



**ESCUELA SUPERIOR POLITÉCNICA DE  
CHIMBORAZO**

**FACULTAD DE MECÁNICA  
ESCUELA DE INGENIERÍA AUTOMOTRIZ**

**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE  
LLAMADA DE AUXILIO AUTOMÁTICA EN CASO DE  
ACCIDENTE DE TRÁNSITO”**

**CÉSAR GERARDO AVILÉS ABARCA**

**TRABAJO DE TITULACIÓN**

**Previa a la obtención del Título de:**

**INGENIERO AUTOMOTRIZ**

**RIOBAMBA – ECUADOR**

**2017**

**CERTIFICADO DE APROBACIÓN DEL TRABAJO**  
**DE TITULACIÓN**

**2016-05-09**

Yo recomiendo que el Trabajo de Titulación preparado por:

**CÉSAR GERARDO AVILÉS ABARCA**

Titulada:

**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE LLAMADA DE AUXILIO**  
**AUTOMÁTICA EN CASO DE ACCIDENTE DE TRÁNSITO”**

Sea aceptado como parcial complementación de los requerimientos para el Título de:

**INGENIERO AUTOMOTRIZ**

**Ing. Carlos José Santillán Mariño**  
**DECANO DE LA FAC. DE MECÁNICA**

Nosotros coincidimos con esta recomendación:

**Ing. Wilson Javier Villagrán Cáceres**  
**DIRECTOR**

**Ing. Luis Fernando Buenaño Moyano**  
**ASESOR**

---

---

**EXAMINACIÓN DEL TRABAJO DE TITULACIÓN**

---

---

**NOMBRE DEL ESTUDIANTE: CÉSAR GERARDO AVILÉS ABARCA**

**TRABAJO DE TITULACIÓN: “DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE LLAMADA DE AUXILIO AUTOMÁTICA EN CASO DE ACCIDENTE DE TRÁNSITO”**

**Fecha de Examinación:** 17 de Julio del 2017

**RESULTADO DE LA EXAMINACIÓN:**

<b>COMITÉ DE EXAMINACIÓN</b>	<b>APRUEBA</b>	<b>NO APRUEBA</b>	<b>FIRMA</b>
Ing. José Francisco Pérez Fiallos <b>PRESIDENTE TRIB. DEFENSA</b>			
Ing. Wilson Javier Villagrán Cáceres <b>DIRECTOR DE TESIS</b>			
Ing. Luis Fernando Buenaño Moyano <b>ASESOR DE TESIS</b>			

\* Más que un voto de no aprobación es razón suficiente para la falla total.

**RECOMENDACIONES:** \_\_\_\_\_

El Presidente del Tribunal certifica que las condiciones de la defensa se han cumplido.

---

Ing. José Francisco Pérez Fiallos  
**PRESIDENTE DEL TRIBUNAL**

## **DERECHOS DE AUTORÍA**

El trabajo de grado que presentamos, es original y basado en el proceso de investigación y/o adaptación tecnológica establecido en la Facultad de Mecánica de la Escuela Superior Politécnica de Chimborazo. En tal virtud, los fundamentos teóricos - científicos y los resultados son de exclusiva responsabilidad de los autores. El patrimonio intelectual le pertenece a la Escuela Superior Politécnica de Chimborazo.

---

**César Gerardo Avilés Abarca**

## **DEDICATORIA**

Este trabajo está dedicado para mi familia que han sido parte fundamental en mi vida estudiantil, mis padres que me apoyaron en todo momento ya sea económica como emocionalmente , en especial a mi madre quien nunca hizo faltar esas palabras de cariño y apoyo para continuar con mis estudios y poder terminarlos.

Mis abuelos César y Rosario que me impulsaron a seguir con sus consejos y palabras de aliento.

Mis tíos Jack y Mercedes que con sus consejos muchas veces me hicieron que escoja el camino correcto y hoy estoy culminando mis estudios.

De igual manera para mi esposa que ha sido, es y será mi apoyo incondicional en todo momento y como no a mi hija Vitoria que es la motivación que tengo cada día para seguir adelante.

**César Gerardo Avilés Abarca**



## **DECLARACIÓN DE AUTENTICIDAD**

Yo, César Gerardo Avilés Abarca, declaro que el presente trabajo de titulación es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación.

---

**César Gerardo Avilés Abarca**

## **AGRADECIMIENTO**

Agradezco a Dios por haberme permitido llegar a la finalización de mi carrera, cuidarme en todo momento y sobre todo regalarme salud, a mis padres que son pilar fundamental en mi vida , gracias por brindarme su apoyo y cariño en todo momento en especial a mi madre ejemplo de lucha y perseverancia.

A mi esposa Nicole por ser darme esas palabras de aliento y animo cundo lo necesite, a mis amigos con los que compartimos momentos gratos durante la vida universitaria.

Por último quiero agradecer a mis maestros en especial al Ing. Javier Villagrán que a más de ser mi maestro fue un amigo y apoyo para mí.

A todas los familiares y amigos que no nombre al momento de escribir ustedes saben quiénes son.

**César Gerardo Avilés Abarca**

# CONTENIDO

Pág.

<b>1.</b>	<b>INTRODUCCIÓN</b>	
1.1	Antecedentes .....	1
1.2	Objetivos .....	2
1.2.1	<i>Objetivo general.</i> .....	2
1.2.2	<i>Objetivos específicos.</i> .....	2
<b>2.</b>	<b>MARCO TEÓRICO</b>	
2.1	Definición de accidente de tránsito .....	4
2.1.1	<i>Factores que intervienen en un accidente de tránsito.</i> .....	4
2.2	Funcionamiento del sistema de llamada de auxilio automática. ....	4
2.3	Definición de ADAS. ....	5
2.4	Seguridad Activa en el vehículo .....	5
2.4.1	<i>Elementos de seguridad activa.</i> .....	5
2.5	Seguridad Pasiva. ....	5
2.5.1	<i>Elementos de seguridad pasiva.</i> .....	6
2.6	Elementos de un sistema electrónico en el vehículo.....	6
2.7	Definición de Sensor. ....	7
2.7.1	<i>Tipos de sensores.</i> .....	7
2.7.1.1	<i>Sensor de impacto.</i> .....	8
2.8	Definición de Actuador. ....	8
2.8.1	<i>Tipos de actuadores.</i> .....	9
2.9	ECU.....	9
2.10	El micro controlador de Arduino .....	10
2.10.1	<i>Elementos de la placa.</i> .....	10
2.10.1.1	<i>Puerto USB.</i> .....	11
2.10.1.2	<i>Microprocesador.</i> .....	11
2.10.1.3	<i>Botón Reset.</i> .....	11
2.10.1.4	<i>Pines de entrada y salida.</i> .....	11
2.10.2	Elementos del microcontrolador .....	11
2.10.3	Entradas y Salidas. ....	12
2.10.4	Señales analógicas y digitales. ....	13
2.11	Funcionamiento del microcontrolador .....	13
2.12	Circuitos para conectar las entradas y salidas .....	14
2.13	Software de comunicación .....	15
2.13.1	Programación. ....	16
2.13.1.1	Declaración de variables. ....	16
2.13.1.2	Void setup{ }. ....	16
2.14	Sistema GPS.....	17
2.15	Comunicación GSM.....	17
2.16	Comunicación GPRS .....	17
2.17	Shields para Arduino.....	18
<b>3.</b>	<b>DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO</b>	
3.1	Metodología .....	20

3.1.1	Definición de necesidades. ....	21
3.1.2	Caracterización del prototipo.....	21
3.1.3	Estudio de los dispositivos disponibles. ....	22
3.1.3.1	Arduino mega 2560.....	22
3.1.4	IDE (software) de Arduino. ....	24
3.1.5	Arduino y módulo bluetooth.....	25
3.1.6	Arduino y LCD 12C. ....	27
3.1.7	Módulos I2C. ....	28
3.1.8	Puente H LN298 .....	28
3.1.9	Arduino – I/O digital.....	30
3.1.10	Arduino GPS.....	30
3.1.11	Tarjeta SIM 900 GSM/GPRS. ....	33
<b>4.</b>	<b>CONSTRUCCIÓN DEL SISTEMA DE LLAMADA DE EMERGENCIA.</b>	
4.1	Esquema general de conexiones del auto prototipo.....	35
4.1.1	Programación del prototipo. ....	36
4.1.2	LiquidCrystal lcd(13, 12, 8, 5, 4, 3).....	37
4.1.3	Conexión Arduino y LCD 12C.....	41
4.1.4	Arduino & Módulo Puente H L298N. ....	43
4.1.5	Arduino y módulo bluetooth HC-05.....	44
4.2	Resultados Obtenidos.....	45
4.3	Pruebas realizadas.....	45
<b>5.</b>	<b>CONCLUSIONES Y RECOMENDACIONES</b>	
5.1	Conclusiones .....	46
5.2	Recomendaciones .....	<b>¡Error! Marcador no definido.</b>

## ANEXOS

## LISTA DE TABLAS

	<b>Pág.</b>
<b>Tabla 1-2.</b> Elementos digitales .....	14
<b>Tabla 2-2.</b> Lenguaje de programación Arduino .....	16

## LISTA DE FIGURAS

Pág.

<b>Figura 1-2.</b> Funcionamiento del sistema de llamada automática .....	4
<b>Figura 2-2.</b> Seguridad activa .....	5
<b>Figura 3-2.</b> Seguridad pasiva .....	6
<b>Figura 4-2.</b> Elementos de un sistema electrónico en el vehículo .....	7
<b>Figura 5-2.</b> Sensor.....	7
<b>Figura 6-2.</b> Sensor de impacto-airbags .....	8
<b>Figura 7-2.</b> Tipos de actuadores.....	9
<b>Figura 8-2.</b> ECU.....	10
<b>Figura 9-2.</b> Placa.....	11
<b>Figura 10-2.</b> Microcontrolador Arduino .....	12
<b>Figura 11-2.</b> Entradas y salidas .....	13
<b>Figura 12-2.</b> Función Digital Write.....	13
<b>Figura 13-2.</b> Función DigitalRead .....	14
<b>Figura 14-2.</b> Shields para Arduino.....	18
<b>Figura 1-3.</b> Metodología para diseño y construcción del prototipo .....	20
<b>Figura 2-3.</b> Chasis del prototipo.....	21
<b>Figura 3-3.</b> Conexión Arduino Mega.....	22
<b>Figura 4-3:</b> IDE Arduino, Selección de la Placa Arduino Mega2560 .....	24
<b>Figura 5-3:</b> IDE Arduino, Selección del Puerto COM.....	25
<b>Figura 6-3.</b> Módulo Bluetooth HC-05.....	25
<b>Figura 7-3.</b> Conexión Arduino Mega y LCD con módulo I2C .....	27
<b>Figura 8-3.</b> LCD i2C 16x2 .....	28
<b>Figura 9-3.</b> Terminales puente H, módulo de Arduino .....	28
<b>Figura 10-3.</b> Conexiones de alimentación al puente H, módulo de Arduino .....	29
<b>Figura 11-3.</b> Sensor / Entrada Digital .....	30
<b>Figura 12-3.</b> Sensor / Entrada Digital .....	31
<b>Figura 13-3.</b> Diagrama esquemático de la tarjeta de control.....	32
<b>Figura 14-3.</b> Conexiones.....	33
<b>Figura 15-3.</b> Shield Sim900 GSM/GPRS .....	33
<b>Figura 1-4.</b> Esquema de conexiones eléctricas en el prototipo .....	35
<b>Figura 2-4.</b> Conexiones prototipo / Sistema centralizado con Arduino Mega 2560 .....	36
<b>Figura 3-4.</b> Conexión Arduino Mega y módulo LCD I2C 16x2.....	42
<b>Figura 4-4.</b> Pantalla de visualización en el LCD.....	42
<b>Figura 5-4.</b> Montaje LCD en el prototipo .....	43

<b>Figura 6-4.</b> Conexión Arduino y módulo L298N.....	43
<b>Figura 7-4.</b> Conexión Puente H LN298 motores delanteros.....	44
<b>Figura 8-4.</b> Conexión Arduino Mega y módulo Bluetooth H-05.....	44
<b>Figura 9-4.</b> Conexión módulo bluetooth.....	45
<b>Figura 10-4.</b> Interfaz de control Android.....	45
<b>Figura 11-4.</b> Análisis de confiabilidad.....	47

## **LISTA DE ABREVIACIONES**

GPS	Global positioning system
GPRS	General Packet Radio Service
IDE	Software de Arduino
ADAS	Advanced Driver Assistance System
USB	Universal Serial Bus
LED	Light-Emitting Diode

## LISTA DE ANEXOS

❖ A.- Prototipo.....	50
❖ B.- Prototipo.....	50
❖ C.- Prototipo.....	50
❖ D.- Prototipo.....	50
❖ E.- Prototipo.....	50

## RESUMEN

Se ha diseñado y construido un sistema de llamada de auxilio en caso de accidente de tránsito, basado en los sistemas avanzados de asistencia a la conducción ADAS (Advanced Driver Assistance Systems). El objetivo del sistema es proporcionar información del siniestro a los organismos de socorro para que estos puedan brindar auxilio médico en el menor tiempo posible. El sistema utiliza un módulo de control electrónico que recibe las señales de los diferentes sensores ubicados en el vehículo, en nuestro caso consta de 4 sensores. El módulo procesa la información recibida para realizar una llamada de auxilio seguido de un mensaje de texto con las coordenadas del vehículo al ecu 911, además podemos visualizar en nuestro vehículo las coordenadas mediante una pantalla lcd. Los elementos que componen el sistema son: Arduino como módulo de control, fines de carrera, Pantalla Serial LCD, sistema GPS, sistema Gsm, resistencias e interruptores. Para desarrollar el software usamos el compilador del sistema Arduino que nos permite configurar nuestro sistema electrónico con las condiciones necesarias, tanto el software como el hardware son de libre acceso. El sistema es capaz de activarse tanto con golpes leves como con golpes a mayor velocidad. Mediante pruebas realizadas al sistema en 30 escenarios diferentes y con la ayuda de un GPS adicional se logró el objetivo deseado ya que las llamadas y los mensajes con las coordenadas del accidente fueron las correctas, además se obtuvo un factor de confiabilidad del 87%. Logrando implementar de manera exitosa el sistema en nuestro prototipo siendo un sistema confiable en diferentes condiciones y en cualquier parte de nuestro país. El sistema puede implementarse a gran escala en nuestro país contribuyendo a la seguridad de nuestros compatriotas en las carreteras. Se recomienda tener recargado con saldo el sistema debido a que sin este factor el sistema no funciona.

PALABRAS CLAVE: <INGENIERÍA DE SOFTWARE>, <SISTEMA DE LLAMADA DE AUXILIO AUTOMÁTICO EN CASO DE ACCIDENTE DE TRÁNSITO>, <SISTEMA ARDUINO>, <SISTEMA ELECTRÓNICO>, <ANÁLISIS DE CONFIABILIDAD>.

## **ABSTRACT**

A system of call for help in cases of traffic accidents has been designed and built, based on ADAS (Advanced Driver Assistance System). The purpose of the system is to provide disaster information to the relief agencies so that they can provide medical assistance in the shortest possible time. The system utilizes an electronic control module that receives signals from the different sensors located in the vehicle, in the particular case of this inquiry it consists of 4 sensors. The module processes the received information to make a distress call followed by a text message with the coordinates of the vehicle to the ecu 911, moreover it is possible to visualize in the vehicle the coordinates through an LCD screen. The elements that make up the system are as follows: Arduino as control module, limit switches, LCD serial display, GPS system, GSM system, resistors and switches. To develop the software it is used the compiler of the Arduino system which allows configuring the electronic system with the necessary conditions; both the software and the hardware are freely accessible. The system is able to be activated both with light strokes and with faster strokes. By testing the system in 30 different scenarios and with the help of an additional GPS the desired objective was achieved as the calls and messages with the coordinates of the accident were correct, furthermore a reliability factor of 87% was obtained, succeeding in implementing successfully the system in our prototype being a reliable system in different conditions and in any part of our country. The system can be implemented on a large scale in our country contributing to the safety of our compatriots on the roads. It is recommended to have the system provided with available mobile phone credit because without this factor the system does not work.

**KEY WORDS:** <SOFTWARE ENGINEERING> , <AUTOMATIC SYSTEM OF CALL FOR HELP IN CASES OF TRAFFIC ACCIDENTS>, <ARDUINO SYSTEM <ELECTRONIC SYSTEM>, <RELIABILITY ANALYSIS>

# CAPÍTULO I

## 1. INTRODUCCIÓN

### 1.1 Antecedentes

La llamada automática de emergencia o “eCall” consiste en un dispositivo inteligente de llamada de emergencia que puede ser activado manualmente por los ocupantes del vehículo o automáticamente en caso de accidente de tráfico gracias a sensores instalados en el interior del vehículo. La activación del sistema eCall produce una llamada de alerta localizable por el servicio de urgencias 112, sea cual sea el país de la Unión Europea en el que se haya producido el accidente o la emergencia. Ello permite activar con el menor retraso posible los servicios de socorro y asistencia.

La activación de la alerta de emergencia, manual o automáticamente, pone en marcha un protocolo de actuación que arranca con el envío de una llamada que consta de dos elementos: una conexión de voz a través del número 112 y un paquete de datos esenciales entre los que se encuentra la localización exacta del vehículo por medio de tecnología de localización o posicionamiento global por satélite (en estos momentos GPS y en el futuro Galileo).

Las distintas investigaciones desarrolladas alrededor de este sistema de llamada de emergencia a bordo llegan a la conclusión de que uno de los factores más importantes que contribuyen a la gravedad de las lesiones producidas en accidentes de tráfico es el tiempo de respuesta de los servicios de emergencia para encontrar y atender a los heridos. En dicho tiempo de respuesta influye un cúmulo de circunstancias, como la organización y la coordinación del equipo que recibe las llamadas, el tiempo que transcurre desde el momento en que se recibe la llamada hasta que se alerta al servicio de emergencia oportuno y las precisión y rapidez con que es posible localizar con exactitud el lugar del accidente, entre otros.

Por otra parte, de un estudio realizado en 1995 en Alemania, en el área de Stuttgart, sobre “Evaluación de la influencia de los Sistemas Telemáticos en la accidentalidad”, se desprende que con un sistema eCall los tiempos de rescate se mejoran en un 43 por

ciento, son igualmente datos reveladores los resultados del sistema japonés HELPNET Service que fue puesto en funcionamiento como servicio de emergencia de pago en septiembre del año 2000, este servicio ha demostrado que la probabilidad de salvar la vida a una persona que ha sufrido una parada respiratoria aumenta con dicho sistema en un 45%, mientras que el caso de infarto cardiaco lo hace en un 25%. El sistema también consigue una la mejora del 2% en la congestión de tráfico.

El presente trabajo busca construir un sistema de llamada de auxilio automática en caso de accidente de tránsito, el cual se está llevando a cabo en la Escuela Superior Politécnica de Chimborazo con la Carrera de Ingeniería Automotriz de esta manera se contribuye a la sociedad y a futuras generaciones.

Este sistema actualmente no existe en el país, sumado a esto el alto índice de accidentes de tránsito que nos deja alarmantes estadísticas de pérdidas humanas solo entre enero y junio del año 2015 un promedio de 5,7 personas murieron diariamente en nuestro país a esto se añade 12.415 personas heridas, en este mismo tiempo, en 2014, las autoridades registraron 19.842 accidentes de diferente magnitud, que dejaron 1.276 víctimas mortales y 14.612 heridos. Por esta razón observamos la necesidad de iniciar nuestro proyecto y de esta manera reducir el número de víctimas mortales debido a la falta de auxilio inmediato.

## **1.2      Objetivos**

**1.2.1**      *Objetivo general.* Diseñar y construir un sistema de llamada de auxilio automática en caso de accidente de tránsito con la ayuda de sensores de impacto, para disminuir el número de víctimas mortales en accidentes de tránsito.

**1.2.2**      *Objetivos específicos:*

- Revisar tipos de sensor de impacto, mediante fuentes bibliográficas para determinar cuál es el más óptimo para nuestro sistema.
- Seleccionar el sensor adecuado para nuestro prototipo , para poder realizar las pruebas correspondientes en el prototipo.

- Diseñar el circuito del prototipo utilizando un software de programación, para realizar el control del sistema.
- Construir el prototipo utilizando los componentes necesarios para comprobar su funcionamiento.
- Realizar las pruebas necesarias mediante ensayos de funcionamiento para demostrar la efectividad del prototipo.

## CAPÍTULO II

### 2. MARCO TEÓRICO

#### 2.1 Definición de accidente de tránsito

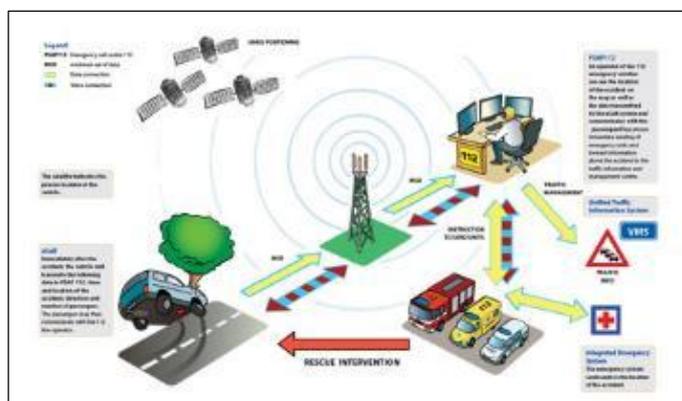
Colisión u otro tipo de impacto dentro de la vía pública que causa muerte, lesión u otro daño (SÁNCHEZ, 2013).

Según la OMS Ecuador, ocupa el segundo lugar en mortalidad por accidentes de tránsito en toda América Latina, según un informe sobre la seguridad de las vías de la Organización Mundial de la Salud.

**2.1.1 Factores que intervienen en un accidente de tránsito.** Un accidente de tránsito, es ocasionado por 3 factores principales que son: el factor humano, factor vehicular y factor climático.

#### 2.2 Funcionamiento del sistema de llamada de auxilio automática.

El sistema detecta si un suceso es grave gracias a unos sensores instalados en el vehículo, y en ese momento se graba un mensaje de auxilio que llegará a la centralita del ECU 911, mediante un sistema de GPS, con el que se podrá detectar fácilmente donde se encuentra el vehículo. La llamada podrá ser realizada de manera manual por los ocupantes o automáticamente en caso de accidente grave.



**Figura 1-2.** Funcionamiento del sistema de llamada automática

Fuente: <http://www.seguridad-vial.net/vehiculo/seguridad-pasiva/120-que-es-el-e-call>

## 2.3 Definición de ADAS.

Siglas de Sistemas Avanzados de Asistencia a la Conducción (en inglés, Advanced Driver Assistance System). Denominación de Honda que engloba a un conjunto de tres tecnologías encaminadas a mejorar la seguridad activa.

Hay muchas formas de ADAS disponibles; algunas características están integradas en los coches o están disponibles como un paquete adicional. También, hay soluciones del mercado de accesorios disponibles (TELEMATIC NEWS, 2012).

ADAS se basa en las aportaciones de múltiples fuentes de datos, incluyendo imágenes del automóvil, radar, procesamiento de imágenes, la creación de redes de visión por ordenador, y en el automóvil (TELEMATIC NEWS, 2012).

## 2.4 Seguridad Activa en el vehículo

La seguridad activa en el vehículo se considera a todos los elementos o componentes cuyo principal objetivo sea evitar que se produzca un accidente de tránsito.

**2.4.1** *Elementos de seguridad activa.* Estos factores son numerosos, y entre ellos pueden destacarse los siguientes: la dirección, el sistema de alumbrado, los frenos, los neumáticos, entre otros (VALCÁRSEL, 2013).



**Figura 2-2.** Seguridad activa

Fuente: <http://www.cars.com.ar/noticia/548/caracteristicas-de-seguridad-activa>

## 2.5 Seguridad Pasiva.

El principio de la seguridad pasiva significa que el vehículo debe proteger en todo momento la integridad física de sus ocupantes cuando por impericia, imprudencia o cualquier otro motivo se produzca un accidente de tránsito.

**2.5.1** *Elementos de seguridad pasiva.* Los elementos que hacen parte de la seguridad activa del vehículo son los encargados de mantener el control del mismo y realizar adecuadamente las acciones que realiza el conductor con el fin de evitar accidentes de tránsito (LOAIZA, 2012). Entre ellos se encuentran: Frenos, neumáticos, suspensión, dirección, iluminación, entre otros.

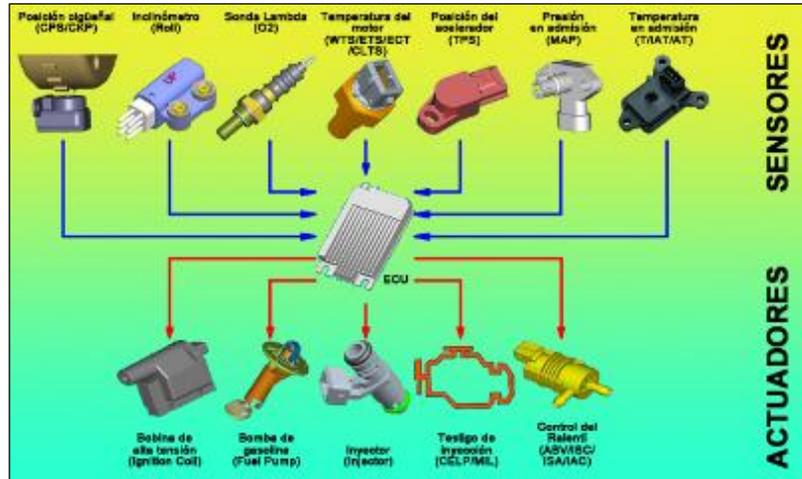


**Figura 3-2.** Seguridad pasiva

Fuente: <http://www.cars.com.ar/noticia/548/caracteristicas-de-seguridad-pasiva>

## **2.6 Elementos de un sistema electrónico en el vehículo**

En el vehículo existen varios sistemas electrónicos, que cumplen tareas específicas. Cada uno de estos sistemas se compone de tres elementos principales que son los sensores, actuadores y una unidad central de procesamiento.



**Figura 4-2.** Elementos de un sistema electrónico en el vehículo

Fuente: <http://www.moto125.cc/f125cc/reportajes/tenica/item/981>

## 2.7 Definición de Sensor.

El sensor (también llamado sonda o transmisor) convierte una magnitud física (temperatura, revoluciones del motor, etc) o química (gases de escape, calidad del aire, etc) en una magnitud eléctrica que pueda ser entendida por la unidad de control (CISNEROS).



**Figura 5-2.** Sensor

Fuente: <http://gerardomontoya2amm.blogspot.com/2012/05/sensor-tps.html>

**2.7.1 Tipos de sensores.** La respuesta que proporciona un sensor depende de la magnitud física que puede ser detectada y “traducida” en una variación eléctrica y del principio físico en que se base. Existen numerosos sensores que miden parámetros muy diversos (SEAT, 1998).

Según el principio de funcionamiento se clasifican en los siguientes tipos:

- Magnético.
- Por efecto hall.
- Por conductividad eléctrica.
- Termoeléctricos.
- Fotoeléctricos.
- Piezoeléctricos.
- Por ultrasonidos.
- Por radiofrecuencia.
- Interruptores y conmutadores.

**2.7.1.1** *Sensor de impacto.* El sensor debe tener la capacidad de detectar un impacto y transformarla en la señal correcta en un lapso muy corto de tiempo. Las magnitudes de aceleración que trabajan en los sensores después de un impacto llegan alcanzar 100 g (100 veces la fuerza gravitatoria terrestre) (VOLKSWAGEN, 2010).

Los sensores de impacto comúnmente aplican el principio de medición por efecto de la inercia. Al momento de un impacto, los objetos que no estén sujetos firmemente al vehículo continúan moviéndose con la misma velocidad de impacto. El circuito interno del sensor es capaz de medir esta aceleración y transmitir la información a la unidad de control (VOLKSWAGEN, 2010).



**Figura 6-2.** Sensor de impacto-airbags

Fuente: (VOLKSWAGEN, 2010)

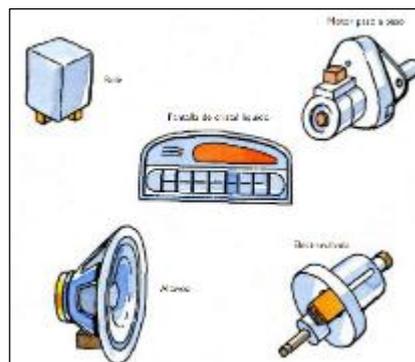
## **2.8 Definición de Actuador.**

Los actuadores son la interfaz que utiliza el tratamiento de la señal (proceso de la información) y el proceso (mecánico). Don los encargados de convertir las señales que

transmiten la información de ajuste, de baja potencia, en señales con alta potencia y con suficiente energía para participar en el proceso. Existen convertidores de señales combinados con elementos amplificadores, estos sustentan su funcionamiento con los principios de transformación física con diferentes tipos de energía (eléctrica – mecánica – hidráulica – térmica).

**2.8.1 Tipos de actuadores.** Los actuadores según su principio básico de funcionamiento se clasifican de la siguiente manera:

- Electromagnéticos
- Calefactores
- Electromotores
- Acústicos
- Pantallas de cristal líquido



**Figura 7-2.** Tipos de actuadores

Fuente: (SEAT, 1998)

## 2.9 ECU

La unidad de control de motor o ECU es la encargada de controlar los parámetros necesarios para el funcionamiento del motor de combustión interna. Existen varios tipos de unidades de control desde las más simples que se encargan de administrar únicamente la cantidad de combustible necesario en cada cilindro y en el tiempo correcto, hasta las más complejas que son capaces de controlar el punto de ignición, el tiempo de apertura/cierre de las válvulas, el tiempo óptimo para el cambio de marcha, además administran otros dispositivos como son dispositivos de seguridad (BOHACZ, 2011).



**Figura 8-2.** ECU

**Fuente:** <http://www.tucocheytu.com/foro/forumdisplay.php/52-Flasheo-de-ECU>

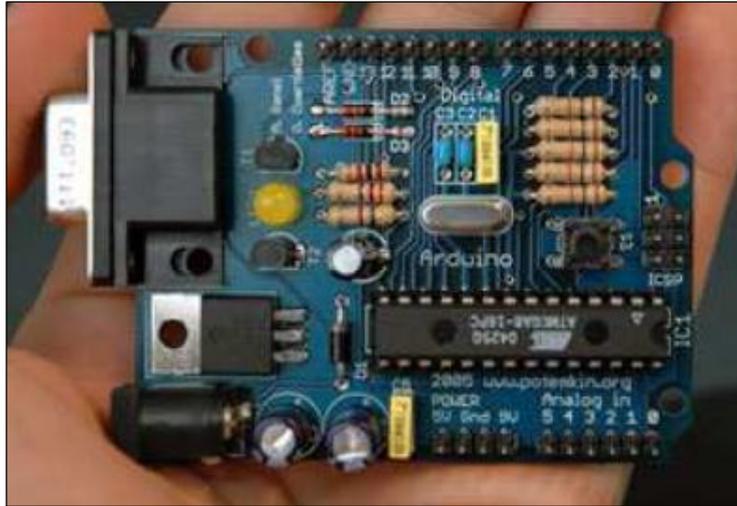
Las unidades de control de motor determinan la cantidad de combustible, el punto de ignición y otros parámetros monitorizando el motor a través de sensores. Estos incluyen: sensor MAP, sensor de posición del acelerador, sensor de temperatura del aire, sensor de oxígeno y muchos otros. Frecuentemente esto se hace usando un control repetitivo (como un controlador PID).

Antes de que las unidades de control de motor fuesen implantadas, la cantidad de combustible por ciclo en un cilindro estaba determinada por un carburador o por una bomba de inyección (BOHACZ, 2011).

## **2.10 El micro controlador de Arduino**

Arduino es una plataforma libre de computación de bajo coste basada en una placa de entrada-salida y en un entorno de desarrollo IDE que implementa el lenguaje Processing/Wiring Hardware. Arduino se puede usar para desarrollar objetos interactivos automáticos o conectarse a software en el ordenador (Pure Data, Flash, Processing; MaxMSP. (ARDUINO, 2009).

**2.10.1 Elementos de la placa.** En la placa de circuito impreso va instalado el microprocesador, la memoria, las conexiones de entrada y salida y la conexión para el puerto USB.



**Figura 9-2.** Placa

**Fuente:** (ARDUINO, 2009)

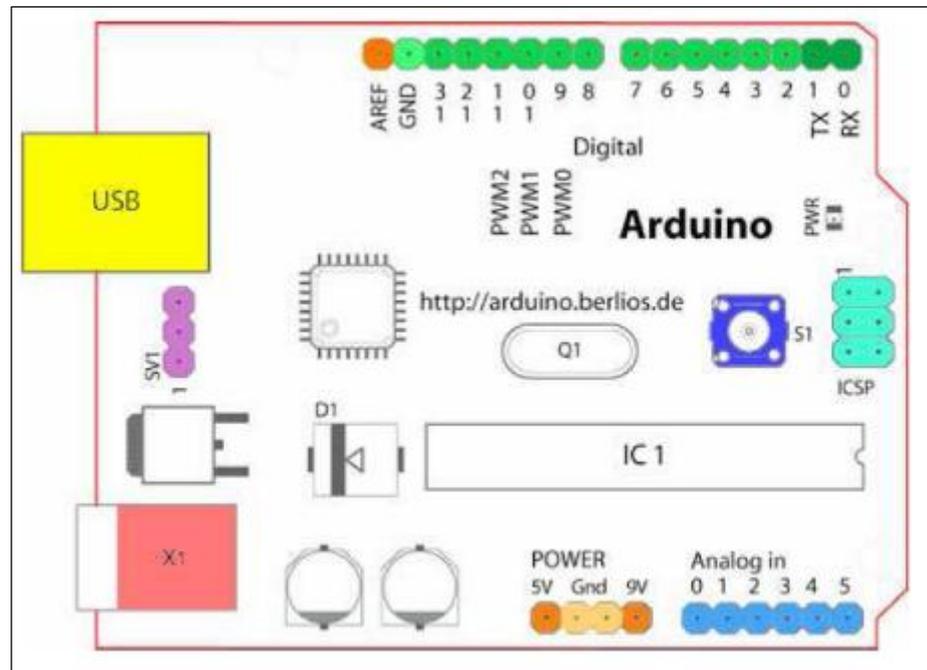
**2.10.1.1** *Puerto USB.* a través de él se cargan las instrucciones a ejecutar, el programa que es realizado en el entorno de programación de Arduino. Comunicación Arduino-Ordenador.

**2.10.1.2** *Microprocesador.* Realiza las instrucciones almacenadas en el programa de forma cíclica. Es un circuito integrado que contiene muchas de las mismas cualidades que una computadora. Escribe en los pines DS2-13 y lee en los DE2-13 AE0-5.

**2.10.1.3** *Botón Reset.* Permite resetear el programa y cargar uno nuevo.

**2.10.1.4** *Pines de entrada y salida.* Permiten crear elementos que dan información y crean actuaciones.

**2.10.2** *Elementos del microcontrolador.* A continuación se detallan cada uno de los elementos del microcontrolador.



**Figura 10-2.** Microcontrolador Arduino

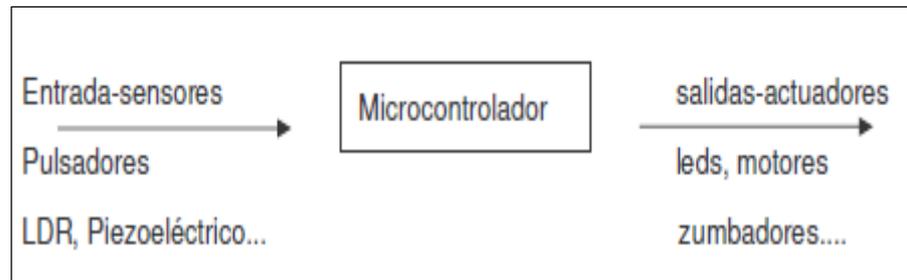
Fuente: (ARDUINO, 2009)

Donde:

- Aref: Pin de referencia analógica (naranja)
- GND: Señal de tierra digital (verde claro)
- Pines digitales 2-13: Entrada y salida (verde)
- Pines digitales 0-1 / entrada y salida del puerto serie: TX/RX (verde oscuro)
- Botón de reset: pulsador (azul oscuro)
- Pines de entrada analógica 0-5 (azul claro)
- Pines de alimentación y tierra (fuerza: naranja, tierra: naranja claro)
- Entrada de la fuente de alimentación externa (9-12V DC) X1 (rosa)
- Puerto USB (amarillo)

**2.10.3 Entradas y Salidas.** El microcontrolador recibe información de las entradas (read), la procesa y escribe un 1 o un 0 (5v o 0v) en las salidas (write), actuando sobre el dispositivo que tenemos conectado.

Al microcontrolador por lo tanto conectamos unos sensores a la entrada y unos actuadores a la salida, para que en función del programa y de la lectura de los sensores se produzcan una serie de actuaciones.



**Figura 11-2.** Entradas y salidas

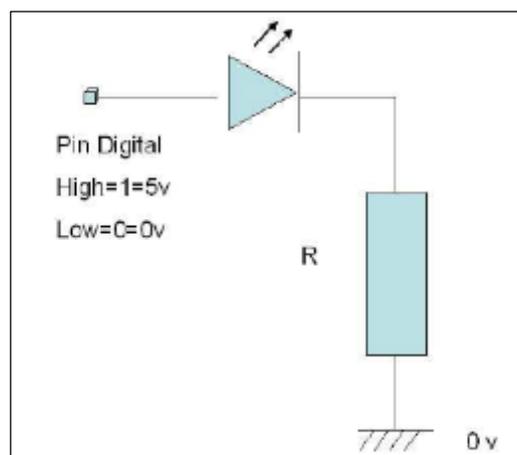
Fuente: (ARDUINO, 2009)

**2.10.4** *Señales analógicas y digitales.* Una señal es analógica cuando puede tomar infinitos valores entre su valor mínimo y máximo. Existen elementos analógicos como el potenciómetro, ldr, motor eléctrico, entre otros.

Una señal es digital cuando solo puede tomar 2 valores, el máximo asociado a 1 o a cierto “on” y el mínimo asociado a cero, falso o “off”. Existen elementos digitales como el pulsador, detector de presencia, entre otros (ARDUINO, 2009).

## 2.11 Funcionamiento del microcontrolador

El microcontrolador está en la función DigitalWrite cuando está como se muestra en la siguiente figura.



**Figura 12-2.** Función Digital Write

Fuente: (ARDUINO, 2009)

- Función Write: pines de salida (escribir)

- High: el microcontrolador escribe un 1 en el pin de salida, es decir 5v. El led se enciende.
- Low: el microcontrolador escribe un 0 en el pin de salida, es decir 0v. El led no enciende.

En la función DigitalRead el microcontrolador lee la tensión en el pin de entrada si  $V > 3,5v$  asigna un 1, si  $V < 3,5v$  asigna un 0.

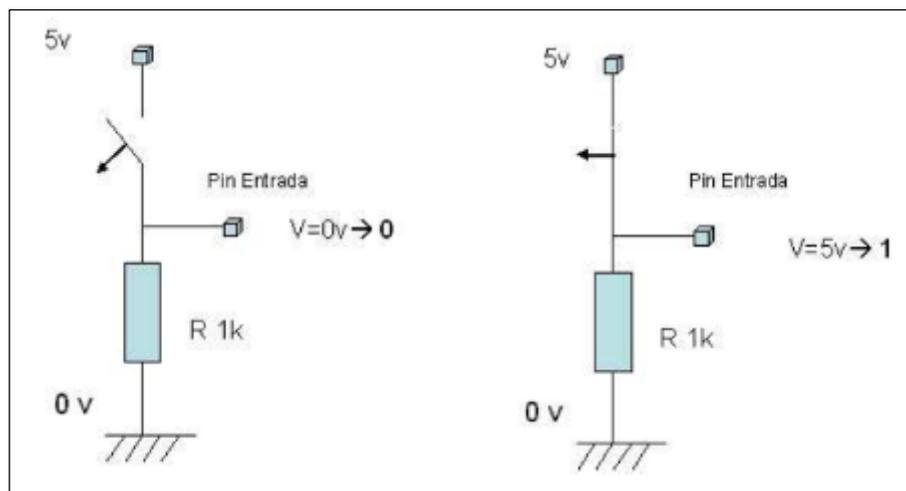


Figura 13-2. Función DigitalRead

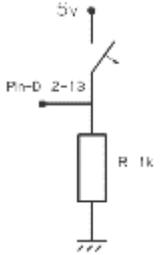
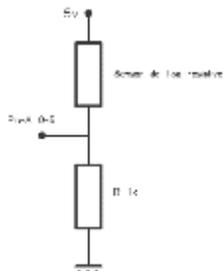
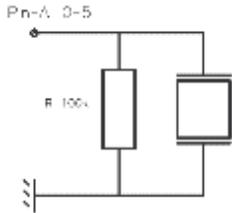
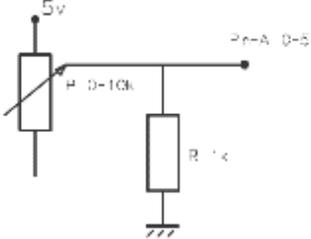
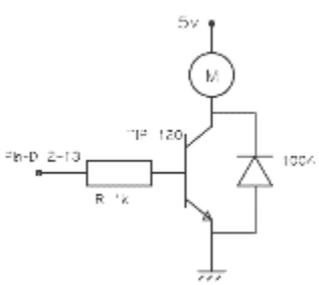
Fuente: (ARDUINO, 2009)

## 2.12 Circuitos para conectar las entradas y salidas

Los circuitos para conectar los principales elementos de entrada y salida son:

Tabla 1-2. Elementos digitales

Elementos	Esquema	Conexión
Leds		Conectar a los pines 2-13 Digitales (El 13 lleva resistencia interna)
Zumbadores		Conectar a los pines 2-13 Digitales. si queremos modular la frecuencia 9-11

Pulsadores		Conectar a los pines 2-13 Digitales
Sensores		Conectar a los pines 0-5 analógicos
Piezoeléctrico		Conectar a los pines 0-5 analógicos
Potenciómetros		Conectar a los pines 0-5 analógicos
Motores		Conectar a los pines 2-13 Digitales. si queremos controlar la velocidad 9-11

Fuente: (ARDUINO, 2009)

## 2.13 Software de comunicación

Para comunicarnos con el microcontrolador y cargarle los programas usamos IDE (Integrated Development Environment), sistema de desarrollo Arduino, sirve para escribir programas, compilarlos y descargarlos a la placa de Arduino.

**2.13.1 Programación.** Para programar el micro necesitamos utilizar el IDE de Arduino, en este se escribe un programa que posteriormente con la ayuda de un usb se transmite al micro.

Para iniciar el proceso es necesario ejecutar arduino.exe desde la carpeta c:/arduino. Se debe configurar el puerto a usar (como requisito hay que instalar el driver de nuestra placa para que el puerto pueda ser reconocido, buscarlo en c:/arduino/drivers/FTDI USB Drivers). Podemos observar el puerto a utilizar Equipo>Propiedades del Sistema>Herramientas> Administrador de dispositivos> Puertos.

Arduino posee un lenguaje de programación muy fácil. En general un programa consta de una estructura como la que se muestra a continuación

**Tabla 2-2.** Lenguaje de programación Arduino

Columna a	Columna b
// Titilar: Programa que hace que un	Led parpadee
// Declaración de variables int ledrojo=13	// Asocia el 13 al nombre ledrojo
// Decalración de set up Void set up()	// Configuración de los puertos
{ PinMode (ledrojo, OUTPUT)	// Configura el Pin 13 como una salida

**Fuente:** (ARDUINO, 2009)

La columna A nos muestra lo que interpreta el microprocesador y la columna B es una explicación para poder entender el lenguaje del micro. Es de suma importancia mantener el orden y la limpieza al momento de realizar un programa además del uso de comentarios //, y por último la tabulación en los bucles para hacer más fácil su comprensión.

**2.13.1.1 Declaración de variables.** Este punto del proceso es muy importante ya que se declara todas las variables que posteriormente va a necesitar el programa. Permite relacionar nombres con números esto facilitara el proceso al momento de modificar un programa.

**2.13.1.2 Void setup{}**. Es la función de configuración de los pines de Arduino y sólo se ejecuta una vez, en tanto que loop() se ejecuta de manera repetida mientras el sistema se encuentre encendido, o se agote la batería

## **2.14 Sistema GPS**

GPS es un sistema de posicionamiento global por medio de satélites fue creado por el Departamento de la Defensa de los E.U., fue concebido para cumplir los estrictos requerimientos de navegación y posicionamiento con fines militares. Hoy en día su uso es fundamental para aplicaciones de navegación, localizar puntos en tierra, mar y aire (INEGI, 2006).

Las coordenadas se consiguen a través de la determinación de las distancias a los satélites disponibles en ese momento. Este desarrollo de información tiene el nombre de "trilateración". Durante este proceso se debe comparar la transmisión de la señal desde el satélite y la captación de la misma en el receptor, la diferencia de los dos tiempos permite saber cuál fue el tiempo exacto para que la señal viaje desde el emisor hacia el receptor. El rango de distancia entre ambos está determinado por el producto entre la velocidad de la luz y el tiempo de viaje. (INEGI, 2006).

## **2.15 Comunicación GSM**

LA comunicación GSM es un sistema estándar, libre de regalías, de telefonía móvil digital. Un cliente GSM puede conectarse a través de su teléfono con su computador y enviar y recibir mensajes por correo electrónico, faxes, navegar por Internet, acceder con seguridad a la red informática de una compañía (red local/Intranet), así como utilizar otras funciones digitales de transmisión de datos, incluyendo el servicio de mensajes cortos(SMS) o mensajes de texto (Siegmund M. Redl, 1995).

## **2.16 Comunicación GPRS**

El sistema GPRS fue creado en la década de los noventa, las redes de telefonía móvil se encontraban en su segunda generación. Esta se caracterizaba por usar sistemas digitales, frente a los analógicos que se venían usando hasta el momento. Se conseguía de esta forma mejorar la calidad de la señal de voz y dar posibilidad de transmitir paquetes de datos. El estándar que se impuso en Europa, y sigue usándose en la actualidad, es GSM (PRIETO, 2007).

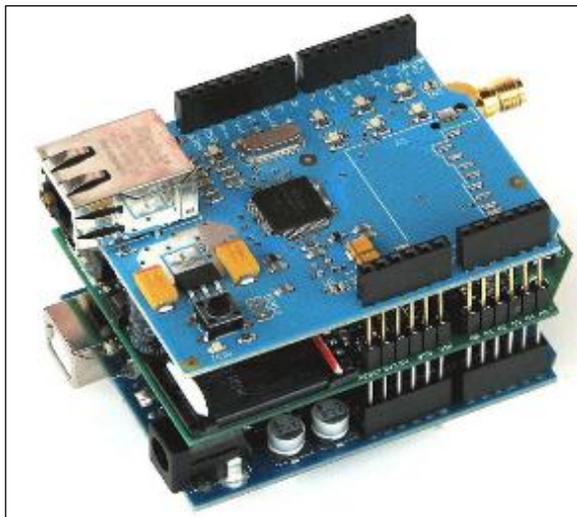
El GPRS aparece como una evolución no traumática de la actual red GSM, no conlleva

grandes inversiones y reutiliza parte de las infraestructuras actuales de GSM. Por este motivo, GPRS tiene desde sus inicios la misma cobertura que la actual red GSM. Los principales objetivos de esta tecnología son:

- Mantener los equipos de transmisión y la misma interfaz radio que GSM.
- Lograr Transmitir datos a mayor velocidad realizando las modificaciones (PRIETO, 2007).

## 2.17 Shield para Arduino

Un shield es utilizado para expandir el hardware y el rango de trabajo del Arduino, es circuito integrado que se puede apilar en el Arduino o sobre otro shield, (ARDUINO, 2009).



**Figura 14-2.** Shields para Arduino

**Fuente:** (ARDUINO, 2009)

Para tener comunicación con el Arduino lo hacen mediante pines digitales o analógicos o por algún bus de datos como el SPI, I2C o puerto serie, también pueden usar pines como interrupción. Comúnmente los shields se alimentan por medio del Arduino a través del pin de 5V y el pin GND.

En a ocasiones es posible observar que los shields no tienen soldadas sus patillas o en otros casos faltan componentes soldados. En Arduino es necesario que cada shield tenga

la misma forma que el standard de Arduino con un espaciado de pines de esta forma solo existe una manera de encajarlo(ARDUINO, 2009).

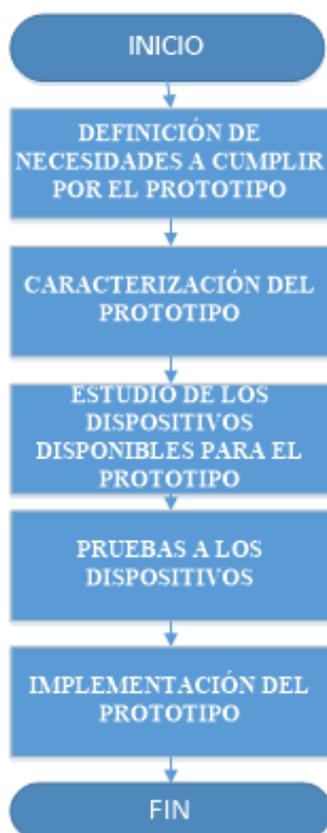
## CAPÍTULO III

### 3. DISEÑO Y MATERIALES DEL PROTOTIPO

En el presente capítulo se realizará el diseño del circuito electrónico y para su posterior implantación en un prototipo.

#### 3.1 Metodología

La principal razón para la construcción del prototipo es la de probar el funcionamiento del sistema de llamada de emergencia, para en un futuro poder ser implantado en un vehículo real, ayudando a que las personas que sufren un accidente de tránsito puedan ser atendidas con mayor rapidez. La metodología usada se muestra en la siguiente figura.



**Figura 1-3.** Metodología para diseño y construcción del prototipo

Fuente: (GARCÍA, 2016)

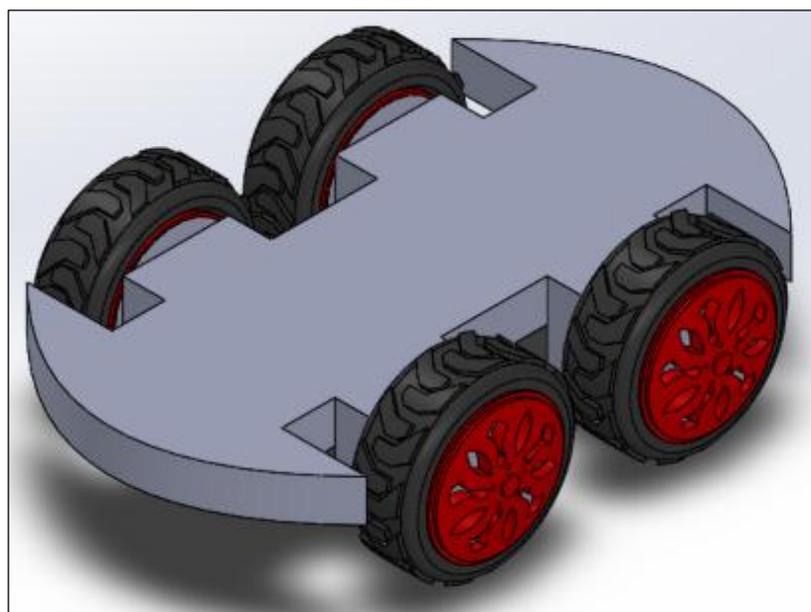
La metodología planteada en el diagrama de flujo, definen los pasos a seguir para el diseño e implementación del prototipo.

**3.1.1** *Definición de necesidades.* El prototipo a implementarse debe simular la situación real de un vehículo estándar dotado de ciertas características innovadoras que se plantean en el presente proyecto.

En mira de la innovación tecnológica el sistema a implementarse en el vehículo prototipo deberá cubrir las siguientes condiciones:

- Capacidad de conducir el prototipo por medio de una interfaz bluetooth, ayudado de una aplicación soportada bajo la plataforma Android.
- Proporcionar la ubicación geográfica del vehículo prototipo en un indicador local.
- Sensar los puntos del perímetro del vehículo prototipo que son: - Punta delantera derecha, Punta delantera izquierda, Punta trasera derecha, Punta trasera izquierda, Lateral izquierdo y Lateral derecho
- En caso de colisión del vehículo prototipo emitir señales de alarma.

**3.1.2** *Caracterización del prototipo.* El prototipo de vehículo debe contar con un chasis de soporte para todos los elementos mecánicos y electrónicos. Además debe contar con una carrocería.

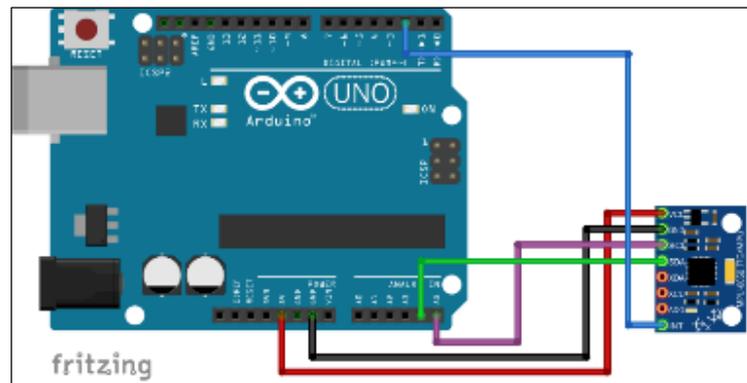


**Figura 2-3.** Chasis del prototipo

Fuente: Autor

**3.1.3** *Estudio de los dispositivos disponibles.* Se tiene varias opciones a elegir, entre las principales tenemos a los siguientes:

**3.1.3.1** *Arduino mega 2560.* El encargado de recibir y procesar la información en nuestro sistema es el Arduino mega 2560.



**Figura 3-3.** Conexión Arduino Mega

**Fuente:** <http://www.prometec.net/usando-el-mpu6050/>

Posiblemente Arduino Mega es el microcontrolador con mas capacidades en la familia Arduino. Contiene 16 entradas analógicas, 54 pines digitales que trabajan como entradas y salidas, entrada de alimentacion para la placa, un puerto de conexión USB, un pulsador para reiniciar y un cristal oscilador de 16 MHz.

La computadora y arduino se comunican mediante el puerto serie, a pesar de esto cuenta con un convertidor usb-serie, debido a esto es necesario unicamente conectar el cable usb entre el micro y la computadora. (MONK, 2012).

Las caracteristicas del arduino mega 2560 se detallan a continuación:

- Microcontrolador: ATmega2560
- Voltaje de operación: 5V
- Voltaje de Entrada: 7-12V
- Voltaje de Entrada (límites): 6-20V

- Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)
- Pines análogos de entrada: 16
- Corriente DC por cada Pin Entrada/Salida: 40 mA
- Corriente DC entregada en el Pin 3.3V: 50 mA
- Memoria Flash: 256 KB (8KB usados por el bootloader)
- SRAM: 8KB
- EEPROM: 4KB
- Clock Speed: 16 MHz

Existen dos maneras de alimentar al Arduino la primera mediante el cable USB o la segunda opción mediante una fuente, cualquiera de las dos formas que se implemente es escogida de forma automática por el dispositivo. (MONK, 2012).

Independientemente de la fuente con que se trabaje, el rango de voltaje va de 7 a 12 voltios, si el micro utiliza una fuente externa es necesario usar un convertidor AC/DC y regular el voltaje para que se encuentre dentro del límite de operación del Arduino. Otra forma de alimentación es el uso de baterías. Arduino Mega además del adaptador específico para alimentación tiene pines adicionales con el fin de poder alimentar el circuito, a continuación se detalla estos pines:

VIN: Permite alimentar la placa.

5V: Entrega 5V y una corriente de 40mA .

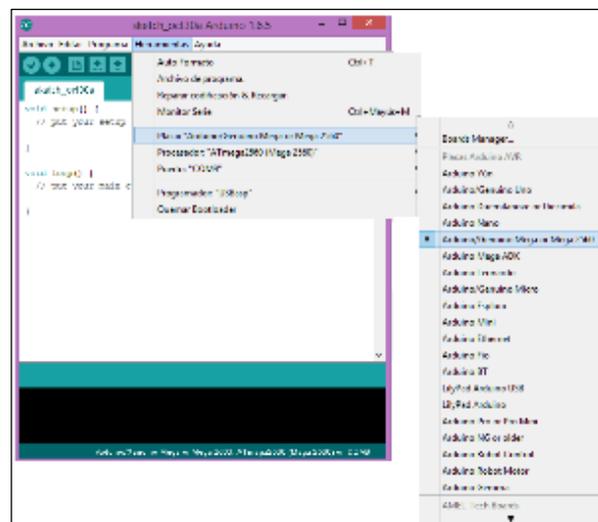
3.3V: Entrega 3.3V y una corriente de 50mA.

GND: (0V) de la placa.

**3.1.3.2 IDE (software) de Arduino.** Se tiene conocimiento sobre el microcontrolador y los pines indispensables para el funcionamiento de la placa, es necesario instalar el software para así realizar la programación por medio del computador.

Se comunica a la placa con el computador mediante el cable USB, mientras la placa permanezca conectada el led PWR (led de alimentación) tiene que permanecer encendido.

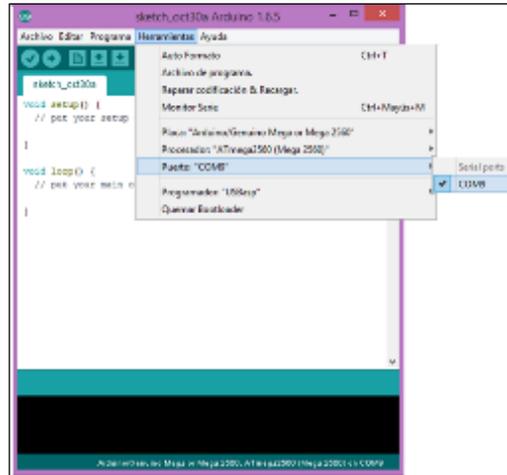
Se instala los respectivos drivers y una vez conectado el Arduino, Windows de forma automática inicia la instalación de los drivers. Al ejecutar la aplicación Arduino, se debe escoger la placa y el puerto serie. Para realizar esta acción es necesario que la aplicación se encuentre en ejecución, la opción está disponible en Tools-Board-Arduino Mega 2560.



**Figura 4-3:** IDE Arduino, Selección de la Placa Arduino Mega2560

**Fuente:** Autor

Una vez seleccionado el modelo de nuestra placa tendremos que seleccionar el dispositivo serie de la placa.

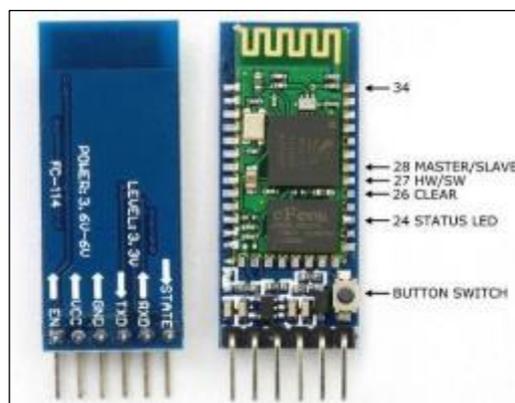


**Figura 5-3:** IDE Arduino, Selección del Puerto COM

Fuente: Autor

**3.1.3.3** *Arduino y módulo bluetooth. La manera más apropiada para controlar nuestro prototipo es mediante bluetooth ya que es un sistema inalámbrico y muy facil de usar , pues hoy en dia la mayoría de nosotros contamos con un telefono inteligente y es lo unico que se necesita para poder controlar la movilidad del prototipo, es necesario que la instalación del modulo bluetooth se haga de forma correcta y confiable para que en un futuro no exista fallas.*

Debido al costo en el mercado el módulo bluetooth HC-05 presenta un precio y características aceptables ,es un módulo Maestro-Esclavo, en otras palabras recibe conexiones desde una PC o telefono inteligente, y genera conexiones dirigidas a otros dispositivos bluetooth. La ventaja de este modulo es la capacidad de conectar dos módulos bluetooth y construir una conexión punto a punto y de esta manera transmitir datos entre distintos dispositivos (MONK, 2012).



**Figura 6-3.** Módulo Bluetooth HC-05

Fuente: <http://cursoarduinoomega.blogspot.com/2015/09/hc-06-bluetooth-module-slave-with.html>

El BlueTooth HC-05 es un shield que para comunicarse con el modulo arduino es posible escoger entre los pines D0 y D7 que son pines digitales en el arduino. El objetivo es usar estos pines como RX y TX y asi tener comunicaci3n con el m3dulo HC-05 (MONK, 2012).

Se puede comunicar unicamente dispositivos entre maestro y esclavo. El resto de configuraciones tales como maestro/maestro o esclavo/esclavo son imposibles de realizar. Por ejemplo el modo esclavo puede funcionar para conectar con PDA, PC, etc. y otros dispositivos que por su configuracion se conectan como maestros (MONK, 2012).

A continuacion se detalla las caracteristica generales del modulo bluetooth HC-05

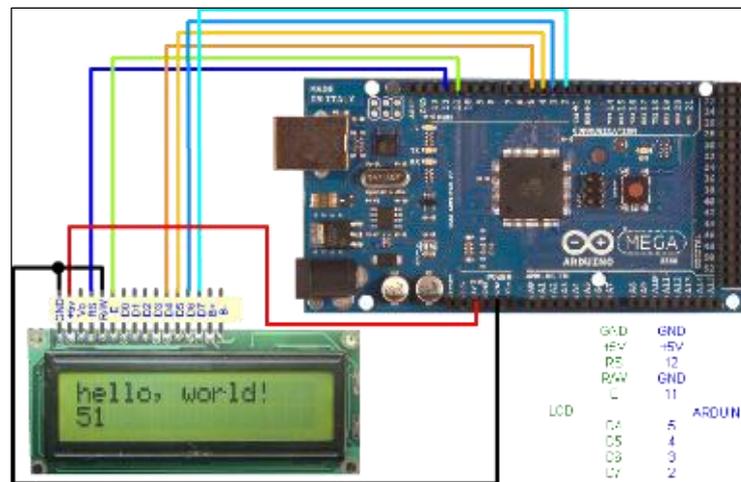
- Compatible con Arduino
- Sensibilidad Típica: -80dBm.
- Hasta +4 dBm de potencia de transmisi3n RF.
- Fully Qualified Bluetooth V2.0 +modulation EDR 3Mbps.
- Funcionamiento de bajo consumo.
- PIO control.
- Interfaz UART con velocidad de modulaci3n en baudios programable.
- Antena PCB Integrada.

Las característic3s del software son las siguientes:

- Velocidad en baudios (Modo comandos AT): 38400, Bits de datos: 8, Bit de parada: 1, Paridad: Sin paridad.
- Tasa de velocidad de modulaci3n en baudios soportadas: 9600, 19200, 38400, 57600, 115200, 230400, 460800.

- Auto-conexión del dispositivo con la última configuración por defecto.
- Permiso conectar el dispositivo emparejado de forma predeterminada.
- Por defecto PINCODE: "1234".
- Reconexión automática en 30 min cuando se desconecta como consecuencia de pérdida de conexión por salirse del rango de alcance.

**3.1.3.4** *Arduino y LCD I2C.* Para la visualización de la ubicación del vehículo prototipo asignado por un valor de latitud y de longitud se utiliza un lcd de 16x2 con conexión I2C.



**Figura 7-3.** Conexión Arduino Mega y LCD con módulo I2C

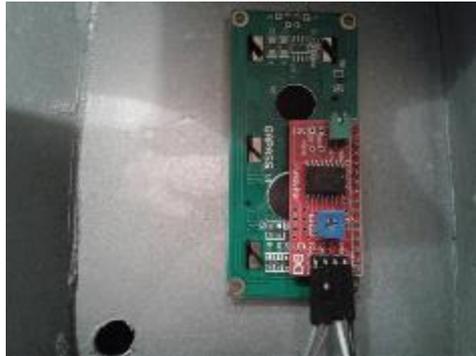
Fuente: <http://unicarlos.com/pezazul/index.php?articulo=7>

Las pantallas LCD requieren el uso de muchos pines de tipo digital para que logren funcionar, generalmente el número varía de 6 a 13 este valor depende de la pantalla, esto se convierte en un problema debido a que resta pines útiles a nuestro Arduino, quitando la posibilidad de conectar otros dispositivos, este problema puede resolverse de manera sencilla, se debe usar un módulo cuyo función es la de convertir la conexión de la pantalla, la transformación se realiza de paralelo a conexión en serie, este proceso se debe llevar a cabo con la utilización de protocolos de comunicación compatibles para Arduino (SABIK, 2010).

Podemos encontrar varios tipos de módulos por ejemplo con conexión ISP, Serial e I2C,

siendo el de tipo serial el mas popular.

**3.1.3.5 Módulos I2C.** *El encargado de desactivar la iluminación es el jumper. El modulo reduce la conexión únicamente a 4 cables que se dividen en solo tiene 4 cables GND, 5V, SDA y SCL. Los pines necesarios para la conexión son SDA y SCL y estos varían de acuerdo a la placa por lo que debemos revisar su ubicación*

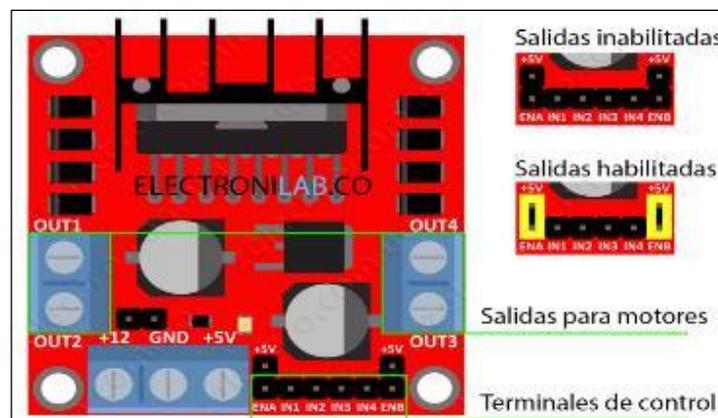


**Figura 8-3.** LCD i2C 16x2

Fuente: Autor

Además de la conexión se debe instalar una librería adicional debido a que el IDE contiene una librería por defecto que específicamente funciona si la pantalla está conectada de forma directa, el nombre de la librería adicional es LiquidCrystal.h

**3.1.3.6 Puente H LN298.** *Los motores encargados del movimiento del prototipo son DC , estos reciben ordenes de un módulo en nuestro caso el puente H, por esta razón se selecciona el puente H y Arduino; la base de este módulo está en su chip que es capaz de controlar dos motores DC o a su vez un motor paso a paso de tipo bipolar con una corriente máxima de 2 amperios. (MONK, 2012).*



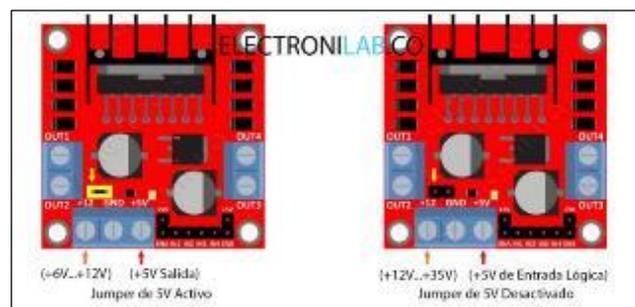
**Figura 9-3.** Terminales puente H, módulo de Arduino

Fuente: <http://electronlab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>

El módulo funciona sin elementos adicionales, los componentes más importantes del módulo son los diodos de protección, regulador LM7805 este es el encargado de entregar 5V que son ocupados por la parte lógica del L298N. Además, posee jumpers de selección esto sirve para operar las salidas del módulo en forma individual (A y B). La salida A está la forman OUT1 y OUT2 y la salida B constituida por OUT3 y OUT4, además se necesita los pines que sirven para habilitar en este caso son ENA y ENB respectivamente (MONK, 2012).

El módulo cuenta con 4 pines de control que se ubican en la parte inferior del mismo, identificados como IN1, IN2, IN3 e IN4.

El módulo cuenta con dos tipos de conexión para poder recibir alimentación esto debido al uso del regulador integrado LM7805.



**Figura 10-3.** Conexiones de alimentación al puente H, módulo de Arduino

**Fuente:** <http://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>

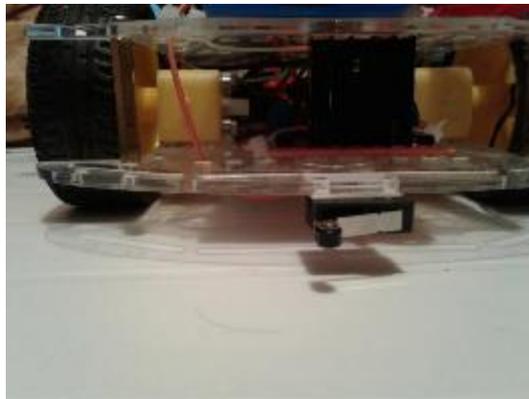
Para que el módulo autorice la alimentación el jumper de selección debe encontrarse activo esta alimentación se encuentra en el rango de 6V a 12V con corriente directa. El regulador está activo, por consecuencia el pin identificado como +5V tiene un voltaje de 5 voltios en corriente directa. Es recomendable que el consumo no supere los 500mA, independientemente de su aplicación ya que el voltaje puede alimentar el control del módulo pudiendo ser microcontrolador o Arduino, (MONK, 2012).

Caso contrario si el jumper de 5V de selección está en modo off, el módulo admite una alimentación dentro del rango 12V a 35V en corriente directa. Debido a que el jumper está en modo off por ende el regulador tampoco funciona, es necesario que el pin de +5V se encuentre conectado con una tensión de 5V y de esta manera pueda funcionar la parte lógica del L298N. Generalmente la tensión es similar en la parte de control, independiente

si es microcontrolador o Arduino (MONK, 2012).

Nunca se debe alimentar el pin de +5V, cuando el jumper de selección de 5V se encuentre seleccionado debido a que podría generar un corto y averiar de forma permanente el módulo (MONK, 2012).

**3.1.3.7** *Arduino – I/O digital. Para la detección de las colisiones se requieren señales de tipo discreto que indiquen SI HAY COLISIÓN o NO HAY COLISIÓN. En Arduino las señales discretas son interpretadas como HIGH o LOW lo que permitirá relacionarlas con el estado de las colisiones, para ello se utiliza los sensores mostrados en la siguiente figura.*



**Figura 11-3.** Sensor / Entrada Digital

Fuente: Autor

La selección de los sensores se la realizará en base al requerimiento planteado, que tenga salidas del tipo digital que puedan ser manejadas por el controlador Arduino.

**3.1.3.8** *Arduino GPS. El módulo GPS de Arduino proporcionará los datos de longitud y latitud para determinar la ubicación geográfica del vehículo prototipo.*

El módulo GPS en su modelo GY-GPS6MV2 viene con un módulo de serie U-Blox NEO 6M equipado en el PCB, una EEPROM con configuración de fábrica, una pila de botón para mantener los datos de configuración en la memoria EEPROM, un indicador LED y una antena cerámica. También posee los pines o conectores Vcc, Rx, Tx y Gnd por el que se puede conectar a algún microcontrolador mediante una interfaz serial. Para que nuestro módulo GPS funcione a la perfección se recomienda hacer las pruebas en un ambiente abierto o cercano a la ventana para una correcta recepción de la señal. (MONK, 2012)



**Figura 12-3.** Sensor / Entrada Digital

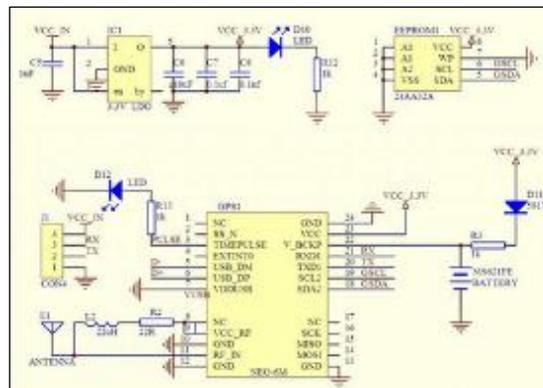
**Fuente:** [http://www.naylampmechatronics.com/blog/18\\_Tutorial-M%C3%B3dulo-GPS-con-Arduino.html](http://www.naylampmechatronics.com/blog/18_Tutorial-M%C3%B3dulo-GPS-con-Arduino.html)

A continuación se describen algunas de las características del módulo:

- Modulo GPS Ublox NEO-6M
- Comunicación serial
- Voltaje de alimentación: (3.5 – 5 )VDC
- Antena cerámica activa incluida
- LED indicador de señal
- Tamaño de antena 22x22mm
- Tamaño de modulo 23x30mm
- Batería incluida
- BAUDRATE: 9600
- EEPROM para guardar configuración de parámetros
- Sistema de coordenadas: WGS-84
- Sensibilidad de captura -148dBm

- Sensibilidad de rastreo: -161 dBm
- Máxima altura medible: 18000
- Máxima velocidad 515 m/s
- Exactitud: 1micro segundo
- Frecuencia receptora: L1 (1575.42 Mhz)
- Código C/A 1.023 Mhz
- Tiempo de inicio primera vez: 38s en promedio
- Tiempo de inicio : 35s en promedio

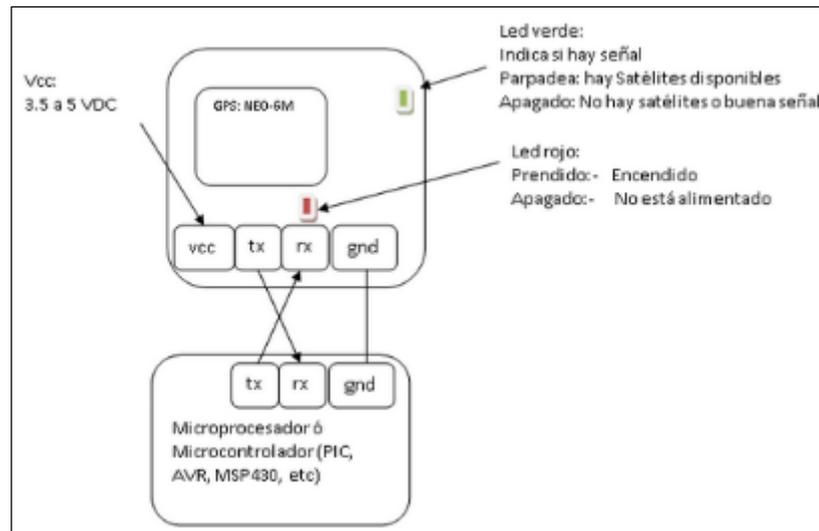
A continuación en la siguiente figura se muestra el diagrama esquemático de la tarjeta, se puede observar que incluye además del circuito NEO-6M una memoria EEPROM la cual se puede utilizar para guardar cualquier dato que se quiera (MONK, 2012).



**Figura 13-3.** Diagrama esquemático de la tarjeta de control

Fuente: <http://hetpro-store.com/TUTORIALES/gps-ublox-neo-6m-modulo-con-matlab/>

A continuación se muestra el esquema de conexiones.



**Figura 14-3.** Conexiones

Fuente: <http://hetpro-store.com/TUTORIALES/gps-ublox-neo-6m-modulo-con-matlab/>

**3.1.3.9** *Tarjeta SIM 900 GSM/GPRS. El Shield GPRS se basa en el módulo SIM900 de SIMCOM, y es compatible con Arduino y sus clones; este Shield proporciona una manera de comunicarse a través de la red de telefonía celular GSM.*



**Figura 15-3.** Shield Sim900 GSM/GPRS

Fuente: <http://tecmikro.com/modulos-de-rf-modulos-bluetooth/483-modulo-gsmsim900-shield-arduino.html>

Las especificaciones de esta tarjeta son las siguientes:

- Quad-Band 850/900/1800/1900 MHz - Redes GSM en todos los países de todo el mundo.
- GPRS clase 10.08 multi-slot

- GPRS estación móvil de clase B
- Cumple con GSM fase 2/2 +
- Clase 4 (2 W (AT) 850/900 MHz)
- Clase 1 (1 W (AT) 1800 / 1900MHz)
- Control a través de comandos AT - Comandos estándar: GSM 07.07 y 07.05 | Comandos extendidos: Comandos AT Simcom.
- Servicio de mensajes cortos - de manera que usted puede enviar pequeñas cantidades de datos a través de la red (ASCII o hexadecimal en bruto).
- Pila TCP/UDP incorporada - le permite cargar datos a un servidor web.
- Compatible con RTC.
- Puerto serie seleccionable.
- Tomas para altavoz y auriculares
- Bajo consumo de energía - 1,5 mA (modo de espera)
- Rango de temperatura industrial - -40 ° C a +85 ° C

## CAPÍTULO IV

### 4. IMPLEMENTACION Y PRUEBAS DEL PROTOTIPO

En el capítulo anterior se realizó el respectivo diseño del circuito electrónico que va a ser implantado en el sistema de llamada de emergencia, en el presente capítulo se tratará sobre el proceso de ensamblado total del sistema y la respectiva implementación en el auto prototipo.

Se debe satisfacer todos los requisitos planteados en la metodología de esta manera se determinan los equipos y materiales indispensables además la conexión de todos los elementos para que actúen de forma conjunta y así lograr la implementación del prototipo.

#### 4.1 Esquema general de conexiones del auto prototipo

La figura que se muestra a continuación representa el montaje final de cada elemento utilizado en la construcción del prototipo, se observa cada elemento con sus interfaces de funcionamiento para la adaptación de señales e interfaces de potencia que sirven para el manejo de actuadores

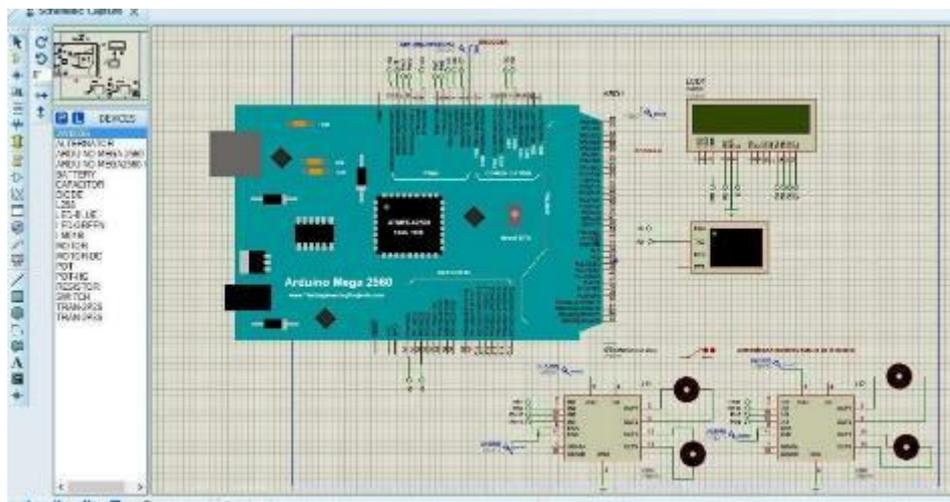


Figura 1-4. Esquema de conexiones eléctricas en el prototipo

Fuente: Autor

La base del control para el prototipo esta comandada por un Arduino Mega 2560 cuyas características fueron estudiadas anteriormente, Arduino mega es el responsable de procesar todas las señales de control enviadas por los elementos que constituyen el

prototipo de igual manera recibir y procesar señales necesarias para el control de variables que actúan en el funcionamiento de nuestro prototipo, este complejo proceso se lleva a cabo basado en el programa cargado anteriormente mediante el software Arduino en el microcontrolador.



**Figura 2-4.** Conexiones prototipo / Sistema centralizado con Arduino Mega 2560

**Fuente:** Autor

El resultado de la aplicación del diagrama de conexiones eléctricas propuesto para la implementación del prototipo se encuentra en la imagen anterior, ya se describió en párrafos anteriores que todo el sistema está centralizado en el Arduino mega 2560.

En los párrafos siguientes se especifica el programa final del microcontrolador y se detalla cada sentencia programada que son necesarias para administrar los diferentes dispositivos que forman parte del prototipo.

**4.1.1 Programación del prototipo.** El programa principal para el trabajo del prototipo se debe empezar por declarar librerías, estas son utilizadas en lo posterior para el manejo y configuración de las características de cada elemento o módulo que actúa en el proceso de trabajo.

- `#include <TimerOne.h>`
- `#include <LiquidCrystal.h>`

El programa general contiene 2 librerías que son la `TimerOne.h` que es necesaria para el control de la función única que se encarga del monitoreo de la velocidad, tiene cierto parecido con el conteo de señales de pulso que son emitidas por el opto acoplador ambas funcionan formando el encoder.

Después de nombrar las librerías el siguiente paso es la declaración de variables, en este punto se definen los pines que posteriormente se va a utilizar del microcontrolador además se declaran variables globales que van estar presentes en el programa.

**4.1.2** *LiquidCrystal lcd(13, 12, 8, 5, 4, 3).* Se empieza por declarar la variable del tipo `lcd` esta será usada en el control del LCD `i2c` que es usado para observar variables y estados dentro del prototipo. En este punto se establece comunicación serial.

```
int Ma1 = 11
int Ma2 = 10
int Mb1 = 7
int Mb2 = 6
```

Las variables asignadas para el control de los 4 motores son `Ma1`, `Ma2`, `Mb1`, `Mb2` del estado de estas en la función escrita para controlar los motores enviaran la señal para avance y retroceso de los mismos. Al mismo tiempo las mismas líneas de código indican los pines físicos del microcontrolador Arduino que están comunicados mediante cables a los pines del módulo puente H del L298N utilizados para gobernar el movimiento de los motores.

```
void setup()
{
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(encoder_pin, INPUT);
  pinMode(Ma1, OUTPUT);
  pinMode(Ma2, OUTPUT);
  pinMode(Mb1, OUTPUT);
  pinMode(Mb2, OUTPUT);
}
```

```
}
```

El void setup ( ) es el encargado de empezar todos los elementos necesarios para el programa, ejecución de sentencias que serán cumplidas por el prototipo. Además se definen las funciones de cada pin que ya fue declarado anteriormente en el grupo de variables, aquí concreta si están actuando como entradas (INPUT), salidas (OUTPUT) o a su vez señales de PWM. Cada módulo que se encuentra conectado al micro tiene funciones propias.

La función primordial en el programa es la void loop ( ), esta define el total de acciones que serán ejecutadas de manera repetida y continua por el microcontrolador para definir el trabajo de los módulos que administra.

```
void loop ( )
```

```
{
```

```
if(Serial.available() > 0)
```

Verifica el estado del puerto Serial que será mayor de cero si existe tráfico de datos a través de él.

```
{
```

```
  estado = Serial.read();
```

```
  Serial.println(estado);
```

El siguiente bloque de programa establece la secuencia de activación de los motores del auto prototipo para realizar un giro hacia la izquierda.

```
  if (estado == 'd')
```

```
  {
```

```
    digitalWrite(Ma1,LOW);
```

```
    digitalWrite(Ma2,HIGH);
```

```
    digitalWrite(Mb1,LOW);
```

```
    digitalWrite(Mb2,HIGH);
```

```
  }
```

El siguiente bloque de programa establece la secuencia de activación de los motores del auto prototipo para realizar un giro hacia la derecha.

```
if (estado == 'b')
{
digitalWrite(Ma1,HIGH);
digitalWrite(Ma2,LOW);
digitalWrite(Mb1,HIGH);
digitalWrite(Mb2,LOW);
}
```

El siguiente bloque de programa establece la secuencia de activación de los motores del auto prototipo para avanzar hacia atrás.

```
if (estado == 'e')
{
digitalWrite(Ma1,HIGH);
digitalWrite(Ma2,LOW);
digitalWrite(Mb1,LOW);
digitalWrite(Mb2,HIGH);
}
```

El siguiente bloque de programa establece la secuencia de activación de los motores del auto prototipo para avanzar hacia adelante.

```
if (estado == 'a')
{
digitalWrite(Ma1,LOW);
digitalWrite(Ma2,HIGH);
digitalWrite(Mb1,HIGH);
digitalWrite(Mb2,LOW);
}
```

El siguiente bloque de programa establece la secuencia de desactivación de los motores del auto prototipo establece un estado de reposo.

```
if (estado == 'c')
{
    digitalWrite(Ma1,LOW);
    digitalWrite(Ma2,LOW);
    digitalWrite(Mb1,LOW);
    digitalWrite(Mb2,LOW);
}
}
```

Las secciones de programa finales son aquellas que controlan directamente a los motores según el caso codificado de información que llega del Serial obtenidas por la comunicación inalámbrica con el bluetooth.

A continuación se muestra las líneas de programación del sistema de llamada:

El siguiente bloque de programación nos muestra la programación que controla la tarjeta sim:

```
{

digitalWrite(9, HIGH);

delay(1000);

digitalWrite(9, LOW);

delay(5000);

Serial.print("encendido");
```

```
}
```

El siguiente bloque muestra el proceso para realizar la llamada

```
{
```

```
digitalWrite(9, HIGH);
```

```
delay(1000);
```

```
digitalWrite(9, LOW);
```

```
delay(5000);
```

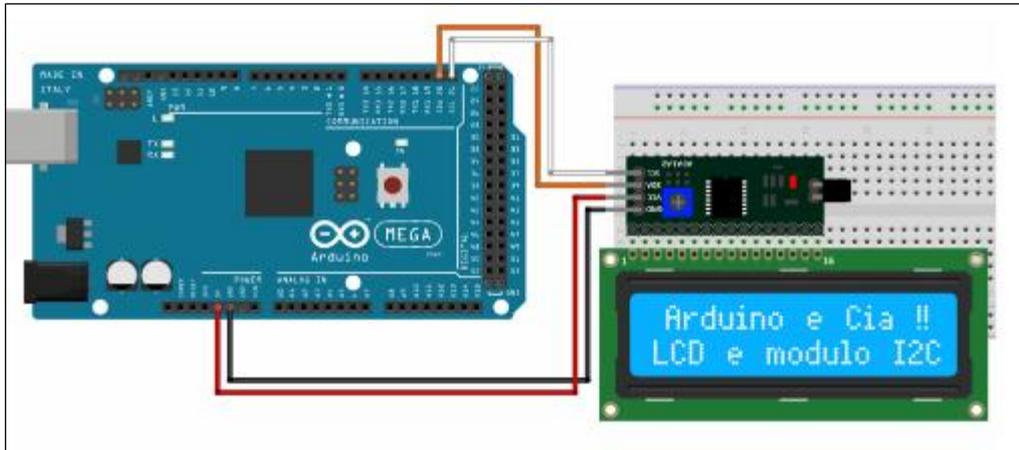
```
Serial.print("encendido");
```

```
}
```

El siguiente bloque nos muestra el proceso para enviar el mensaje de texto con la ubicación

El siguiente sub capítulo muestra las conexiones finales entre los distintos dispositivos y presenta evidencia del trabajo en conjunto de estos en campo y ya montados en el prototipo.

**4.1.3** *Conexión Arduino y LCD 12C.* La función del LCD como se estudió en el capítulo 3 será permitir la visualización de parámetros del auto prototipo.



**Figura 3-4.** Conexión Arduino Mega y módulo LCD I2C 16x2

Fuente: <http://lanzarduino.beautifulcode.com/wp-content/uploads/2016/02/lcd1.png>



**Figura 4-4.** Pantalla de visualización en el LCD

Fuente: Autor

Como se puede observar en las figuras anteriores los parámetros que se visualizan en el LCD incluido en el auto prototipo son:

**Pantalla 1:** Alternada a los datos de la pantalla A aparecerán continuamente los valores de latitud y longitud como información sobre la posición geografica facilitada por el GPS Ublox.

La siguiente figura muestra la ubicación física de la pantalla lcd en el vehículo prototipo.

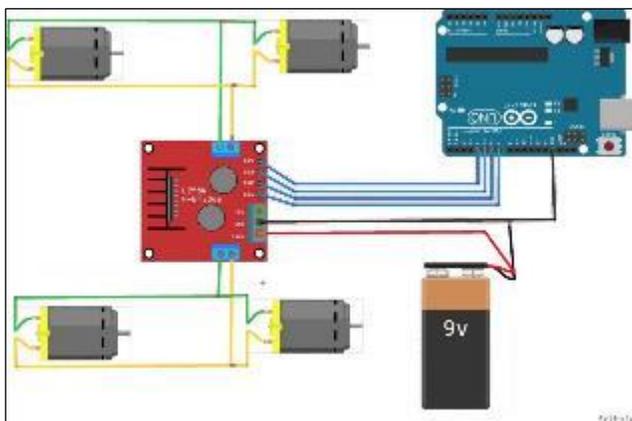


**Figura 5-4.** Montaje LCD en el prototipo

Fuente: Autor

**4.1.4** *Arduino & Módulo Puente H L298N.* El movimiento del prototipo es de suma importancia, se alcanzó de manera exitosa la administración de los 4 motores, instalados uno en cada neumático, estos a su vez reciben del microcontrolador los códigos de funcionamiento para el movimiento del prototipo.

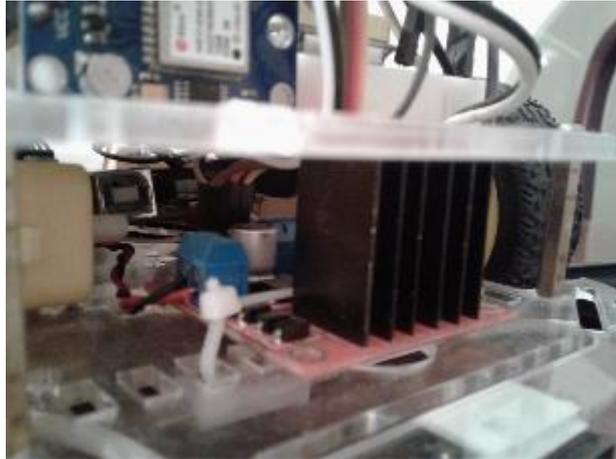
El shield puente H con circuito integrado LN298N cumplió con nuestros requerimientos por esta razón se lo escogió y finalmente fue utilizado en nuestro prototipo. La siguiente figura muestra la conexión del puente H en conjunto con los motores y el microcontrolador.



**Figura 6-4.** Conexión Arduino y módulo L298N

Fuente: <http://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>

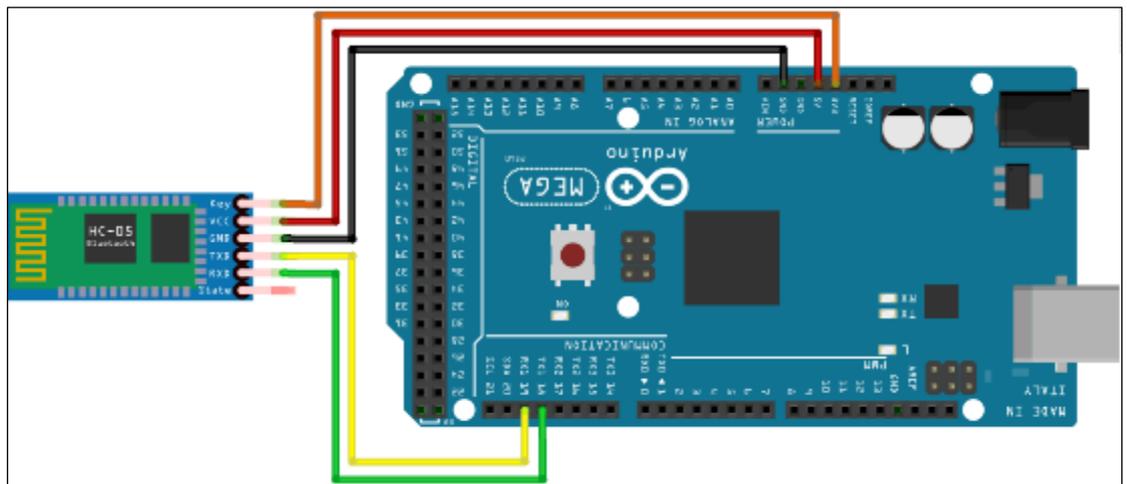
La figura que se muestra a continuación muestra la conexión real del puente H ya en funcionamiento dentro del prototipo.



**Figura 7-4.** Conexión Puente H LN298 motores delanteros.

Fuente: Autor

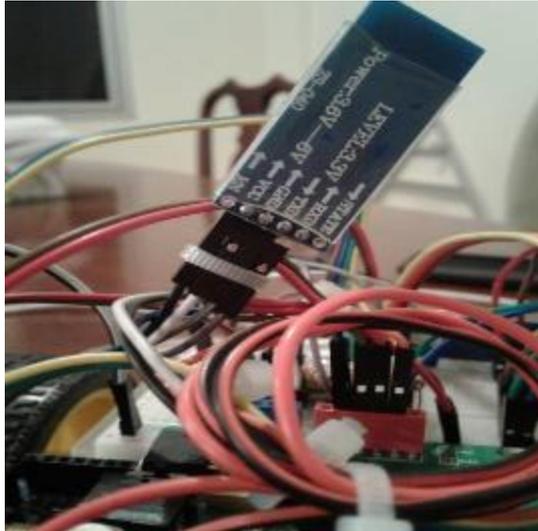
**4.1.5** *Arduino y módulo bluetooth HC-05.* La siguiente figura muestra de manera clara la conexión entre módulo bluetooth y el Arduino, se puede observar de manera clara los pines que se encuentran conectados.



**Figura 8-4.** Conexión Arduino Mega y módulo Bluetooth H-05

Fuente: [http://blascarr.com/wp-content/uploads/2015/05/hc-05-Mega\\_bb.png](http://blascarr.com/wp-content/uploads/2015/05/hc-05-Mega_bb.png)

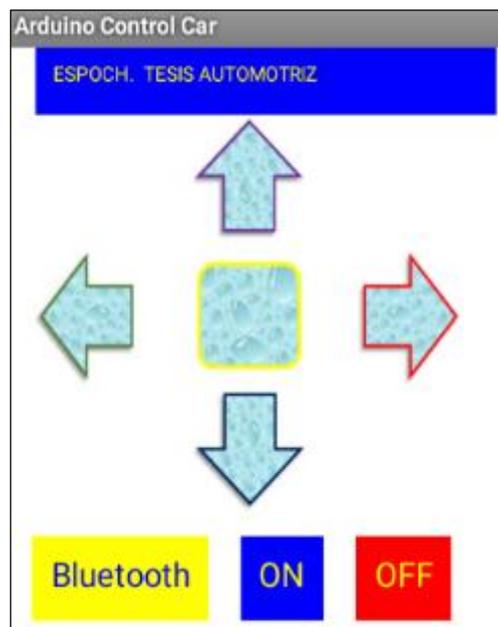
Como producto de todo el proceso de conexión se obtiene el funcionamiento de forma inalámbrica del prototipo, el módulo se encuentra conectado con nuestro smartphone e interactúa con el entorno.



**Figura 9-4.** Conexión módulo bluetooth

Fuente: Autor

El prototipo esta controlado por una interfaz creada por android Android para smarthphones operados por el mismo sistema operativo; la imagen muestra la interfaz la cual es similar a una palanca de direcciones que permite elegir la dirección del vehículo.



**Figura 10-4.** Interfaz de control Android

Fuente: Autor

## 4.2 Resultados obtenidos

Se logró construir un prototipo capaz de comunicarse automáticamente con el sistema integrado ecu 911 mediante un sistema GPS el cual detecta la ubicación del vehículo y GPRS que es el encargado de realizar la llamada y enviar el mensaje de auxilio con la ubicación de vehículo todos los componentes son controlados por un Arduino mega que es el encargado de procesar la información y realizar las acciones correspondientes.

Posee un LCD que nos muestra en estado normal la posición del vehículo y en caso de choque muestra la parte afectada del vehículo (delantera, posterior, lateral).

Para su movimiento el prototipo utiliza una interfaz inalámbrica mediante el dispositivo bluetooth el cual se comunica con un móvil con sistema operativo Android, utilizando la aplicación Arduino control car la cual se encuentra disponible de forma gratuita en la tienda App Store, mediante la cual el prototipo se desplaza en todas las direcciones.

## 4.3 Pruebas realizadas

Se realizó un total de 30 pruebas en cada sensor teniendo en cuenta como 1 cuando el sistema funciona correctamente y 0 cuando no funciona en base a estos datos obtenidos realizamos el análisis de confiabilidad utilizando el coeficiente de confiabilidad KR20 que ocupa la siguiente fórmula  $coe = \frac{n}{n-1} \left[ 1 - \frac{\sum p*q}{s_t^2} \right]$ , éste tipo de análisis es utilizado cuando los datos de las pruebas realizadas están determinados por sí o no , en nuestro caso 1 y 0, el cual nos arrojó un resultado del 87 % de confianza.

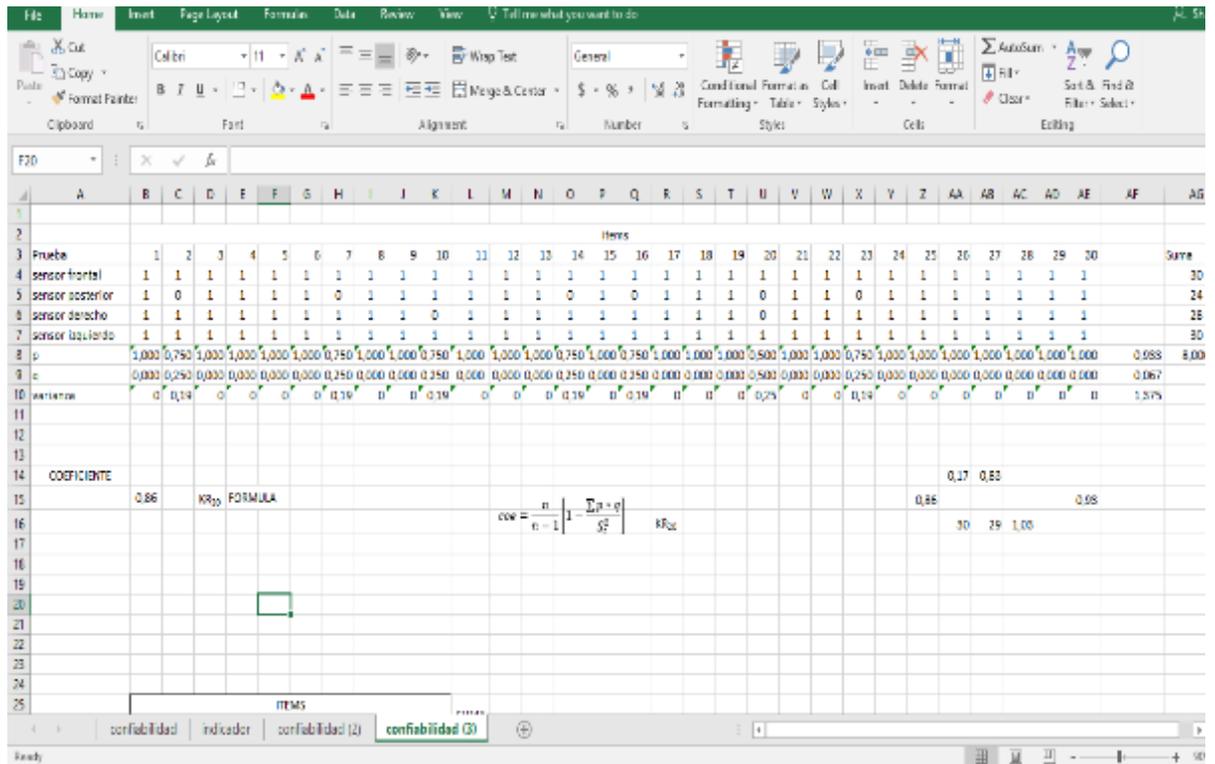


Figura 11-4. Análisis de confiabilidad

Fuente: Autor

## **CAPÍTULO V**

### **5. CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 Conclusiones**

- Se concluyó que los sensores que se acoplan de mejor manera y que poseen las características requeridas son los fines de carrera por lo que han sido seleccionados para la construcción del prototipo.
- Se realizó el diseño usando el software de programación Software Arduino, el cual posee todas las herramientas necesarias para el diseño del circuito del prototipo, además se realizó una interfaz Android, para que el vehículo prototipo sea conducido utilizando un Smartphone o Tablet con la respectiva aplicación a través de una conexión bluetooth.
- Se construyó el prototipo con todos los elementos necesarios para que pueda de manera automática detectar cuando existe una colisión y realizar la respectiva llamada de emergencia al ECU 911 junto con la posición geográfica en la que se encuentra.
- Se comprobó el correcto funcionamiento del prototipo mediante pruebas y ensayos exhaustivos que demostraron la efectividad del mismo.

#### **5.2 Recomendaciones**

- Se recomienda utilizar un dispositivo Android que sea compatible con la aplicación necesaria para conducir el vehículo prototipo
- Es importante comprobar el correcto funcionamiento de los sensores de impacto utilizados en el vehículo prototipo.
- Para el montaje de la carrocería sobre el vehículo prototipo, se recomienda usar uniones fáciles de desmontar, con la finalidad de poder acceder fácilmente al circuito electrónico.
- Siempre comprobar que se cuenta con saldo para que el sistema funcione correctamente.

## BIBLIOGRAFÍA

**ARDUINO.** *Materiales* [En línea] 2009. [Consulta: 13 de 04 de 2016.] Disponible: [http://www.cscjprofes.com/wp-content/uploads/2014/02/teoria\\_arduino2009.pdf](http://www.cscjprofes.com/wp-content/uploads/2014/02/teoria_arduino2009.pdf).

**AZKEN.** *Makina eta Motore Termikoetako Saila.* [En línea] 4 de 10 de 2010. [Consulta: 13 de 04 de 2016.] <http://www.ehu.es/mmtde/bancomot.htm#o>.

**BOHACZ, Ray.** *Training Manual Electronic Fuel Injection.* 2011. Washintong, Penguin pp 51-57

**CEA.** *Seguridad Vial* [En línea] 2010. [Consulta: 2 de 04 de 2016.] Disponible: <http://www.cea-online.es/reportajes/seguridad.asp>.

**CISNEROS, Iván.** *Los sensores en el automóvil.* 2013 México: Grupo SM pp 31-34.

**FRIES, Bruce** *Conversión Analógica y Digital* [En línea] 2005. [Consulta: 31 de 03 de 2016.] Disponible: [https://es.wikipedia.org/wiki/Conversi%C3%B3n\\_anal%C3%B3gica-digital](https://es.wikipedia.org/wiki/Conversi%C3%B3n_anal%C3%B3gica-digital).

**GARCÍA, Edu** *Conversión de señales* [En línea] 2016. [Consulta: 31 de 03 de 2016.] Disponible: [https://es.wikipedia.org/wiki/Conversi%C3%B3n\\_anal%C3%B3gica-digital](https://es.wikipedia.org/wiki/Conversi%C3%B3n_anal%C3%B3gica-digital).

**INEGI.** *Sistema de posicionamiento global.* [En línea] 2006. [Consulta: 2 de 12 de 2016.] Disponible: <http://www.inegi.org.mx/geo/contenidos/geodesia/gps.aspx?dv=c1>.

**Km77.** *Seguridad en el vehículo* [En línea] 2010. [Consulta: 04 de 03 de 2016.] Disponible: <http://www.km77.com/glosario/c/consuespe.asp>.

**LOAIZA, Carlos.** *Seguridad Activa y pasiva en un vehículo.* [En línea] 2012. [Consulta: 01 de 08 de 2016.] Disponible: <http://www.sura.com/blogs/autos/seguridad-activa-pasiva-vehiculo.aspx>.

**MONK, Simon.** *30 Arduino Proyects for the Evil Genius.* New York - EEUU : Mc Graw Hill, 2012.

**PRIETO, Francisco.** *Transmisión de imágenes de video mediante sitios web MXL sobre J2ME.* Mexico DF, Kamite 2007.

**SABIKA, Grupo.** *Ejercicios de Arduino resueltos.* Madrid : Grupo Sabika, 2010. pp 81-83.

UNE ISO 39001 . *Sistemas de Gestión De Seguridad Vial.* 2013.

**SEAT, Organización de Servicio.** *Gestiones electrónicas. Sensores y actuadores.* [En línea] 2015. [Consulta el: 02 de 08 de 2016.] Disponible: <https://es.scribd.com/document/120675634/Sensores-y-Actuadores-SEAT>.

**Siegmund M. Redl. et al.** *An Introduction to GSM*. USA : Artech House, 2014. pp 113-119

**SMITH, C.B.** *Elementos de mecánica aplicada*. Barcelona : Labor, S.A. Calabria, 1971. pp 91-100

**TELEMATIC NEWS.** *Gps* [En línea] 2015. [Consulta el: 06 de 09 de 2016.]: Disponible: <http://telematicsnews.info/about-2/>

**VALCÁRSEL, Josefa.** *Cuestiones de seguridad vial, conducción eficiente, medio ambiente y contaminación*. Madrid: Ediciones Dauro 2013. pp 45-53.

**VOLKSWAGEN.** *Glosario Técnico*. [En línea] 2010. [Consulta el: 06 de 10 de 2016.] Disponible:<http://www.volkswagenlatinoamerica.com/es/technik-lexikon/crash-sensoren.html>.

# **ANEXOS**

**Anexo A**



**Anexo B**



**Anexo C**



**Anexo D**



## Anexo E



