rigen el funcionamiento de los circuitos digitales combinacionales y secuenciales.</i>	/A

*Alto(A), Medio (M), Bajo (B)

10. EVALUACIÓN DE RESULTADOS OBTENIDOS:

Resultados obtenidos	SI	NO	Observaciones
<i>El estudiante ha comprobado mediante el diseño, simulación y armado de los CLC, la aplicación de los aspectos teóricos con la practica la aplicación de los conocimientos parte profesional.</i>	X		

11. BIBLIOGRAFÍA:

	Tipo (básica o complementaria)
	Bibliografía Básica
	Bibliografía Básica
	Bibliografía complementaria
	Bibliografía complementaria

12. RÚBRICA DE EVALUACIÓN DE PRÁCTICA

ASPECTOS A EVALUAR	PUNTUACIÓN
PRESENTACIÓN DEL CONEXIONADO DEL SISTEMA (funcionando correctamente)	2 puntos
Presentación del informe	2 puntos
Preguntas realizadas por el profesor	2 puntos
Pruebas y mediciones	2 puntos
Exposición del trabajo realizado	2 puntos
Total	10 Puntos

<p>Elaborado por:</p>  <p>_____</p> <p>Ing. Elvis Patricio Gualotuña Quishpe. DOCENTE</p> <p>Fecha: 16/10/2018</p>	<p>Revisado por:</p>  <p>_____</p> <p>Ing. Elvis Patricio Gualotuña RESPONSABLE DE ÁREA</p> <p>Fecha: 15/10/2018</p>	<p>Aprobado por:</p> <p>_____</p> <p>Ing. Xabier Orbea COORDINADOR DE CARRERA (E)</p> <p>Fecha: 15/10/2018</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------

**ANEXO 2. GUÍA DE PRÁCTICA PARA LA
PROGRAMACIÓN DEL SISTEMA DE ENCENDIDO Y
BLOQUEO CENTRAL VÍA BLUETOOTH**

ASIGNATURA: ELECTRÓNICA AUTOMOTRIZ	Nivel: 5
------------------------------------	----------

TEMA: PROGRAMACIÓN DEL SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH	Práctica Nº: 1
-----------------------------------------------------------------------------	----------------

1. INTRODUCCIÓN:

Durante la ejecución de esta práctica los estudiantes realizarán la programación en el software Arduino para su compilado y envío al módulo Arduino Leonardo del sistema de encendido y bloqueo central vía Bluetooth, una vez cargado el programa al módulo se realizarán las respectivas pruebas de funcionamiento.

2. OBJETIVO GENERAL:

- Desarrollar el programa en lenguaje C de Arduino que permita el correcto funcionamiento del sistema de encendido y bloqueo central vía Bluetooth

2.1. OBJETIVOS ESPECÍFICOS	Nivel*
Realizar el programa en el software de Arduino	M
Cargar el programa a la tarjeta Arduino	B
Realizar las respectivas pruebas de funcionamiento	M

*Alto(A), Medio (M), Bajo (B)

3. DEFINICIONES:

```
#include <SoftwareSerial.h>

SoftwareSerial Serial_2 (10, 11);
String Mensaje;

void setup() {
  Serial_2.begin(9600);
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
}

void loop() {
  while (Serial_2.available()) {
    delay(5);
    char c = Serial_2.read();
    Mensaje += c;
  }
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  digitalWrite(12,LOW);
  digitalWrite(13,LOW);
}
```




```

if (Mensaje.length()>0){
  Serial.println(Mensaje);
  if (Mensaje == "a"){
    digitalWrite(13,HIGH);
  }
  if (Mensaje == "b"){
    digitalWrite(13,LOW);
  }
  if (Mensaje == "c"){
    digitalWrite(12,HIGH);
  }
  if (Mensaje == "d"){
    digitalWrite(12,LOW);
  }
  if (Mensaje == "e"){
    digitalWrite(2,HIGH);
    digitalWrite(3,LOW);
    delay(200);
  }
  if (Mensaje == "f"){
    digitalWrite(2,LOW);
    digitalWrite(3,HIGH);
    delay(200);
  }
  if (Mensaje == "g"){
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
    delay(200);
  }
  if (Mensaje == "h"){
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
    delay(200);
  }
  if (Mensaje == "i"){
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    delay(200);
  }
  if (Mensaje == "j"){
    digitalWrite(6,LOW);
    digitalWrite(7,HIGH);
    delay(200);
  }
  if (Mensaje == "k"){
    digitalWrite(8,HIGH);
    digitalWrite(9,LOW);
    delay(200);
  }
  if (Mensaje == "l"){
    digitalWrite(8,LOW);
    digitalWrite(9,HIGH);
    delay(200);
  }
}

```

```

if (Mensaje == "m"){
    digitalWrite(2,LOW);
    digitalWrite(3,HIGH);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
    digitalWrite(6,LOW);
    digitalWrite(7,HIGH);
    digitalWrite(8,LOW);
    digitalWrite(9,HIGH);
    delay(200);
}
if (Mensaje == "n"){
    digitalWrite(2,HIGH);
    digitalWrite(3,LOW);
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(8,HIGH);
    digitalWrite(9,LOW);
    delay(200);
}
}
Mensaje="";
}

```

Código de programación Arduino del sistema de encendido y bloqueo central vía Bluetooth.

4. BASE CONCEPTUAL:

El sistema Arduino basa su principio de funcionamiento mediante la familia de microcontroladores *ATMEL* y el lenguaje de programación *C* y *C++* los cuales son herramientas que permiten el modelado, procesado y análisis de programación que representan un sistema empotrado permitiendo automatizar procesos físicos. Una de las grandes ventajas de este sistema es que su software es gratuito el cual puede obtenerse desde la página web principal de Arduino, sus tarjetas o módulos y diversos dispositivos pueden adquirirse como sistemas de desarrollo a precios muy accesibles, los cuales también pueden ensamblarse a mano. Además el sistema mantiene compatibilidad con casi todos los sistemas operativos como Windows, Linux y Macintosh OSX. La empresa bajo el nombre de Creative Commons se encarga de publicar los diagramas esquemáticos y módulos de componentes bajo su licencia, siendo realizadas versiones mejoradas, ampliadas y optimizadas de los módulos por diseñadores de sistemas electrónicos.



Imagen 1. Arduino UNO y cable estándar USB

Librerías Arduino

Dentro de las librerías estándar del lenguaje C existe un subconjunto de librerías avr-libc, las cuales contienen funciones específicas que permiten utilizar los recursos de los microcontroladores RISC ATMEL AVR, por lo que brindan la forma de operar o trabajar para realizar un amplio espectro en cuanto a aplicaciones dentro del ámbito de las ciencias exactas e ingeniería.

A continuación se detallan algunas de las librerías del sistema Arduino y del lenguaje C.

Tabla 1. Librerías del lenguaje C y del sistema Arduino

Librería	Descripción
#include <allocat.h>	Asignación temporal de memoria stack.
#include <math.h>	Funciones matemáticas.
#include <stdlib.h>	Librerías estándar.
#include <avr/cpufunc.h>	Funciones especiales del microcontrolador AVR.
#include <avr/eeprom.h>	Manejo e memoria EEPROM.
#include <avr/interrupt.h>	Manejo de interrupciones.
#include <avr/io.h>	Manejo y definiciones de dispositivos entrada/salida.
#include <Ethernet.h>	Librerías para comunicación por medio de Ethernet.
#include <ArduinoRobot.h>	Librerías para el control de la tarjeta Robot.
#include <GSM.h>	Conexión a red GSM/GRPS con GSM shield.
#include <LiquidCrystal.h>	Control del exhibidor de cristal líquido.
#include <SD.h>	Lectura y escritura de tarjetas SD.
#include <Servo.h>	Funciones para manejo y control de servomotores.
#include <SPI.h>	Comunicación con dispositivos usando el bus de Interface serial periférica (SPI).
#include <Stepper.h>	Control de motores a paso.
#include <WiFi.h>	Comunicación por medio de Internet usando Arduino WiFi shield.
#include <Wire.h>	Comunicación para transmitir/recibir datos por medio del protocolo TWI/I2C.

Instalación del sistema Arduino

Para llevar a cabo todo proceso de instalación se debe emplear una metodología que permita ejecutar ciertos pasos secuenciales y así obtener un correcto funcionamiento en el ambiente de programación con sus respectivas funciones, herramientas y comandos tales como: la compilación, edición, monitoreo serial, descarga de código de máquina, librerías, etc. De igual manera esto permite que el canal de comunicación vía USB opere de manera adecuada con los diferentes modelos de módulos o tarjetas electrónicas ARDUINO.

Para la instalación del sistema ARDUINO se describe a continuación los siguientes pasos:

- Adquirir desde la página web de Arduino (www.arduino.cc) la última versión del software. La dirección específica para la descarga del software es <http://arduino.cc/en/Main/Software>. El paquete de cómputo automáticamente detecta en el equipo o computador el idioma predeterminado. El software cuenta también con la opción Preferencias para el idioma castellano.
- Elegir el archivo arduino-00XX-sistemaoperativo.zip a descargar según el de sistema operativo, ya sea Windows, MacOS X o Linux, donde sistema operativo representa el tipo de sistema operativo y XX indica la última versión del software. El archivo a descargar contiene un conjunto de carpetas (tools, hardware, examples, reference, lib, java, librerías, drivers) y el archivo ejecutable arduino.exe.
- Una vez finalizada la descarga, se procede a descomprimir el archivo arduino-00XX-sistemaoperativo.zip en la carpeta destino a elegir.
- Cuando la descompresión haya finalizado, se deberá encontrar el archivo ejecutable arduino.exe en la carpeta destino seleccionada. Es importante conservar las carpetas y subdirectorios en su estructura original.
- Por último, para completar la comunicación entre software y hardware se debe contar con uno de los modelos de tarjetas electrónicas ARDUINO, como por ejemplo el modelo UNO o Leonardo con su respectivo cable USB, tal como se representa en la figura.

Ambiente de programación Arduino

En Arduino, su ambiente de programación permite el desarrollo, edición de programas, compilación, depuración de errores, al igual que la descarga de los códigos de máquina con los respectivos modelos de tarjetas electrónicas con sus plataformas las cuales basan su funcionalidad en base a un tipo de microcontrolador ATMEL. Además es posible el intercambio de información de comandos o datos a través del puerto USB. La interface de desarrollo entre el usuario y el módulo Arduino es representada por el ambiente de programación lo cual permite realizar amigablemente y específicamente aplicaciones en el campo de la automatización, el control, la ingeniería robótica, la mecatrónica, la física, la matemática, sistemas e instrumentación electrónica.

Una vez hallada la ruta o ubicación del archivo ejecutable arduino.exe por motivos de comodidad se recomienda que el usuario cree un acceso directo de este archivo y lo ubique en el escritorio. En la ventana principal del ambiente de programación Arduino se puede apreciar en la parte superior izquierda el nombre inicial del archivo que genera además de la versión del ambiente, a continuación se muestra el menú principal con todas las opciones que dispone la interface (Archivo, Editar, Sketch, Herramientas, y Ayuda) Los nombres de cada uno de los programas Arduino son denominados sketch cuya extensión es .ino, y se crean dependiendo de la fecha en la que son creados o editados, por ejemplo sketch_jan22 corresponde al 22 de enero del año actual, tal como se indica en la imagen.

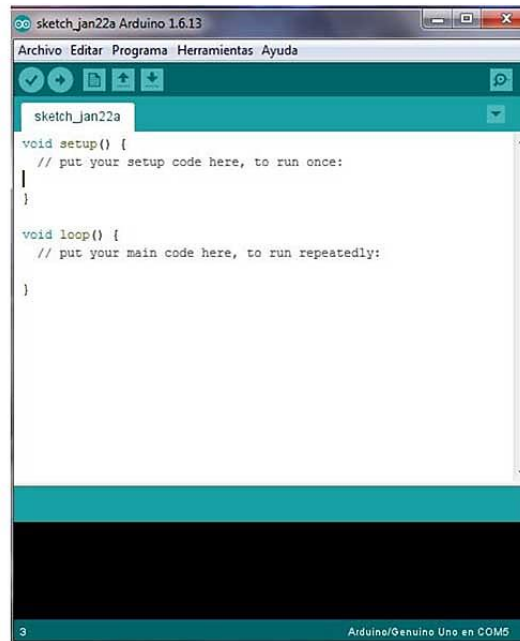








Imagen 2. Ventana principal del ambiente de programación Arduino

El archivo sketch puede ser renombrado con cualquier otro nombre por el usuario con símbolos alfanuméricos, automáticamente se añade al final del nombre del archivo la extensión **.ino**.

En la ventana principal también se pueden apreciar botones en la barra de herramientas que activan ciertos comandos de fácil acceso para el compilado y la descarga de los programas realizados en el ambiente de programación Arduino hacia las tarjetas o módulos, manteniendo de esta manera la comunicación de tipo serial entre el computador y las tarjetas. Con la ayuda del botón derecho del ratón se pueden activar dichos comandos. Se enlistan a continuación los principales comandos para el desarrollo de sketch:

- El icono  permite la verificación y la compilación de la redacción o sintaxis del programa de acuerdo con el lenguaje de programación C/C++.
- Con el icono  se inicia la descarga de códigos de máquina generados de la compilación de sketch hacia la tarjeta o módulo Arduino.
- Con el icono  se crea una nueva programación o sketch, cuyo nombre de archivo dependerá de la fecha en que se genere, y para que éste se distinga de otros archivos creados previamente en la misma fecha, se anexará automáticamente al final del nombre una letra alfabética, por ejemplo, sketch_jan22a puede distinguirse de un programa anterior sketch_jan22 en este caso por la letra agregada a.
- Con el ícono de comando  se puede abrir un sketch o programa antes creado o generado.
- El ícono  permite salvar o guardar programas sketch.

- El último ícono  de la barra de comandos activa la visualización del monitor serial el cual exhibe en una ventana la información de comandos o variables de la ejecución del programa del usuario en el módulo Arduino. Este comando viene incluido en la sección Herramientas del menú; también puede obtenerse oprimiendo de manera simultánea las teclas Ctrl+Mayúsculas+M.

Al iniciar el ambiente de programación Arduino este se encuentra de manera directa en modo de edición, pudiendo verse que en el área blanca que muestra la imagen el usuario puede escribir instrucciones de programación. Al igual que todo editor de texto este también cuenta con barras de desplazamiento horizontal y vertical. En la parte inferior de la barra de desplazamiento horizontal se hallan dos franjas cuyo propósito es mostrar o indicar los procesos de compilado y descarga de códigos hacia el módulo Arduino.

5. ACTIVIDADES PREVIAS:

El estudiante deberá explorar el tema en conceptos como:

- Introducción a Arduino
- Ambiente de programación Arduino
- Librerías Arduino
- Lenguaje de programación C

6. MÉTODO/PROCEDIMIENTO DE PROGRAMACIÓN

- A. DECLARACIÓN DE VARIABLES** En esta parte o sección del programa Arduino se declaran las variables a utilizar en el resto de la programación para que el módulo pueda reconocerlas, al declararlas se deben indicar el tipo de variable y luego un nombre para identificarlas, de manera opcional a estas se les puede asignar un valor. Esta declaración se debe realizar una sola vez en cada programa.

En la primera línea del programa hacemos uso del comando **#include**, el cual nos permite incluir librerías cuyos nuevos comandos necesitaremos para el resto del programa, en este caso incluimos la librería **<SoftwareSerial.h>** cuyos comandos nos permitirá establecer una comunicación serial con el dispositivo Bluetooth.

```
#include <SoftwareSerial.h>
```

Mediante el comando **SoftwareSerial** configuramos los pines digitales 10 y 11 para que funcionen como RX y TX con la declaración de la variable **Serial_2**, ya que será un segundo puerto serial (el primer puerto serial es el que comunica al módulo Arduino con el computador), lo cual permitirá la comunicación con el módulo Bluetooth.

```
SoftwareSerial Serial_2 (10, 11);
```

Se declara una variable con el comando **String**, el cual permite que se establezca una cadena de caracteres (en este caso letras alfabéticas) que se almacenarán en la palabra Mensaje. Estos caracteres tendrán la función de establecer órdenes emitidas por la aplicación de celular para que el módulo Arduino ejecute el envío de señales digitales por los pines a utilizar.

```
String Mensaje;
```

- B. SECCIÓN SETUP** En esta sección del programa “setearemos” las funciones que va a llevar a cabo el microcontrolador del módulo Arduino, para lo cual estableceremos criterios que se ejecutaran una sola vez.

```
void setup() { }
```

Se establece la comunicación serial entre módulo Arduino y módulo Bluetooth a 9600 baudios utilizando el comando **begin**, valor que generalmente se utiliza ya que es un nivel favorable de comunicación para programas no muy complejos.

```
Serial_2.begin(9600);
```

De igual manera iniciamos la comunicación serial entre módulo Arduino y computador, a 9600 baudios. Utilizamos nuevamente el comando **begin**.

```
Serial.begin(9600);
```

Ahora establecemos el modo de función en el que trabajarán los pines a utilizar en este sistema, los cuales deben trabajar como pines de salida de señal, para ello haremos uso del comando **pinMode**. En el primer argumento para este comando dentro de los paréntesis declaramos el número de pin, por ejemplo 13, y en el segundo argumento escribimos el modo de trabajo del pin, en este caso utilizaremos el comando **OUTPUT**. Repetimos esta línea con todos los pines a utilizar como salidas de señal (2, 3, 4, 5, 6, 7, 8, 9, 12, 13).

```
pinMode(13, OUTPUT);
pinMode(12, OUTPUT);
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
```

- C. SECCIÓN LOOP** Para esta sección debemos desarrollar el programa o condiciones que se ejecutarán un infinito número de veces utilizando la o las variables anteriormente declaradas una vez encendido el módulo Arduino.

```
void loop() { }
```


En la primera línea de esta sección estableceremos que todo el programa y condiciones se ejecuten solo cuando los caracteres de orden enviados por la aplicación de celular sean leídos e interpretados por el módulo Arduino. Para esto haremos uso del comando **while**, el cual permite establecer la condición de que otras condiciones se cumplan mientras esté en funcionamiento o ejecución un criterio, en este caso, cuando un mensaje sea enviado al módulo Bluetooth utilizando la comunicación serial de este dispositivo. Seguido en el argumento de este comando habilitamos el puerto serial de la comunicación del módulo Bluetooth a la tarjeta Arduino mediante el comando **available()**.

```
while (Serial_2.available()) {
```

La primera acción dentro de la condición **while** será que antes de que se ejecute cualquier otra acción o condición transcurran 5 ms utilizando el comando **delay** (la unidad para medir el tiempo en el argumento de este comando se da en milisegundos), esto se hace comúnmente para que existan ciertas pauta entre acciones y no se ejecuten al instante, con el fin de darle suavidad y agradable fluidez a una cadena de ejecuciones y acciones.

```
delay(5);
```

Declaramos la variable "c" utilizando el comando **char**, el cual nos permite declarar variables en donde se almacenarán datos (en este caso letras) en 1 byte para luego ser aplicado en fórmulas o cálculos, para este programa lo utilizaremos en el almacenamiento de ordenes por medio de letras para que el módulo Arduino envíe señales digitales por los pines de salida configurados. Seguido a esto indicamos que los datos guardados en "c" serán leídos de la comunicación serial del módulo Bluetooth, utilizando el comando **read()** al final de la declaración.

```
char c = Serial_2.read();
```

Establecemos una pequeña formula en la que indicamos que Mensaje sumará los datos almacenados o guardados en la variable "c", permitiendo de esta manera que se lea siempre el último mensaje de orden enviado por la aplicación de celular y por ende se ejecute o realice la acción correspondiente a esa última orden. Aquí culminan las condiciones a utilizar dentro del comando **while**.

```
Mensaje += c;
```


Configuramos el estado de los pines a utilizar cada vez que se empiece a ejecutar el programa, en este caso todos los pines iniciaran en estado pasivo, es decir, que no emitirán ninguna señal hasta que se les ordene. Para esto hacemos uso del comando **digitalWrite**, en el primer argumento indicamos el pin y en el segundo argumento su estado, **LOW**.

Ahora determinaremos las condiciones de funcionamiento durante la recepción de mensajes de orden de la aplicación para la ejecución de acciones por parte de la tarjeta Arduino, para lo cual haremos uso repetidas veces del comando condicional **if**, el cual establece que debe cumplirse una condición para realizarse una acción. Esta primera línea será la que imponga la condición principal para la ejecución de cualquier otra acción, ya que dentro del argumento del comando **if** se establece que la tarjeta Arduino actuará siempre y cuando la longitud de datos en la variable **Mensaje** sea mayor a 0, es decir, que mientras no se haya enviado ningún mensaje de orden no se llevará a cabo ninguna acción. Para medir la longitud de datos almacenados en una variable se utiliza el comando **length**.

Esta línea del programa sirve para mostrar datos en el monitor serial, en este caso las letras de los mensajes de orden enviados por la aplicación, para ello hacemos uso del comando **Serial.println ()** en cuyo argumento establecemos que se muestre lo almacenado en la variable **Mensaje**.

A partir de esta línea estableceremos las condiciones para que se ejecute una acción en cuanto un mensaje de orden sea recibido por el módulo Arduino a través del módulo Bluetooth. Por ejemplo, para que el pin 13 pase del estado **LOW** a **HIGH** y envíe una señal que permita el accionamiento del relé a ignición para encender el banco de pruebas debe recibirse un mensaje cuyo dato almacenado en la variable "c" corresponda a la letra *a* siendo que **Mensaje == "a"**. El envío de estos datos alfanuméricos es programado en la aplicación de celular. De igual manera, si se quiere que este pin deje de emitir señal y pase al estado **LOW** debe recibirse un mensaje con valor *b*. La misma condición se establece para el pin 12 pero con diferentes valores, o letras en este caso.

```
digitalWrite (2,LOW) ;
digitalWrite (3,LOW) ;
digitalWrite (4,LOW) ;
digitalWrite (5,LOW) ;
digitalWrite (6,LOW) ;
digitalWrite (7,LOW) ;
digitalWrite (8,LOW) ;
digitalWrite (9,LOW) ;
digitalWrite (12,LOW) ;
digitalWrite (13,LOW) ;
```

```
if (Mensaje.length () > 0) {
```

```
Serial.println (Mensaje) ;
```

```
if (Mensaje == "a"){
    digitalWrite (13,HIGH) ;
}
if (Mensaje == "b"){
    digitalWrite (13,LOW) ;
}
if (Mensaje == "c"){
    digitalWrite (12,HIGH) ;
}
if (Mensaje == "d"){
    digitalWrite (12,LOW) ;
}
```

Para la activación individual de los actuadores del bloqueo central, incluiremos dos pines en cada condición, ya que cada actuador es comandado por dos pines, uno hará de señal de alimentación y el otro de señal de tierra. Para ello determinaremos que al ser recibido un mensaje de orden un pin pase a estado activo o **HIGH** y el otro pase o permanezca en estado pasivo o **LOW**. Por ejemplo, en el primer actuador cuyos pines a utilizar son el 2 y el 3, cuando se reciba un mensaje de orden con la letra *e* el pin 2 hará de alimentación pasando o permaneciendo en **HIGH** y el pin 3 hará de tierra pasando o permaneciendo en **LOW**, esto hará que el actuador se accione y gire en un sentido dependiendo de la polaridad de su conexión. Si se desea que gire o se accione de manera opuesta se debe establecer otra condición en la que los pines 2 y 3 cambien de papeles cuando un mensaje de orden con otra letra sea recibida, en este caso será la letra *f*, se hace uso de **delay** con un valor de 200 ms. Estableceremos la misma condición para el accionamiento de todos los actuadores utilizando los demás pines de par en par.

Para el accionamiento simultáneo de los cuatro actuadores estableceremos condiciones donde se involucre uno de los dos pines de cada actuador para configurarlo como alimentación mientras que el otro pin hará de tierra. Para ello estableceremos la misma condición, pero esta vez declararemos todos los pines, identificando el par que le corresponde a cada actuador determinaremos cuales harán de señal de alimentación y cuales harán de señal de tierra. En este caso los pines de número par (2, 4, 6, 8) harán de alimentación y los pines impares (3, 5, 7, 9) harán de tierra, cuando se reciba un mensaje de orden con valor *m*. Para determinar lo contrario simplemente usaremos otra letra, la letra *l*, y configuramos los pines opuestamente a la condición anterior. De esta manera los actuadores se accionarán en uno u otro sentido simultáneamente.

Por último determinaremos que cada vez que se cumpla una acción después de recibir una orden de la aplicación se limpie lo almacenado en Mensaje para recibir y leer otra orden haciendo uso de las comillas "".

```
if (Mensaje == "e"){
    digitalWrite(2,HIGH);
    digitalWrite(3,LOW);
    delay(200);
}
if (Mensaje == "f"){
    digitalWrite(2,LOW);
    digitalWrite(3,HIGH);
    delay(200);
}
```

```
if (Mensaje == "m"){
    digitalWrite(2,LOW);
    digitalWrite(3,HIGH);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
    digitalWrite(6,LOW);
    digitalWrite(7,HIGH);
    digitalWrite(8,LOW);
    digitalWrite(9,HIGH);
    delay(200);
}
```

```
if (Mensaje == "n"){
    digitalWrite(2,HIGH);
    digitalWrite(3,LOW);
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    digitalWrite(8,HIGH);
    digitalWrite(9,LOW);
    delay(200);
}
```

Mensaje="";

Una vez culminada la programación ésta debe ser compilada en el software para su verificación de correcta escritura y utilización adecuada de comandos, luego se debe cargar el programa al módulo Arduino para las respectivas pruebas de funcionamiento.

7. EQUIPOS Y MATERIALES:

EQUIPO DE LABORATORIO	MATERIAL Y EQUIPO DE APOYO
	COMPUTADOR
	SOFTWARE ARDUINO
	SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH
	INFORME DE PRACTICA

8. CONDICIONES DE SEGURIDAD:

- Para el ingreso al laboratorio, el estudiante debe portar mandil azul y su respectiva identificación.
- No ingresar alimentos, bebidas, bebidas alcohólicas, ni fumar dentro o alrededor del laboratorio.
- Se prohíbe estrictamente el ingreso a prácticas habiendo ingerido antes bebidas alcohólicas.
- Utilizar equipos con responsabilidad y bajo la supervisión del Docente o Responsable del laboratorio.
- Bajo ninguna circunstancia se deberá dejar basura en la mesa, piso o cajones de las mesas de trabajo, para esto se cuenta con depósitos apropiados.
- Si el usuario detecta cualquier anomalía en los quipos, tiene que informar inmediatamente al responsable del laboratorio para que se pueda solucionar de manera oportuna.
- Cumplir con las señales, avisos de seguridad y manejo de equipos.
- En el caso de sufrir algún accidente comunicar inmediatamente al Docente o al responsable del laboratorio.

9. EVALUACIÓN DEL APRENDIZAJE:

Resultados de aprendizaje de la carrera	Resultados de aprendizaje de la asignatura	Nivel del aprendizaje de la práctica
<i>Integra las tecnologías de área eléctrica, mecánica, electrónica, control y automatización en el manejo y control de procesos de producción.</i>	<i>1.1 Aplica los principios algebraicos que rigen el funcionamiento de los circuitos digitales combinacionales y secuenciales.</i>	/A

*Alto(A), Medio (M), Bajo (B)

10. EVALUACIÓN DE RESULTADOS OBTENIDOS:

Resultados obtenidos	SI	NO	Observaciones
<i>El estudiante ha comprobado mediante el diseño, simulación y armado de los CLC, la aplicación de los aspectos teóricos con la practica la aplicación de los conocimientos parte profesional.</i>	X		

11. BIBLIOGRAFÍA:

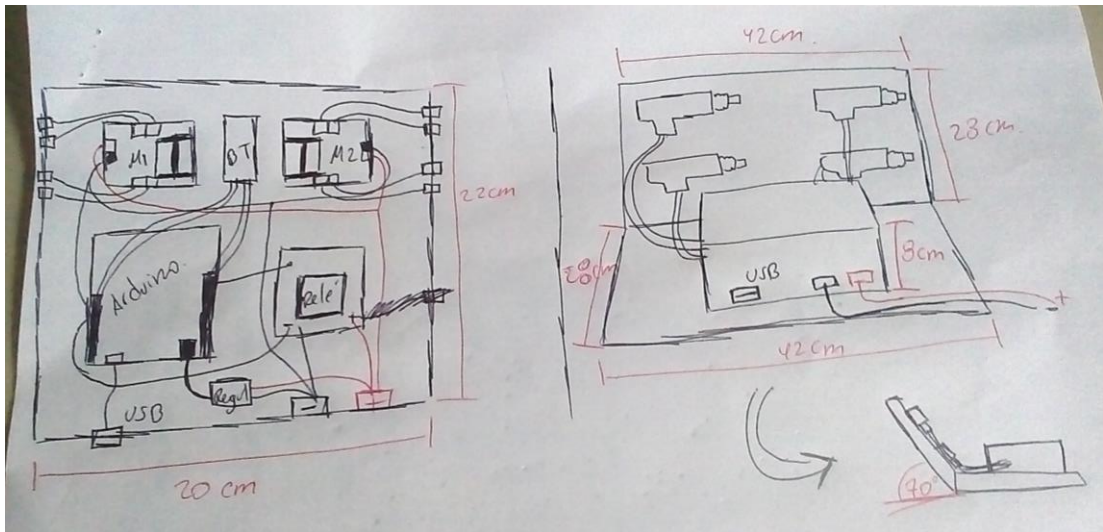
	Tipo (básica o complementaria)
	Bibliografía Básica
	Bibliografía Básica
	Bibliografía complementaria
	Bibliografía complementaria

12. RÚBRICA DE EVALUACIÓN DE PRÁCTICA

ASPECTOS A EVALUAR	PUNTUACIÓN
PRESENTACIÓN DEL PROGRAMA Y ENVÍO AL MÓDULO ARDUINO DEL SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH (funcionando correctamente)	2 puntos
Presentación del informe	2 puntos
Preguntas realizadas por el profesor	2 puntos
Exposición del trabajo realizado	4 puntos
Total	10 Puntos

<p>Elaborado por:</p>  <p>Ing. Elvis Patricio Gualotuña Quishpe. DOCENTE</p> <p>Fecha: 16/10/2018</p>	<p>Revisado por:</p>  <p>Ing. Elvis Patricio Gualotuña RESPONSABLE DE ÁREA</p> <p>Fecha: 15/10/2018</p>	<p>Aprobado por:</p> <p>Ing. Xabier Orbea COORDINADOR DE CARRERA (E)</p> <p>Fecha: 15/10/2018</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------

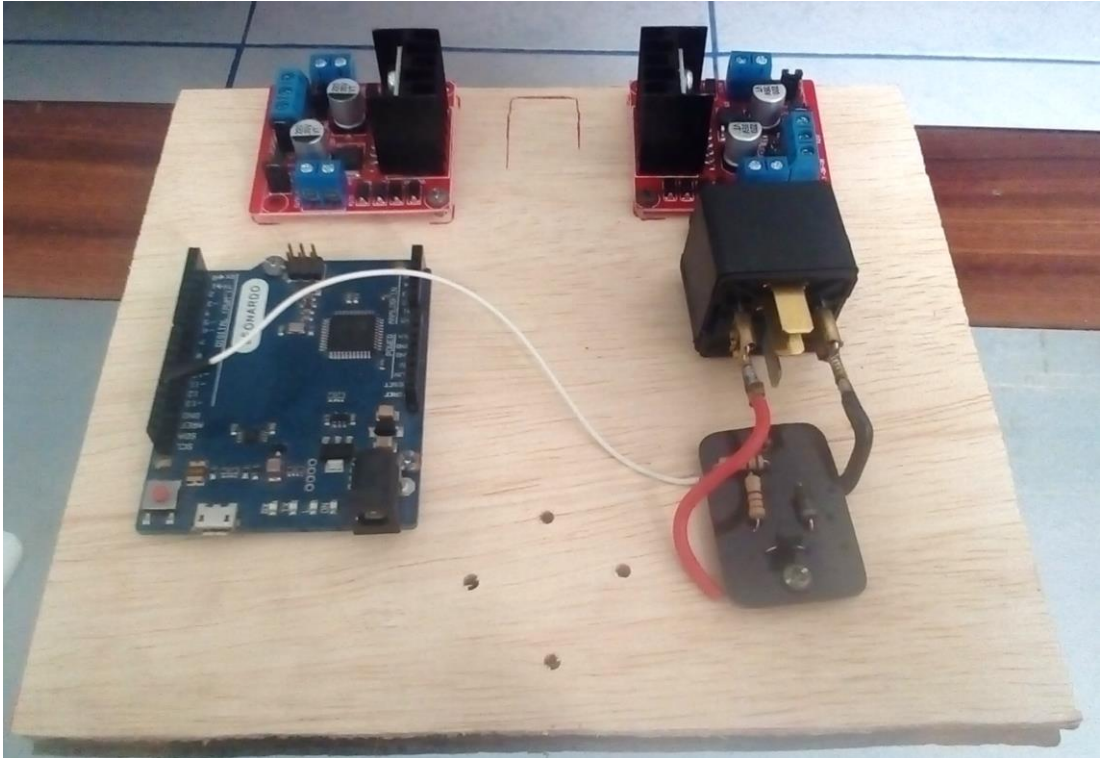
ANEXO 3. BOCETO INICIAL DE LA MAQUETA



ANEXO 4. DISEÑO Y CONTRUCCÓN DE LA CUBIERTA EN MADERA



ANEXO 5. DISTRIBUCIÓN DE LOS DISPOSITIVOS PARA TOMA DE MEDIDAS



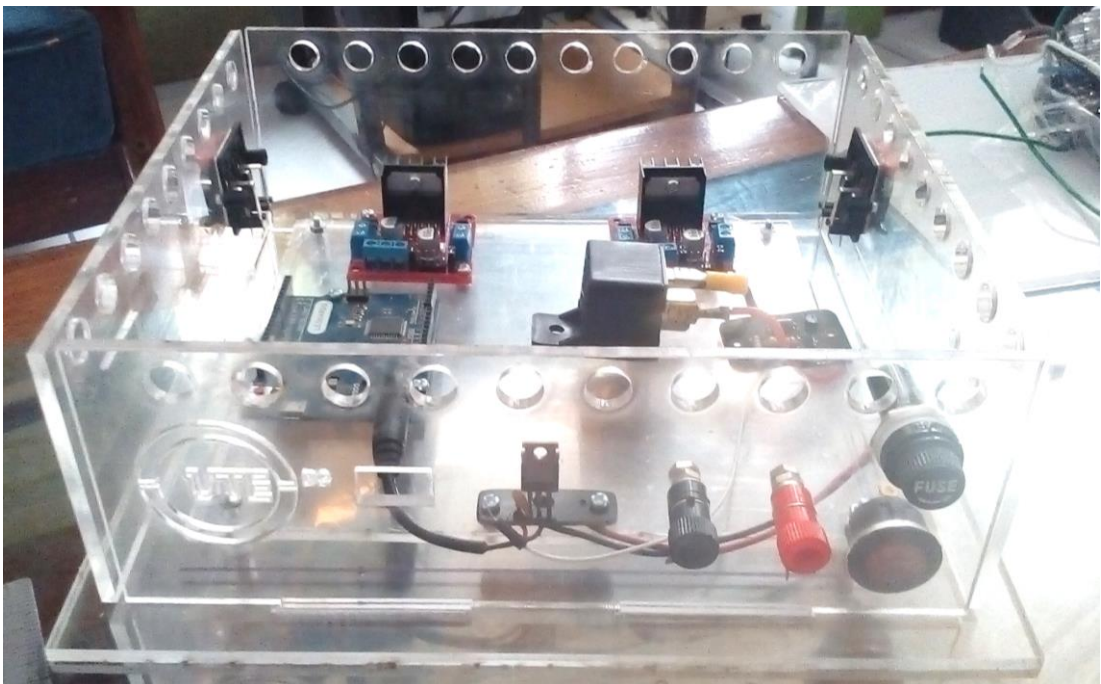
ANEXO 6. CONSTRUCCIÓN DE LA CUBIERTA DE ACRÍLICO



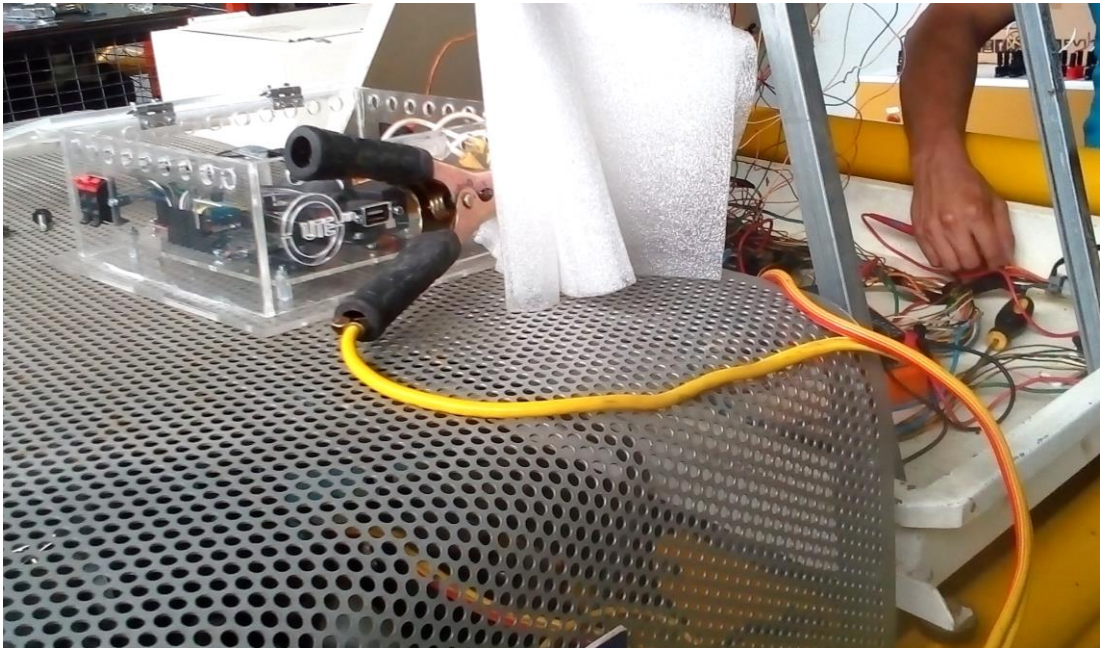
ANEXO 7. PERFORADO EN LA CUBIERTA DE ACRÍLICO



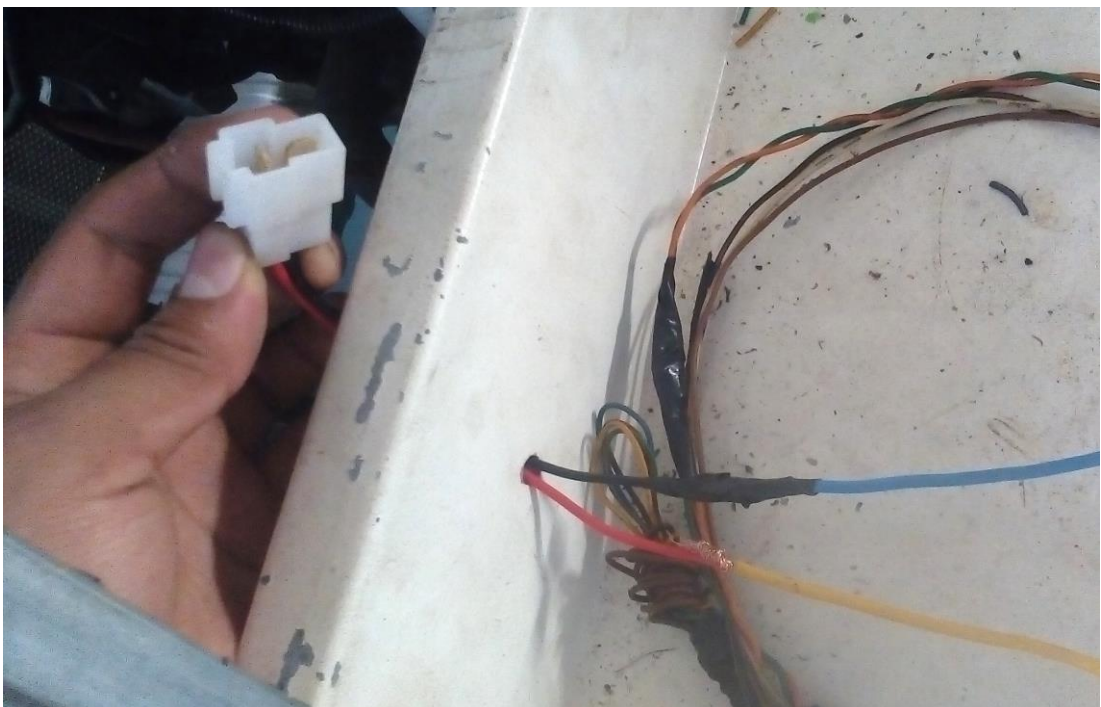
ANEXO 8. MONTAJE DE LOS DISPOSITIVOS EN LA CUBIERTA



ANEXO 9. CONEXIÓN Y PRUEBA DE FUNCIONAMIENTO EN EL BANCO DE PRUEBAS



ANEXO 10 . INSTALACIÓN DE SOQUET EN EL BANCO DE PRUEBAS



ANEXO 11. SISTEMA DE ENCENDIDO Y BLOQUEO CENTRAL VÍA BLUETOOTH TERMINADO

